

IRAF İLE ÖRNEK OLARAK AU SER SİSTEMİNİN İNDİRGEMESİ

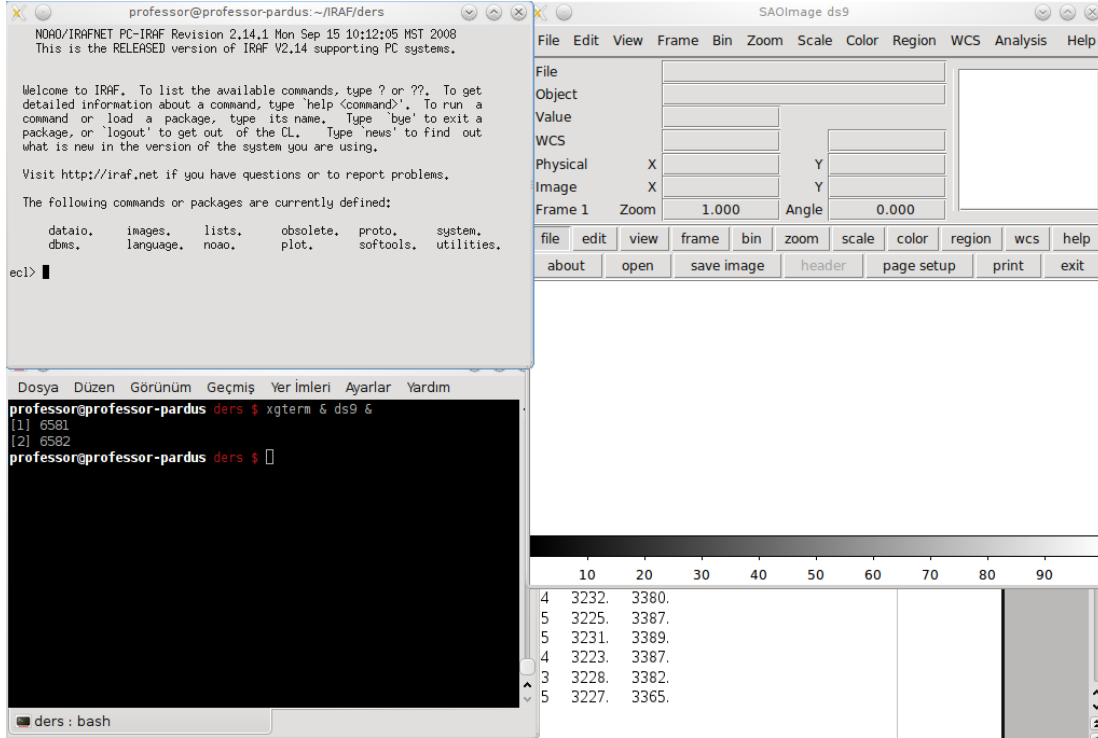
-Yücel KILIÇ – yucelkiloc@myrafproject.org

IRAF'a Hazırlık

1. Öncelikle bilgisayarımızda IRAF'ın kurulu olması gerekmektedir. Bu hususta yardım almak için bu¹ bağlantıdan yararlanabiliriz.
2. IRAF kurulumu tamamlandıktan sonra, **IRAF** klasörü içerisinde **ders** adlı bir klasör oluşturuldu. "**mkiraf**" komutu verilerek yeni çalışmaya özgü parametrelerimizin saklandığı **/uparm dizini**ve **login.cl** oluşturuldu.
3. Sonra **AU Ser** verileri **ders** dizini içerisinde atıldı. Atılan **AU Ser** verileri(AU Ser, Flat, Bias, Dark) "gz" ile arşivlenmiş olduğundan, "**gunzip *gz**" komutuyla sıkıştırılmış arşiv dosyasından çıkarıldı. Şimdi indirgemeye başlamak için herşey hazır.

IRAF'ın açılışı

1. **ders** dizini içerisinde **login.cl** ve **/uparm** dizini oluşturmuştuk verilerimiz de burada olduğu için burada "cl" komutunu verip çalışmamıza başlayabiliriz. Fakat önce **ds9** ve **xgterm** 'ü çalıştırmamız gerekecek;
2. Terminal'de "**xgterm & ds9 &**" komutunu vererek her ikisini de açabiliriz.
3. "**xgterm**" terminal penceresi açıldıktan sonra "**cl**" komutu'nu artık verebiliriz.



¹ <http://yucelkiloc.com/?p=728>

Şekil 1. IRAF'a hazırlık ve Giriş

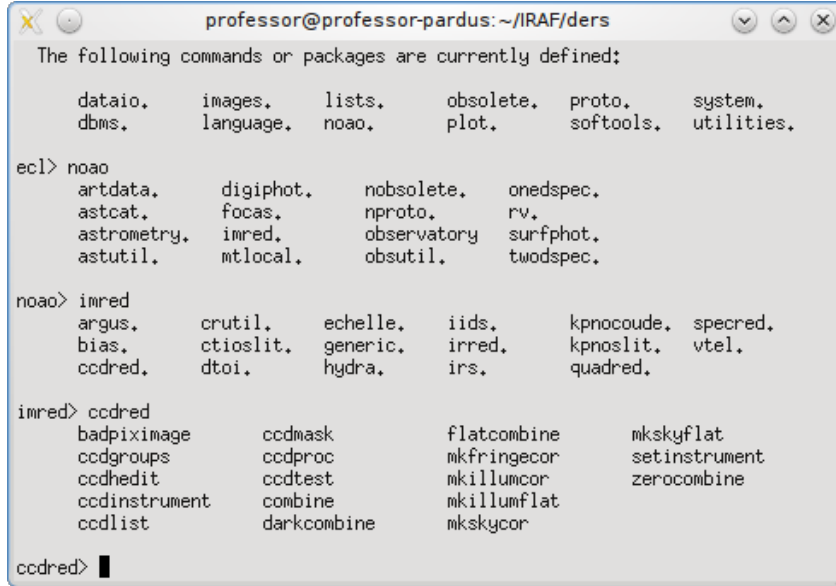
4. Veri düzenimiz Şekil 2.'deki gibidir. İndirmeye ilk olarak Bias görüntülerinden **masterbias** oluşturarak başlayalım. Bu yüzden IRAF'ta “**noao**” paketinin içerisinde bulunan “**imred > ccdred**” paketinin içerisine gidip “**zerocombine**” taskını kullanacağız. Bunun için hazırlık yapalım; (**Not: Bundan sonraki tüm işlemler xgterm üzerinde açtığımız iraf komut satırında olacaktır.**)



```
ders : bash
Dosya Düzen Görünüm Geçmiş Yer İmleri Ayarlar Yardım
professor@professor-pardus ders $ xgterm & ds9 &
[1] 6581
[2] 6582
professor@professor-pardus ders $ ls
AUSer Bias Flat login.cl uparm
professor@professor-pardus ders $
```

Şekil 2. ders dizininin alt dizin ve dosyaları

I



```
professor@professor-pardus: ~/IRAF/ders
The following commands or packages are currently defined:

dataio.  images.  lists.   obsolete. proto.   system.
dbms.    language. noao.    plot.    softtools. utilities.

ecl> noao
artdata.  digiphot.  nobsolete.  onedspec.
astcat.   focas.     nproto.    rv.
astrometry. imred.     observatory surfphot.
astutil.  mtlocal.  obsutil.   twodspec.

noao> imred
argus.    crutil.    echelle.   iids.     kpnoconde.  specred.
bias.     ctioslit.  generic.   irred.    kpnoslit.  vtel.
ccdred.   dtol.     hydra.     irs.      quadred.

imred> ccdred
badpiximage  ccdmask      flatcombine  mkskyflat
cdggroups    ccdproc      mkfringeor   setinstrument
cdhedit      ccdtest      mkillumcor   zerocombine
cdinstrument combine       mkillumflat
cdlist       darkcombine  mkskycor

ccdred>
```

Şekil 3. xgterm penceresinden bir görünüm

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

xgterm penceresinde komut satırında sırasıyla “*noao*” , “*imred*” , “*ccred*” komutu verildikten sonra kullanacağımız paketlerin listesi yukarıda verilmiştir. Kırımızı renkte olanlar şu an kullanacağımız taskı göstermektedir.

5. “*cd Bias*” komutu ile Bias verilerinin içerisine girelim. Sonra “ls” komutu ile Bias verilerimizi görelim.

```
ccdred> ls
Bias-001Bias.fit Bias-004Bias.fit Bias-007Bias.fit
Bias-002Bias.fit Bias-005Bias.fit Bias-008Bias.fit
```

6. “*imstat B*.fit*” komutu yardımıyla *zerocombine* taskını kullanmadan önce görüntü istatistiklerine bakalım;

```
ccdred> imstat B*.fit
#      IMAGE      NPIX      MEAN      STDDEV      MIN      MAX
Bias-001Bias.fit 1048576 3295.  10.67  3228.  3370.
Bias-002Bias.fit 1048576 3295.  10.65  3231.  3380.
Bias-003Bias.fit 1048576 3295.  10.64  3232.  3380.
Bias-004Bias.fit 1048576 3295.  10.65  3225.  3387.
Bias-005Bias.fit 1048576 3295.  10.65  3231.  3389.
Bias-006Bias.fit 1048576 3295.  10.64  3223.  3387.
Bias-007Bias.fit 1048576 3295.  10.63  3228.  3382.
Bias-008Bias.fit 1048576 3295.  10.65  3227.  3365.
```

Burada *imstat* taskı ile Bias görüntüleri üzerinde inceleme yaptık. Buradaki *MEAN* değerlerine dikkat edelim ve not edelim: **3295**.

Şimdi *zerocombine* taskını kullanalım;

7. ccdred> epar zerocombine

```
professor@professor-pardus:~/IRAF/ders
IRAF
Image Reduction and Analysis Facility
PACKAGE = ccdred
TASK = zerocombine

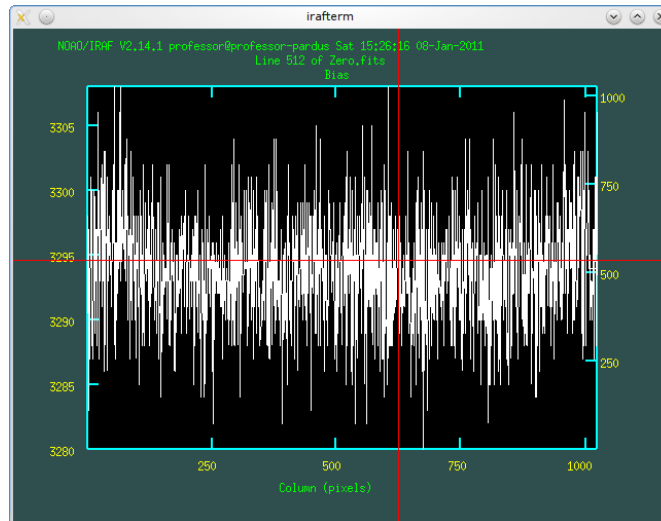
input = B*.fit List of zero level images to combine
(output = Zero) Output zero level name
(combine= median) Type of combine operation
(reject = minmax) Type of rejection
(ccdtype= ) CCD image type to combine
(process= no) Process images before combining?
(delete = no) Delete input images after combining?
(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= 0.) ccdclip: CCD readout noise (electrons)
(gain = 1.) 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)
```

Şekil 4. zerocombine

Bu işlem Şekil 4'de ki gibi yapılmıştır. “:go” komutu ile *Zero.fits* adı ile *masterzero*'yu da oluşturduk;

Implot'la oluşan *Zero.fits* görüntüsünü imstat la gördüğümüz görüntülerin ortalama değerlerine yakın bir trend elde ettiğini görmüş oluruz.

8. ccdred> implot Zero.fits



Şekil 5. implot taskı ile Zero.fits görüntüsünü inceleme

Sonraki adım **Darkcombine**;

Esasında bu adım(**darkcombine**) ve bundan sonra yapacağımız **flatcombine** adımları birbirinin benzeri. Bunun için ekran görüntüsü yerine çıktılar anlatılacaktır.

```
9.    ccdred> imstat Dark*B.fit
#      IMAGE      NPIX    MEAN  STDDEV   MIN    MAX
Dark-001DarkB.fit 1048576  3351.  29.45  3211. 18003.
Dark-002DarkB.fit 1048576  3351.  29.35  3260. 17903.
Dark-003DarkB.fit 1048576  3351.  29.2   3260. 17909.
Dark-004DarkB.fit 1048576  3350.  28.96  3261. 17797.
Dark-005DarkB.fit 1048576  3350.  28.75  3256. 17539
```

Dark görüntüleri bilimsel görüntülerin üzerinde, verildiği poz süresine göre kara akım hatalarını gidermek için kullanılan görüntüler olduğu için ve her filtrede de farklı poz süresi kullanıldığı için bu gözlemde poz süresi yerine onu temsil eden band isimleri ile adlandırılmış. Bu yüzden **darkcombine** taskını her band yani ayrı poz süresi için ayrı ayrı uygulamak gerek. Bu adıma başlarken yine her ayrı **dark görüntüsü** için **imstat** taskı ile istatistiksel değerler incelenebilir.

Darkcombine ile her bant için alınmış ayrı poz sürelerine göre DarkB, DarkV ve DarkR görüntülerini oluşturalım. Bunun için Dark/ dizini içerisine gidelim;

```
ccdred> cd home
ccdred> ls
AUSer Bias Flat login.cl uparm
ccdred> cd AUSer/Dark/
ccdred> ls
Dark-001DarkB.fit Dark-002DarkR.fit Dark-003DarkV.fit Dark-005DarkB.fit
Dark-001DarkR.fit Dark-002DarkV.fit Dark-004DarkB.fit Dark-005DarkR.fit
Dark-001DarkV.fit Dark-003DarkB.fit Dark-004DarkR.fit Dark-005DarkV.fit
Dark-002DarkB.fit Dark-003DarkR.fit Dark-004DarkV.fit
```

```
ccdred> epar darkcombine
```

```
professor@professor-pardus:~/IRAF/ders
IRAF
Image Reduction and Analysis Facility
PACKAGE = ccdred
TASK = darkcombine

input =          Dark*B.fit  List of dark images to combine
(output =        DarkB) Output dark image root name
(combine=        median) Type of combine operation
(reject =        minmax) Type of rejection
(ccdtype=        ) CCD image type to combine
(process=        yes) Process images before combining?
(delete =        no) Delete input images after combining?
(clobber=        no) Clobber existing output image?
(scale =         exposure) 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=        0.) ccdclip: CCD readout noise (electrons)
(gain =          1.) 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
More
ccdred>
```

Şekil 6. *darkcombine* taskı ile *Dark?.fits* görüntüsünü oluşturma

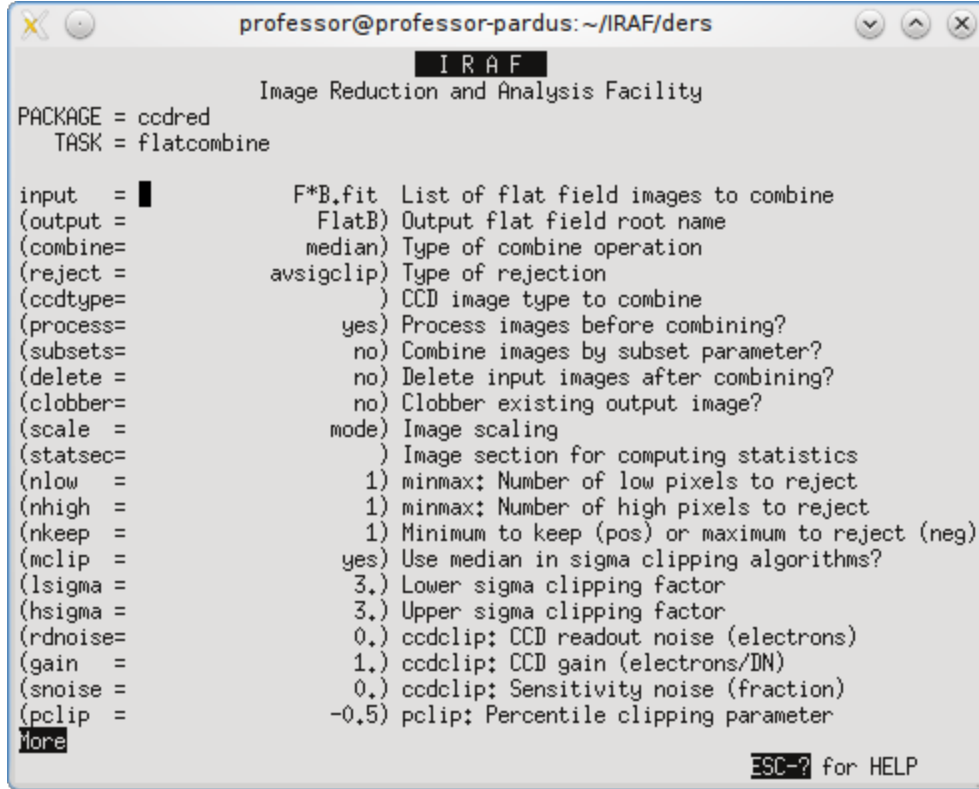
Şekil 6.'da bilerek *Dark?.fits* ifadesi kullanılmıştır. Çünkü diğer bandlara göre de girdi görüntüleri ve çıktı görüntüleri aynı şekilde oluşturulmalıdır. Bu şekilde *DarkV* ve *DarkR* görüntüleri de oluşturulur.

10. ***flatcombine*** taskı kullanarak *FlatB,FlatV,FlatR* görüntülerini oluşturalım;

Flat görüntüleri de *darkcombine* taskında olduğu uygulanacaktır.

```
ccdred> cd home
ccdred> ls
AUSer Bias Flat login.cl uparm
ccdred> cd Flat/
ccdred> ls
Flat-001B.fit Flat-002B.fit Flat-003B.fit Flat-004B.fit Flat-005B.fit
Flat-001I.fit Flat-002I.fit Flat-003I.fit Flat-004I.fit Flat-005I.fit
Flat-001R.fit Flat-002R.fit Flat-003R.fit Flat-004R.fit Flat-005R.fit
Flat-001U.fit Flat-002U.fit Flat-003U.fit Flat-004U.fit Flat-005U.fit
Flat-001V.fit Flat-002V.fit Flat-003V.fit Flat-004V.fit Flat-005V.fit
```

ccdred> epar flatcombine



```
professor@professor-pardus:~/IRAF/ders
IRAF
Image Reduction and Analysis Facility
PACKAGE = ccdred
TASK = flatcombine

input = F*B.fits List of flat field images to combine
(output = FlatB) Output flat field root name
(combine = median) Type of combine operation
(reject = avsigclip) Type of rejection
(ccdtype = ) CCD image type to combine
(process = yes) 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 = 0.) ccdclip: CCD readout noise (electrons)
(gain = 1.) ccdclip: CCD gain (electrons/DN)
(snoise = 0.) ccdclip: Sensitivity noise (fraction)
(pclip = -0.5) pclip: Percentile clipping parameter
More
ESC-? for HELP
```

Şekil 7. darkcombine taskı ile Flat?.fits görüntüsünü oluşturma

Bu şekilde FlatV, FlatR görüntülerini de oluşturalım

11. Şimdi oluşturduğumuz Zero.fits, FlatB, FlatV, FlatR, DarkB, DarkV, DarkR görüntülerini bilimsel verilerle aynı yere kopyalayalım. Bunun için imcopy komutunu kullanacağız;

```
ccdred> cd home
ccdred> imcopy Flat/Flat?.fits AUSER/
Flat/FlatB.fits -> AUSER/FlatB.fits
Flat/FlatR.fits -> AUSER/FlatR.fits
Flat/FlatV.fits -> AUSER/FlatV.fits
ccdred> imcopy AUSER/Dark/Dark?.fits AUSER/
AUSER/Dark/DarkB.fits -> AUSER/DarkB.fits
AUSER/Dark/DarkR.fits -> AUSER/DarkR.fits
AUSER/Dark/DarkV.fits -> AUSER/DarkV.fits
ccdred> imcopy Bias/Zero.fits AUSER/
Bias/Zero.fits -> AUSER/Zero.fits
```

Şimdi ihtiyacımız olan tüm veriler tek bir dizinin içerisinde yani **AUSER** içerisinde. Artık ccdproc taskı ile ön indirgememize geçebiliriz... Tabii “**cd AUSER**” komurunu verdikten sonra=)

11. “*epar ccdproc*” ile girdi dosyalarımızı tüm filtrelere göre şu şekilde yapalım; *AUSer*B.fit*, *AUSer*V.fit*, *AUSer*R.fit* . Çıktılar ise sırasıyla *zdf_//AUSer*B.fit*, *zdf_//AUSer*V.fit*, *zdf_//AUSer*R.fit* şeklinde yazılarak ve *zerocor,darkcor, flatcor* a Zero.fits, Dark?.fits, Flat?.fits yazılarak ön indirgenmiş verileri elde edelim.


```
professor@professor-pardus:~/IRAF/ders
IRAF
Image Reduction and Analysis Facility
PACKAGE = ccdred
TASK = ccdproc

images =          AUser*B.fit  List of CCD images to correct
(output =        zdf_//AUser*B.fit) 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  =        no) Trim the image?
(zero  =        yes) Apply zero level correction?
(dark  =        yes) Apply dark count correction?
(flat  =        yes) Apply flat field correction?
(illum =        no) Apply illumination correction?
(fringe=       no) Apply fringe correction?
(read  =        no) Convert zero level image to readout correction?
(scan  =        no) Convert flat field image to scan correction?

(readaxi=        line) Read out axis (column/line)
(fixfile=        ) File describing the bad lines and columns
(biassec=        ) Overscan strip image section
(trimsec=        ) Trim data section
(zero  =        Zero.fits) Zero level calibration image
(dark  =        DarkB.fits) Dark count calibration image
(flat  =        FlatB.fits) 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?
(funcio=        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)

ESC-? for HELP
```

Şekil 8. ccdproc taskı ile ön indirgeme işlemi

Not(*): Burada her işlemin tekrar anlatılması gereksiz olduğundan sadece B bandı üzerinden anlatım yapılmaktadır. Diğer bandları da sırasıyla görüntüler de ki gibi yapınız.

Not: Esasında ccdproc işlemi, setinstrument taskı yardımı ile bütün görüntülerin görüntü başlıklarına, “**IMAGETYP**” ve “**SUBSET**” düzenlemesi yapılarak, tek adımda ön indirgeme tamamlanabilir.

12. Bütün bandlar'a Dark, Bias ve Flat düzeltmesi yukarıda anlatılan şekilde yapıldıktan sonra indirgenmiş görüntülerimizden bir tanesinin başlığını inceleyelim. Bunun için imhead taskını kullanacağız;

```
ccdred> imhead zdf_AUSer-022R.fit l+
```

```
...
```

```
ZEROCOR = 'Jan 8 16:21 Zero level correction image is Zero.fits'
```

```
DARKCOR = 'Jan 8 16:21 Dark count correction image is DarkR.fits with scale=1.'
```

```
FLATCOR = 'Jan 8 16:21 Flat field image is FlatR.fits with scale=15007.67'
```

```
CCDSEC = '[1:1024,1:1024]'
```

```
CCDMEAN = 85.97369
```

```
CCDMEANT= 978970877
```

```
CCDPROC = 'Jan 8 16:21 CCD processing done'
```

Dikkat edilirse başlığın en sonunda bir takım anahtarlar eklenmiş durumda. Bunlar IRAF'ın görüntü üzerinde bize ne gibi işlemler yaptığını belirten anahtarlar. Bu anahtarlar sayesinde IRAF bu anahtarlar sayesinde aynı işlem tekrar edilemez. Bu IRAF'ın bize sunduğu iyi bir özellik.

13. *ccdproc*'tan sonra hava kütlesi hesabımız için bizim de bir takım başlıklar eklememiz ve düzenlememiz gerekecek;

Öncelikle *ccdlist* taskının görüntülerimizi şu şekilde göstermesinden ben hoşlanmadığım için IRAF'ın tanıyacağı şekile getirmek için şu işlemleri yaptım;

```
ccdred> ccdlist zdf*.fit
```

```
...
```

```
zdf_AUSer-110B.fit[1024,1024][real][unknown][][ZDF]:AU_Ser
```

```
zdf_AUSer-110R.fit[1024,1024][real][unknown][][ZDF]:AU_Ser
```

```
zdf_AUSer-110V.fit[1024,1024][real][unknown][][ZDF]:AU_Ser
```

Bunun manası görüntüler üzerinde *Zero, Dark, Flat* düzeltmesi yapılmış ama; IRAF, görüntülerin *IMAGETYP*'ı düzgün olmadığından ne görüntüsü olduğunu algılamıyor. Ayrıca Filtre'yi de başlıklardan algılayamıyor. Eğer bu düzenlemeyi en başta yapsa idik Not(*)'da anlatılan adımı direk yapabiliirdik. Fakat temel aşamada indirgeme yapmayı amaçladığımız için yapılmadı. Hadi başlıkları düzeltelim;

```
ccdred> hedit zdf_*.fit IMAGETYP 'object' ver- show+ upd+ add+
```

```
...
```

```
zdf_AUSer-110V.fit,IMAGETYP: LIGHT -> object
```

```
zdf_AUSer-110V.fit updated
```

```
ccdred> hedit Flat*.fits IMAGETYP 'flat' ver- show+ upd+ add+
FlatB.fits,IMAGETYP: LIGHT -> flat
FlatB.fits updated
FlatR.fits,IMAGETYP: LIGHT -> flat
FlatR.fits updated
FlatV.fits,IMAGETYP: LIGHT -> flat
FlatV.fits updated
```

```
ccdred> hedit Zero.fits IMAGETYP 'zero' ver- show+ upd+ add+
Zero.fits,IMAGETYP: BIAS -> zero
Zero.fits updated
```

Dark görüntülerinin IMAGETYP'ı düzgün dokunmaya gerek yok:)

Başlıklardan *FILTER* değerini alıp *SUBSET* olarak ekleyelim;

```
ccdred> hedit zdf_*.fit SUBSET '("@FILTER")' add+ ver- show+ upd+
...
add zdf_AUSer-110V.fit,SUBSET = V
zdf_AUSer-110V.fit updated
```

Şimdi *ccdlis*t ile tekrar bakalım;

```
ccdred> ccdlis zdf*.fit
...
zdf_AUSer-110B.fit[1024,1024][real][object][B][ZDF]:AU_Ser
zdf_AUSer-110R.fit[1024,1024][real][object][R][ZDF]:AU_Ser
zdf_AUSer-110V.fit[1024,1024][real][object][V][ZDF]:AU_Ser
```

Görüldüğü gibi artık hiç bir sıkıntı yok IRAF görüntülerimizin ne olduğunu ve hangi filtrede olduğunu artık biliyor:)

14. Birazcık oyalanmadan sonra hava kütlesi hesabı(*setairmass(asutil)* paketi içerisinde)) yapabilmemiz için gereken anahtarlarında görüntü başlığına eklememiz gerekli. Bunlar şunlar olacak;

Gözlemevi bilgisine ve bunun yanında **RA**, **DEC**, **EPOCH** ve **ST** ve **DATE-OBS**, **JD** (Bu bilgi ve **HJD**, **LJD** gibi bilgiler *astutil* paketi içerisinde *setjd* taskı ile hesaplanacak) bilgilerine ihtiyacımız var. Gözlemevi bilgisini "*iraf/iraf/noao/lib/obsdb.dat*" isimli dosyanın içerisine kendi gözlemevimizin bilgisini ekleyeceğiz. (Ya da verilerin alındığı gözlemevi bilgisi eklenmeli.). **EPOCH**, **RA** ve **DEC** ise simbad güncel olarak alınarak görüntü başlıklarına eklenmeli.

İlk olarak gözlemevi bilgisini *obsdb.dat* dosyasının içerisine eklemekle başlayalım;

```
#AUG TURKEY/ANKARA
observatory = "AUG"
  name="Ankara University Observatory"
  longitude = -32:46:45
  latitude = 39:50:57
  altitude = 1256.69
  timezone = -2
```

AUG'nin sitesinden aldığımız verileri düzenleyerek “/iraf/iraf/noao/lib/obsdb.dat” dosyasının sonuna ekliyoruz.

Daha sonra görüntü başlıklarına;

```
ccdred> hedit zdf_*.fit CEXPTIME '(EXPTIME+0.004)' ver- show+ add+
```

Komutunu vererek shutter'ın kapanmasından kaynaklı 0.004 sn lik zaman düzeltmesini CEXPTIME anahtarı ile ekliyoruz.(astutil komutu ile astutil paketine gidelim.)

```
astutil> hedit zdf_*.fit OBSERVAT 'AUG' ver- show+ add+
```

```
...
zdf_AUSer-110R.fit,OBSERVAT: AUG -> AUG
zdf_AUSer-110R.fit updated
zdf_AUSer-110V.fit,OBSERVAT: AUG -> AUG
zdf_AUSer-110V.fit updated
```

Komutu ile tüm görüntü başlıklarına “OBSERVAT” anahtarını “AUG” değeri ile ekliyoruz.

Sağ açıklık(RA) ve Dik Açıklık(DEC) ve EPOCH değerlerini Simbad'dan güncel olarak alıp başlığa ekliyoruz.

```
astutil> hedit zdf_*.fit RA '15:56:49.4' ver- show+ add+ upd+
```

```
...
zdf_AUSer-110B.fit updated
add zdf_AUSer-110R.fit,RA = 15.94706
zdf_AUSer-110R.fit updated
add zdf_AUSer-110V.fit,RA = 15.94706
zdf_AUSer-110V.fit updated
```

```
astutil> hedit zdf_*.fit DEC '22:16:01' ver- show+ add+ upd+
```

```
...
add zdf_AUSer-110R.fit,DEC = 22.26694
zdf_AUSer-110R.fit updated
add zdf_AUSer-110V.fit,DEC = 22.26694
zdf_AUSer-110V.fit updated
```

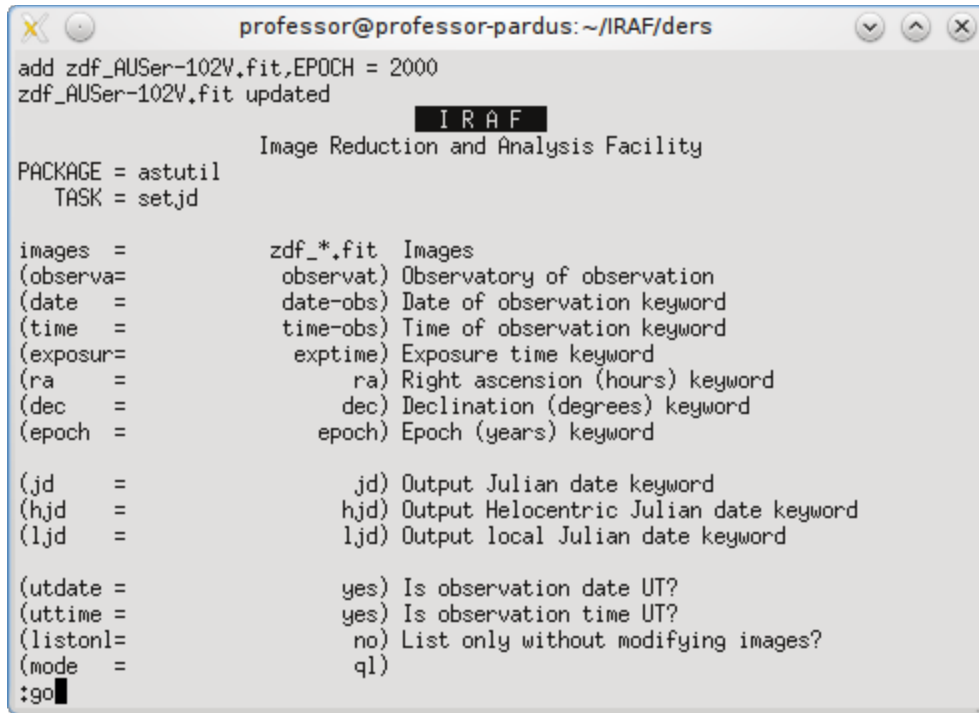
```
astutil> hedit zdf_*.fit EPOCH '2000' ver- show+ upd+ add+
```

```
...  
zdf_AUSer-110B.fit updated  
add zdf_AUSer-110R.fit,EPOCH = 2000  
zdf_AUSer-110R.fit updated  
add zdf_AUSer-110V.fit,EPOCH = 2000  
zdf_AUSer-110V.fit updated
```

RA,DEC ,OBSERVAT, EPOCH anahtarları ve değerleri eklendi. Şimdi görüntü başlıklarına JD,HJD,LJD yi ekleyelim bunun için "**setjd**" komutunu kullanacağız.

```
astutil> epar setjd
```

komutunu **Şekil 9**'daki gibi veriniz.



```
professor@professor-pardus:~/IRAF/ders  
add zdf_AUSer-102V.fit,EPOCH = 2000  
zdf_AUSer-102V.fit updated  
IRAF  
Image Reduction and Analysis Facility  
PACKAGE = astutil  
TASK = setjd  
  
images =          zdf_*.fit  Images  
(observa=        observat) Observatory of observation  
(date =          date-obs) Date of observation keyword  
(time =          time-obs) Time of observation keyword  
(exposur=        exptime) Exposure time keyword  
(ra =            ra) Right ascension (hours) keyword  
(dec =           dec) Declination (degrees) keyword  
(epoch =         epoch) Epoch (years) keyword  
  
(jd =            jd) Output Julian date keyword  
(hjd =           hjd) Output Heliocentric Julian date keyword  
(ljd =           ljd) Output local Julian date keyword  
  
(utdate =        yes) Is observation date UT?  
(uttime =        yes) Is observation time UT?  
(listonl=        no) List only without modifying images?  
(mode =          ql)  
:go
```

Şekil 9. setjd taskı

```
...  
zdf_AUSer-109R.fit 2454977.46603 2454977.47027 2454977  
zdf_AUSer-109V.fit 2454977.46578 2454977.47002 2454977  
zdf_AUSer-110B.fit 2454977.46638 2454977.47062 2454977  
zdf_AUSer-110R.fit 2454977.46700 2454977.47124 2454977  
zdf_AUSer-110V.fit 2454977.46675 2454977.47099 2454977
```

15. Görüntü başlıklarına Yıldız Zamanını ST anahtarı olarak eklemek için ise *astutil paketi* içerisinde *st.cl* dosyası oluşturulur. İçeriği şöyledir;

```
astutil> !kwrite st.cl (kwrite sizde olmayabilir gedit, vi ya da nano'yu deneyin.)  
astcalc  
    observatory = "AUG"  
    st = mst('@DATE-OBS', '@TIME-OBS', obsdb(observatory, "longitude"))  
quit
```

Editöre yukarıdaki betik yazıldıktan sonra kaydedilip çıkarılır.

Sonra bütün görüntülere ST başlığını eklemek için;

```
astutil> asthedit zdf_AUSer*.fit st.cl ver+ upd+  
...  
zdf_AUSer-110B.fit:  
    $I = zdf_AUSer-110B.fit  
    observat = AUG -> AUG  
    st = 17:37:05.85  
zdf_AUSer-110R.fit:  
    $I = zdf_AUSer-110R.fit  
    observat = AUG -> AUG  
    st = 17:38:10.03  
zdf_AUSer-110V.fit:  
    $I = zdf_AUSer-110V.fit  
    observat = AUG -> AUG  
    st = 17:37:42.96
```

Evet uzun bir başlık düzenleme işleminden sonra setairmass takını kullanabilir duruma geldik. Ama yine de başlamadan önce eklediğimiz anahtarlara bir bakalım değil mi?:)

```
astutil> imhead zdf_AUSer-110R.fit l+  
SUBSET = 'R      '  
CEXPTIME= '10.004 '  
RA    =      15.94706  
DEC   =      22.26694  
EPOCH =      2000  
JD    = 2454977.46700232  
HJD   = 2454977.47124298  
LJD   =      2454977.  
OBSERVAT= 'AUG  '  
ST    = '17:38:10.03'
```

16. Şimdi "*setairmass*" taskını kullanabilir duruma geldik. Etkin hava kütlesi *astutil>setairmass setairmass* taskının gerekli parametrelerini "*lpar setairmass*" ile görelim;

astutil> lpar setairmass

```
professor@professor-pardus:~/IRAF/ders
images =          Input images
(observatory = )_observatory) Observatory for images
(intype = "beginning") Input keyword time stamp
(outtype = "effective") Output airmass time stamp\n
(ra = "ra") Right ascension 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
(exposure = "exptime") Exposure time keyword (seconds)
(airmass = "airmass") Airmass keyword (output)
(utmiddle = "utmiddle") Mid-observation UT keyword (output)
(scale = 750.) The atmospheric scale height\n
(show = yes) Print the airmasses and mid-UT?
(update = yes) Update the image header?
(override = yes) Override previous assignments?
(mode = "ql")
astutil> █
```

Şekil 10. lpar setairmass taskı

Burada gerekli parametreleri zaten oluşturduk fakat bu task bizim için önemli olarak şu anahtarları ve değerlerini çıktı olarak başlığa ekleyecektir. Bunlar "**airmass**" ve "**utmiddle**" dir.

setairmass taskını çalıştıralım.

astutil> epar setairmass

```
professor@professor-pardus:~/IRAF/ders
IRAF
Image Reduction and Analysis Facility
PACKAGE = astutil
TASK = setairmass

images =          zdf_AUSe*.fit Input images
(observat=        observat) Observatory for images
(intype =         middle) Input keyword time stamp
(outtype=        effective) Output airmass time stamp

(ra =            ra) Right ascension keyword (hours)
(dec =            dec) Declination keyword (degrees)
(equinox=        epoch) Equinox keyword (years)
(st =            st) Local siderial time keyword (hours)
(ut =            time-obs) Universal time keyword (hours)
(date =          date-obs) Observation date keyword
(exposur=        exptime) Exposure time keyword (seconds)
(airmass=        airmass) Airmass keyword (output)
(utmiddle=       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?
(override=       yes) Override previous assignments?
(mode =          ql)

:go █
```

Şekil 11. setairmass taskı

```
...
zdf_AUSer-109B.fit 23:09:55.0 1.1252 1.1248 1.1252 1.1256 yes
zdf_AUSer-109R.fit 23:11:00.0 1.1270 1.1268 1.1270 1.1271 yes
zdf_AUSer-109V.fit 23:10:33.0 1.1262 1.1259 1.1262 1.1265 yes
zdf_AUSer-110B.fit 23:11:20.0 1.1275 1.1271 1.1275 1.1279 yes
zdf_AUSer-110R.fit 23:12:24.0 1.1293 1.1292 1.1293 1.1295 yes
zdf_AUSer-110V.fit 23:11:57.0 1.1286 1.1283 1.1286 1.1288 yes
```

Çıktı sonucu oluşan airmass ve utmiddle anahtarları ve değerleri;

```
astutil> imhead zdf_AUSer-110R.fit l+
```

```
...
AIRMASS = 1.129312
UTMIDDLE= '2009-05-25T23:12:24.00'
```

Artık ön indirgemeyi tamamen bitirdik.

17. Şimdi görüntülerimizi hizalamaya geçelim.

Öncelikle “epar display” diyerek ds9'un görüntüleme ayarını düzenleyelim “fill=no” olan ayarı “yes” olarak değiştirelim(:q ile çıkalım);

Şimdi ise imexamine ile bandlara göre tüm görüntülerin koordinatlarını bir dosyanın içerisine yazacağız. Şöyle ki;

```
apphot> imexamine zdf_AUSer*B.fit 1 > corB
```

Karşımıza gelecek olarak ds9 görüntülerinde “x” tuşuna basarak referans yıldızın görüntü üzerindeki (x,y) koordinatlarını ve sayım değerini bu komut sayesinde “**refcoorB**” isimli bir dosyaya kaydediyoruz. “n” tuşuna basarakta bir diğer görüntüye geçiyoruz. Aynı şekilde **refcoorV** ve **refcoorR** yi de elde ediyoruz. Yalnız bu aşamada çok hassas olmamız ve hiç bir görüntüyü atlamamız gerekiyor.;

```
apphot> imexamine zdf_AUSer*V.fit 1 > corV
```

```
apphot> imexamine zdf_AUSer*R.fit 1 > corR
```

18. Bu aşamada ise oluşturduğumuz “cor?” dosyalarından sadece (x,y) koordinatlarını almamız gerekli;

Bunu da unix'in güçlü bir komutu olan “**awk**” sayesinde yapacağız.


```
apphot> !cat corB | grep -v 'z' | awk '{print $1,$2}' > refcoorB
```

```
apphot> !cat corV | grep -v 'z' | awk '{print $1,$2}' > refcoorV
```

```
apphot> !cat corR | grep -v 'z' | awk '{print $1,$2}' > refcoorR
```

```
apphot> coun refcoorB
```

```
110 220 1540 refcoorB
```

```
apphot> coun refcoorV
```

```
110 220 1540 refcoorV
```

```
apphot> coun refcoorR
```

```
110 220 1540 refcoorR
```

Herhangi bir dosyanın içerisine bakacak olursak;

```
apphot> type refcoorB
```

```
...
```

```
333.00 708.00
```

```
332.00 718.00
```

```
335.00 703.00
```

```
333.00 727.00
```

Artık bütün görüntülerimde referansımızın koordinatları(X,Y) elimizde. Şimdi ilk görüntüye göre diğer görüntüler ne kadar kayma göstermiş onu hesaplayalım ve bir dosyaya kaydedelim. Çünkü **imshift** taskı görüntüleri ne kadar kaydıracağını o dosya sayesinde bilecek.

```
apphot> !awk '{print 340.0-$1,615.0-$2}' refcoorB > shiftB
```

```
apphot> !awk '{print 340.0-$1,615.0-$2}' refcoorV > shiftV
```

```
apphot> !awk '{print 340.0-$1,615.0-$2}' refcoorR > shiftR
```

Herhangi bir dosyanın içerisine bakacak olursak;

```
apphot> type shiftV
```

```
...
```

```
3 -112
```

```
1 -107
```

```
2 -100
```

```
3 -124
```

Çıktıda ilk görüntüye göre diğer görüntünün ne kadar kaymış olduğunu görüyorsunuz.

19. **imshift** ile align etme;

lpar ile **imshift**'in parametrelerine bakacak olursak;

```
apphot> lpar imshift
```

input = Input images to be fit
output = Output images
xshift = Fractional pixel shift in x
yshift = Fractional pixel shift in y
(shifts_file =) Text file containing shifts for each image
(interp_type = "linear") Interpolant (nearest,linear,poly3,poly5,spline3)
(boundary_typ = "nearest") Boundary (constant,nearest,reflect,wrap)
(constant = 0.) Constant for boundary extension
(mode = "ql")

Klasik girdi ve çıktı verilerinin yanı sıra kaydırma dosyamızı “*shifts_*” ile gösterecekmiz.Şöyle ki;

apphot> imshift zdf_AU*B.fit s//zdf_AU*B.fit shifts_=shiftB

Bu işlem biraz uzun sürecek bekleyeylim;

Diğer bandları da aynı şekilde yapacak olursak;

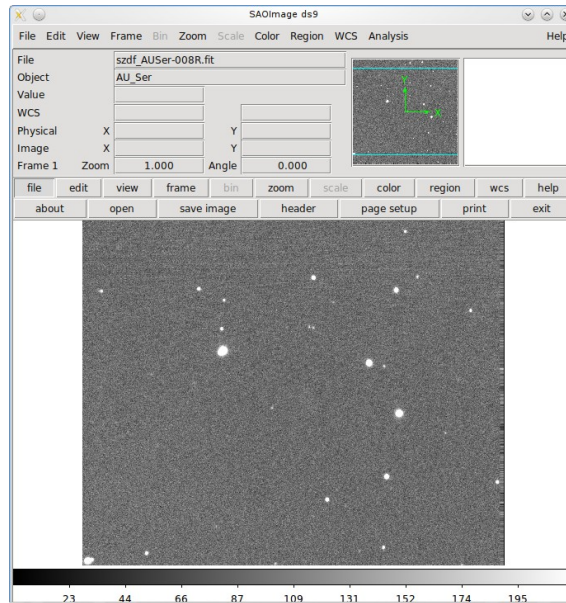
apphot> imshift zdf_AU*V.fit s//zdf_AU*V.fit shifts_=shiftV

apphot> imshift zdf_AU*R.fit s//zdf_AU*R.fit shifts_=shiftR

DS9 ile şimdi yaptığımız hizalamayı kontrol edelim;

apphot> imexamine szdf_AU*B.fit 1

“*n*” tuşuna basarak(bir sonraki görüntü), “*p*” tuşuna basarak(bir önceki görüntü) hizalamamızı kontrol edelim. Bozuk olanları not alalım. Onları *imshift* ile tekrar hizalamamız gerekli. Bütün görüntüleri kontrol ettikten sonra “*q*” tuşu ile çikalım. Bu işlemi diğer bandlar için de yapalım.



Şekil 12. imexamine taskı ve DS9 ile hizalamama kontrolü

*apphot> imexamine szdf_AU*V.fit 1*

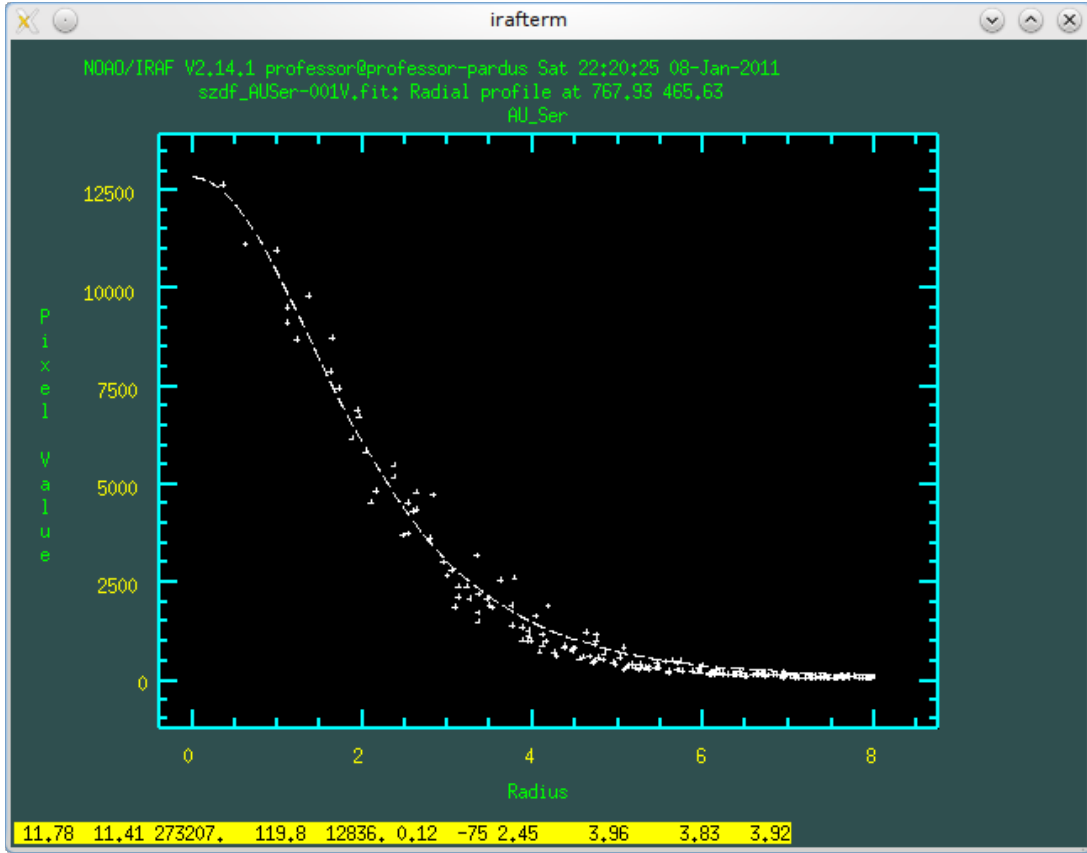
*apphot> imexamine szdf_AU*R.fit 1*

Evet hizalama işlemini de bitirdik. Her şey yolunda... Artık fotometri'ye geçebilmek için bir mani kalmadı:)

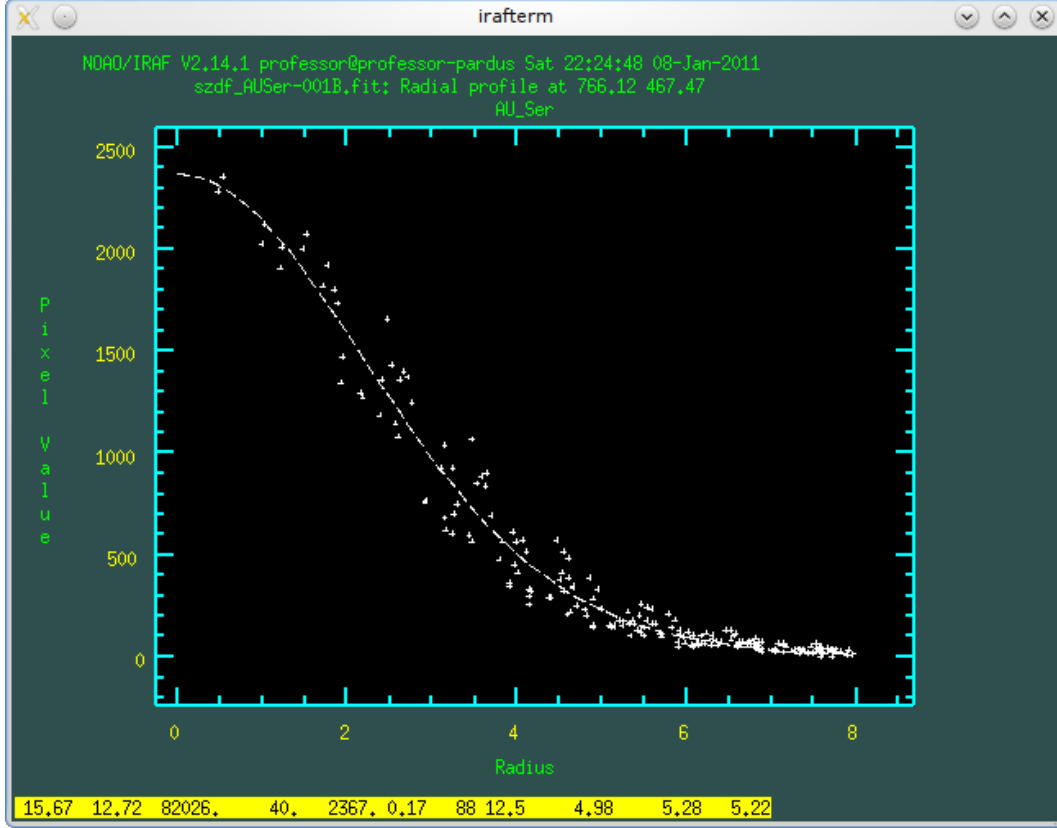
20. Yalnız önce *Cbox*, *Aparture*, *Annulus* ve *Dannulus* değerlerimizi belirlemek için *FWHM* değerimizi öğrenmemiz gerekli. Bunu belirlemek için de yine imexamine taskından yardım alacağız.

noao> imexamine szdf_AUSer-001B.fit 1

Komutundan sonra ds9 penceresinde “r” tuşuna basarak değişenin radyal profilini inceleyiniz. V bandında 3.83 gibi bir *FWHM* değerimiz var. B de ise 5.83 çıktı. R' de 3.79 ... O halde ortalama olarak *FWHM* değerimizin “4.4833” olduğunu düşünürsek;



Şekil 13. imexamine taskı ve DS9 ile radyal profil inceleme



Şekil 14. imexamine taskı ve DS9 ile radyal profil inceleme

Yaklaşık olarak;

$$Cbox = FWHM \times 2 = 10$$

$$Aperture = FWHM \times 4 = 20$$

$$Annulus = 25$$

$$Dannulus = 5$$

alınabilir.

21. Şimdi *noao>digiphot>apphot* paketinin içerisine girip *phot taskını* çalıştırabiliriz.

Fakat bizden ne isteniyor iyice bilelim değil mi?

apphot> lpar phot

Komutu ile neler gerekli imiş kontrol edelim;

```

professor@professor-pardus: ~/IRAF/ders
apphot> lpar phot
  image = ""           The input image(s)
  skyfile = ""        The input sky file(s)
  (coords = "")       The input coordinate files(s) (default: image.c
  (output = "default") The output photometry file(s) (default: image,m
  (plotfile = "")     The output plots metacode file
  (datapars = "")     Data dependent parameters
  (centerpars = "")   Centering parameters
  (fitskypars = "")   Sky fitting parameters
  (photpars = "")     Photometry parameters
  (interactive = yes) Interactive mode ?
  (radplots = no)     Plot the radial profiles in interactive mode ?
  (icommands = "")    Image cursor: [x y wcs] key [cmd]
  (gcommands = "")    Graphics cursor: [x y wcs] key [cmd]
  (wcsin = )_wcsin)   The input coordinate system (logical,tv,physica
  (wcsout = )_wcsout) The output coordinate system (logical,tv,physic
  (cache = )_cache)   Cache the input image pixels in memory ?
  (verify = )_verify) Verify critical parameters in non-interactive m
  (update = )_update) Update critical parameters in non-interactive m
  (verbose = )_verbose) Print messages in non-interactive mode ?
  (graphics = )_graphics) Graphics device
  (display = )_display) Display device
  (mode = "ql")
apphot> █

```

Şekil 15. phot parametreleri

Temel girdi değerleri haricinde dikkatimizi çeken “**coords**” parametresi. Değişen ve mukayesemizin koordinatlarını bu parametreye göstermemiz lazım. Bunun için Simbad'dan değişen ve mukayesemizin hangisi olduğunu bulmamız gerek;

Simbad'dan ;

V* AU Ser -- Eclipsing binary of W UH

Other object types: [WU*](#) () , [V*](#) (V*,AN) , *

ICRS coord. (ep=j2000) : [15 56 49.4 +22 16 01 \(](#)

FK5 coord. (ep=j2000 eq=2000) : [15 56 49.4 +22 16 01 \(](#)

FK4 coord. (ep=B1950 eq=1950) : [15 54 39.1 +22 24 38 \(](#)

Gal coord. (ep=j2000) : [036.8573 +47.8444 \(~ |](#)

Radial velocity / Redshift / cz : [V\(km/s\) -62.90 \[1.50\] ,](#)

Parallax mas: [5.46 \[0.68\] D \[2005MNRAS\]\(#\)](#)

Spectral type: [G5 D ~](#)


Fluxes (1) : [V 10.9 \[-\] V3 E \[2003As\]\(#\)](#)

Identifiers (4) :

[V* AU Ser](#) [AN 48.1935](#) [GSC 01502-01](#)

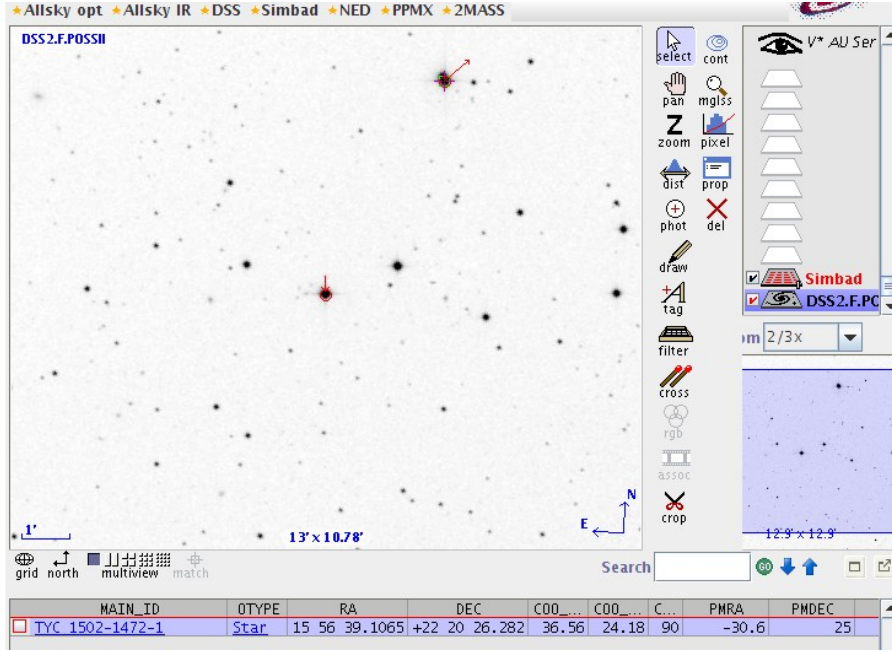
Plots and Images

radius arcmin



Şekil 16. AU Ser ve Mukayese Arayışı

AU Ser arandıktan sonra etrafındaki mukayese olabilecek cisimleri görmek için “*Plots and Images*” bölümünden “*plot around*” ile 20 yay dak lık bir alanı taradık; Sonra *Aladin Applet*'den bilgi aldığımızda gördük ki bölgede “TYC 1502-1472-1” adında mukayese olabilecek bir yıldız bulduk. Bu yıldızın mukayese olduğuna ise “http://cdsads.u-strasbg.fr/cgi-bin/nph-bib_query?2005NewA...10..653G&db_key=AST&nosetcookie=1” linkinde ki yayından karar verdik.



Şekil 17. AU Ser(Ortada) ve Mukayese (Üstte sağda)

Şimdi koordinat dosyamızı oluşturabiliriz. DS9'da Aladin Applet'te ki gibi **WCS** türünde yön tayini yapalım. **Zoom** menüsünden **270 derece** görüntüyü çevirdiğimizde Kuzey üste, Doğu sola gelecek şekilde ayarlanabildi. Diğer türlü görüntüler de Doğu üst, Batı alt tarafa gelecek şekilde ayarlanmış idi. Şimdi **Şekil 17.** deki gibi DS9'la da görüntüleyebiliriz.

apphot> imexamine szdf_AUSer-001?.fit 1

Komurunu vererek önce değişenin üzerine sonra da mukayesenin üzerine gelerek “x” tuşuna basıp sonra “n” tuşuna basarak tüm bandlarda ki birinci görüntülerde bulunan değişen ve mukayesenin koordinatları alınır ve ayrı ayrı **coorB**, **coorV**, **coorR** dosyalarını oluşturulur.

Örnek olarak coorB koordinat dosyasının içeriği şu şekildedir.

apphot> type coorB

765.50 467.50

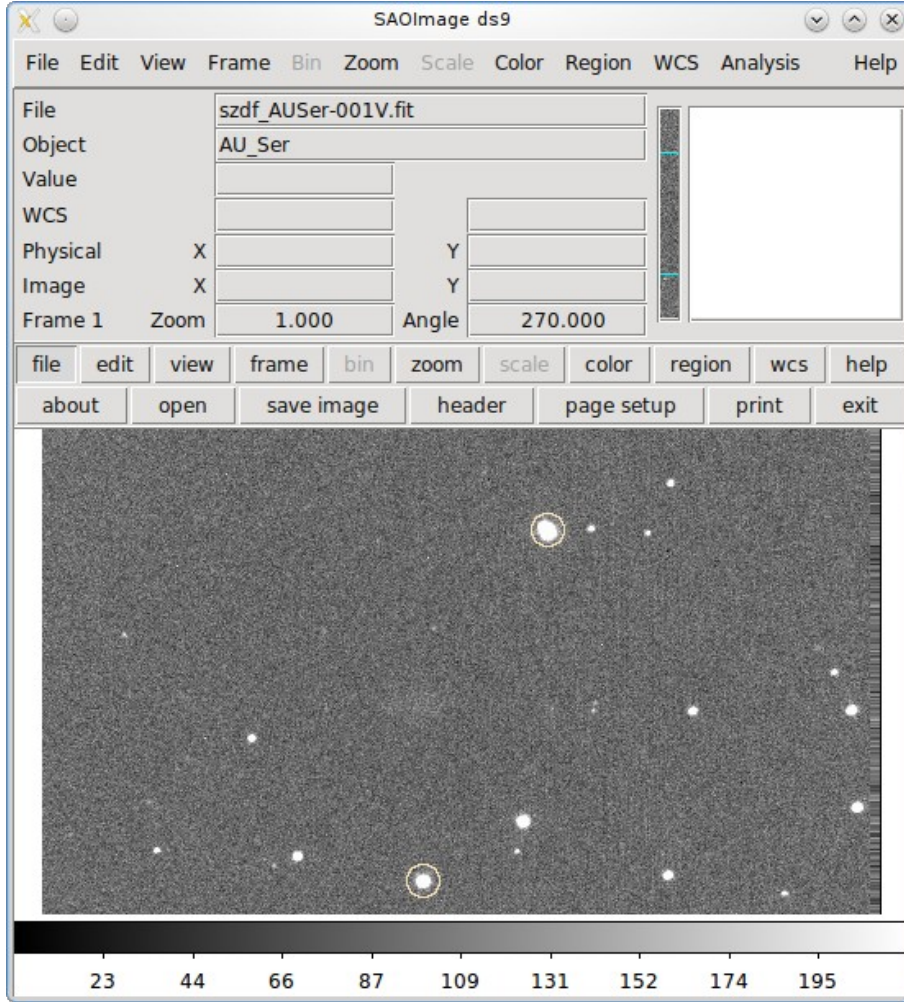
339.50 619.50

Diğerlerinin içerdiği de bu şekilde olmalıdır. 1. satırdaki (x,y) değişimin koordinatları 2. satırdaki (x,y) ise mukayesenin koordinatlarıdır. Şimdi koordinat dosyalarımızın doğruluğunu “*tvmark*” ile kontrol edelim;

```
apphot> display szdf_AUSer-001V.fit 1
```

```
z1=80.35357 z2=172.0281
```

```
apphot> tvmark 1 coorV mark=circle radii="20"
```



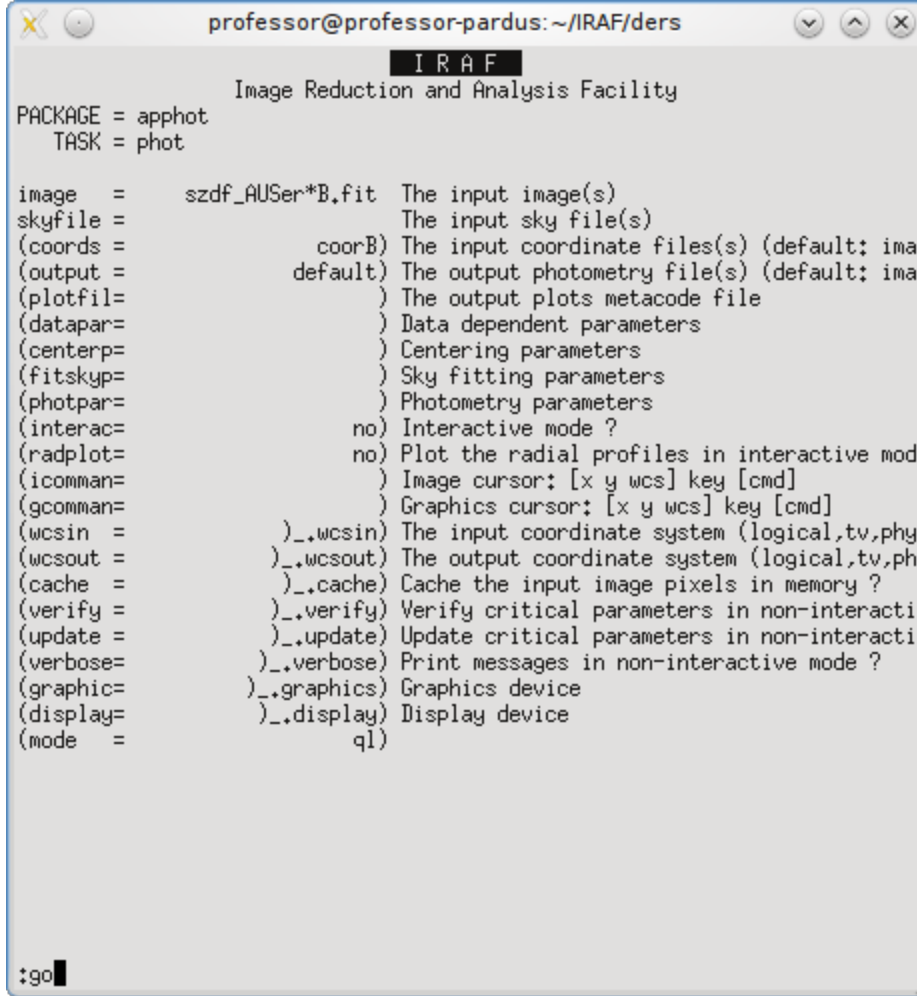
Şekil 18. *tvmark* AU Ser(Altta) ve Mukayese (Üstte)

Fotometre heyecanı sardı bizi değil mi o halde başlayalım:)

22. “*lpar phot*” komutunu daha önceden vermiştik. (Bkz. Şekil 15.)

Şimdi istenilen değerleri girelim;

apphot> epar phot



```
professor@professor-pardus:~/IRAF/ders
IRAF
Image Reduction and Analysis Facility
PACKAGE = apphot
TASK = phot

image =      szdf_AUSe*r*B.fit  The input image(s)
skyfile =    The input sky file(s)
(coords =    coorB) The input coordinate files(s) (default: ima
(output =    default) The output photometry file(s) (default: ima
(plotfil=    ) The output plots metacode file
(datapar=    ) Data dependent parameters
(centerp=    ) Centering parameters
(fitskyp=    ) Sky fitting parameters
(photpar=    ) Photometry parameters
(interac=    no) Interactive mode ?
(radplot=    no) Plot the radial profiles in interactive mod
(icomman=    ) Image cursor: [x y wcs] key [cmd]
(gcomman=    ) Graphics cursor: [x y wcs] key [cmd]
(wcsin =     )_._wcsin) The input coordinate system (logical,tv,phy
(wcsout =    )_._wcsout) The output coordinate system (logical,tv,ph
(cache =     )_._cache) Cache the input image pixels in memory ?
(verify =    )_._verify) Verify critical parameters in non-interacti
(update =    )_._update) Update critical parameters in non-interacti
(verbose=    )_._verbose) Print messages in non-interactive mode ?
(graphic=    )_._graphics) Graphics device
(display=    )_._display) Display device
(mode =     )_._ql)

:go
```

Şekil 19. “phot” taskı.

(Not: Şekilde ki gibi hemen “:go” komutunu vermeyiniz. Yazıyı okuyunuz.)

B bandı için örnek olarak anlatalım diğerleri için ekran görüntüsü verelim.İndirgemeyi yapacağımızı görüntüleri **Şekil 19**'da ki gibi girelim.(**Örneğin:** B bandı için **image=szdf_AUSe*r*B.fit**) Sonra indirgeme yapılan banda göre oluşturulan koordinat dosyamızı gösterelim. Örneğin B bandı için “**coorB**”, eğer V bandı indirgeniyorsa “**coorV**” gibi...

Sonra **datapar** parametresine gelini. “:e” tuşuna basılır ve yeni bir ayar yerine girilir. **Şekil 20**'de ki gibi düzenlenir.;



```
professor@professor-pardus:~/IRAF/ders
IRAF
Image Reduction and Analysis Facility
PACKAGE = apphot
TASK = datapars

(scale = 1.) Image scale in units per pixel
(fwhmpsf= 2.5) FWHM of the PSF in scale units
(emissio= yes) Features are positive ?
(sigma = INDEF) Standard deviation of background in counts
(datamin= INDEF) Minimum good data value
(datamax= INDEF) Maximum good data value
(noise = poisson) Noise model
(ccdread= ) CCD readout noise image header keyword
(gain = ) CCD gain image header keyword
(readnoi= 0.) CCD readout noise in electrons
(epadu = 1.) Gain in electrons per count
(exposur= exptime) Exposure time image header keyword
(airmass= airmass) Airmass image header keyword
(filter = subset) Filter image header keyword
(obstime= hjd) Time of observation image header keyword
(itime = 1.) Exposure time
(xairmas= INDEF) Airmass
(ifilter= INDEF) Filter
(otime = INDEF) Time of observation
(mode = ql)

ESC-? for HELP
```

Şekil 20 apphot “datapar”

Burada **exposure** kısmına görüntü başlığındaki “**exptime**”, **airmass** kısmına başlığındaki “**airmass**”, **filter** kısmına ise başlığındaki “**subset**” anahtarları yazılarak “:q” denilerek çıkarılır.

Sonra **centerpar** parametresine gelinir ve “:e” tuşuna basılır ve **Şekil 21.** deki gibi düzenlenir.

```
professor@professor-pardus:~/IRAF/ders
IRAF
Image Reduction and Analysis Facility
PACKAGE = apphot
TASK = centerpars

(calgori=          centroid) Centering algorithm
(cbox =            10.) Centering box width in scale units
(cthresh=          0.) Centering threshold in sigma above backgrou
(minsnra=          1.) Minimum signal-to-noise ratio for centering
(cmaxite=          10) Maximum number of iterations for centering
(maxshif=          1.) 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)
```

Şekil 21. apphot “centerpar”

Belirlediğimiz **cbox** değeri yazılır ve sonra “:q” diyerek buradan çıkılır.

Sonra “**fitskyp**” a gelinir ve aynı şekilde “:e” tuşuna basılarak Şekil 22. de ki gibi düzenlenir.

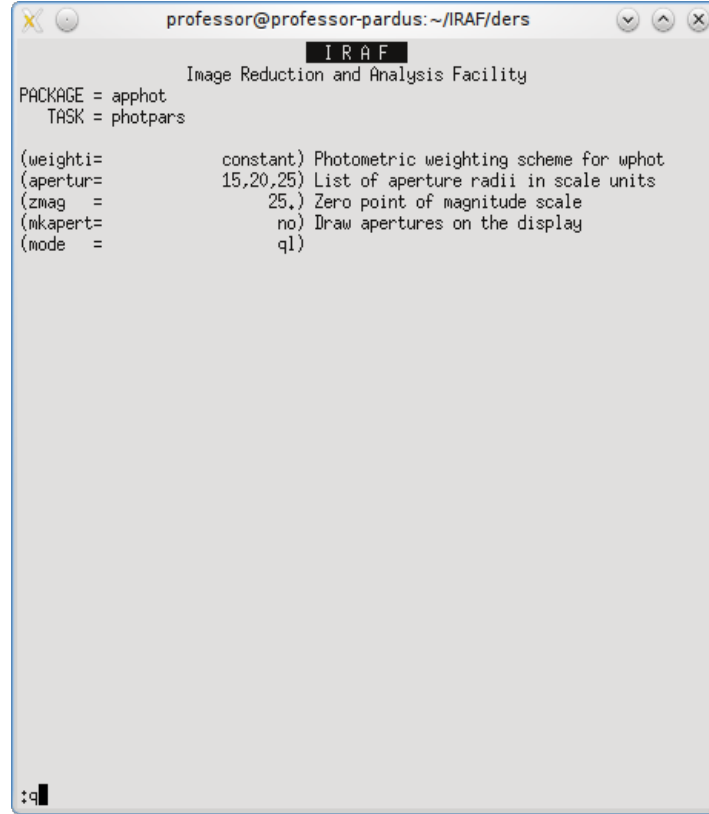
```
professor@professor-pardus:~/IRAF/ders
IRAF
Image Reduction and Analysis Facility
PACKAGE = apphot
TASK = fitskypars

(salgori=          centroid) Sky fitting algorithm
(annulus=          25.) Inner radius of sky annulus in scale units
(dannulu=          5.) 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 ite
(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=          0,1) Binsize of histogram in sky sigma
(smooth =          no) Boxcar smooth the histogram
(rgrow =           0.) Region growing radius in scale units
(mksky =           no) Mark sky annuli on the display
(mode =            ql)
```

Şekil 22. apphot “fitskyp”

Burada “*annulus*” ve “*dannulus*” değerleri yazılır “:q” diyerek buradan da çıkılır.

Şimdi “*photpar*” parametresine gelinir “:e” tuşuna basılır ve Şekil 23. deki gibi düzenlenir.



```
professor@professor-pardus:~/IRAF/ders
IRAF
Image Reduction and Analysis Facility
PACKAGE = apphot
TASK = photpars
(weight=      constant) Photometric weighting scheme for wphot
(apertur=    15,20,25) List of aperture radii in scale units
(zmag =      25.) Zero point of magnitude scale
(mkapert=    no) Draw apertures on the display
(mode =      q1)
```

Şekil 23. apphot “photpar”

Burada belirlenen “*aperture*” değeri yazılır. Ben burada *virgül ayırıcı* kullanarak bir kaç tane aperture göre fotometre yapmak istedim bu durum isteğe bağlıdır... Ayar yapıldıktan sonra buradan da “:q” komutu ile çıkılır.

Şimdi Şekil.19 gibi ayarlar yapıldıktan sonra “:go” komutu verilir. Size Şekil 24. deki gibi sorular soulacak. “Enter” tuşuna basarak ilerleyiniz.

```
professor@professor-pardus:~/IRAF/ders
(wcsout =      )_wcsout) The output coordinate system (logical,tv,ph
(cache =       )_cache) Cache the input image pixels in memory ?
(verify =     )_verify) Verify critical parameters in non-interacti
(update =     )_update) Update critical parameters in non-interacti
(verbose=     )_verbose) Print messages in non-interactive mode ?
(graphic=    )_graphics) Graphics device
(display=    )_display) Display device
(mode =      )_ql)

Centering algorithm (centroid) (CR or value):
  New centering algorithm: centroid
Centering box width in scale units (10.) (CR or value):
  New centering box width: 10, scale units 10, pixels
Sky fitting algorithm (centroid) (CR or value):
  Sky fitting algorithm: centroid
Inner radius of sky annulus in scale units (25.) (CR or value):
  New inner radius of sky annulus: 25, scale units 25, pixels
Width of the sky annulus in scale units (5.) (CR or value):
  New width of the sky annulus: 5, scale units 5, pixels
File/list of aperture radii in scale units (15,20,25) (CR or value):
  Aperture radius 1: 15, scale units 15, pixels
  Aperture radius 2: 20, scale units 20, pixels
  Aperture radius 3: 25, scale units 25, pixels
Standard deviation of background in counts (INDEF) (CR or value):
  New standard deviation of background: INDEF counts
Minimum good data value (INDEF) (CR or value):
  New minimum good data value: INDEF counts
Maximum good data value (INDEF) (CR or value): █
```

Şekil 24. *apphot "phot" :go*

Voila! İşlem bittikten sonra **.mag** dosyalarımız oluşmuş olmalı! Kontrol edin:)

```
apphot> ls *B*.mag.?
```

Bu işlemler diğer aynı şekilde V ve R bandlarına göre de uygulanır ve o bandlara ait **"mag"** dosyaları da elde edilir.

Evet tüm bandlarda **"mag"** dosyalarını elde ettik şimdi **magnitude** dosyaları içerisinden sonuçlarımızı (**ID, HJD, Parlaklık, Parlaktaki Hata, Hava Kütleli**) ayrı bir dosyaya çekelim; Şöyle ki;

Bu şekilde diğer bandlarda ayrı şekilde kontrol edilidir.

```
apphot> txdump *.mag.1 id,otime,mag,merr,xairmass > auser_sonucB
Boolean expression for record selection (yes): ID=1
```

Burada ID=1 den kastımız bize sadece değişen yıldızın sonuçlarını ver demek istersek ID=2 yazarak mukayesenin de fotometre sonuçlarını alabiliriz. Şöyle ki;

```
apphot> txdump szdf_AUSeR-*B*.mag.1 id,otime,mag,merr,xairmass > auser_sonucB
Boolean expression for record selection (ID=2): ID=1
apphot> txdump szdf_AUSeR-*B*.mag.1 id,otime,mag,merr,xairmass > auser_muk_sonucB
Boolean expression for record selection (ID=1): ID=2
apphot> txdump szdf_AUSeR-*V*.mag.1 id,otime,mag,merr,xairmass > auser_sonucV
Boolean expression for record selection (ID=2): ID=1
apphot> txdump szdf_AUSeR-*V*.mag.1 id,otime,mag,merr,xairmass > auser_muk_sonucV
Boolean expression for record selection (ID=1): ID=2
apphot> txdump szdf_AUSeR-*R*.mag.1 id,otime,mag,merr,xairmass > auser_sonucR
Boolean expression for record selection (ID=1): ID=1
apphot> txdump szdf_AUSeR-*R*.mag.1 id,otime,mag,merr,xairmass > auser_muk_sonucR
Boolean expression for record selection (ID=2): (Burada direk "Enter" 'e de basılabilir.)
```

```
apphot> ls auser_*
auser_muk_sonucB auser_muk_sonucV auser_sonucR
auser_muk_sonucR auser_sonucB auser_sonucV
```

İşte emeklerimizin hepsi mavi ile boyanmış dosyalar içindi. “**Değişen Yıldızımız**” ve “**Mukayese Yıldızımız**” için fotometre sonuçlarını elde ettik. Örnek olarak “**auser_sonucB**” dosyasının içine bakalım;

```
apphot> type auser_sonucB
1 2454977.46470374 16.382 16.352 16.336 0.006 0.009 0.012 1.114069
1 2454977.46568752 16.360 16.344 16.331 0.006 0.008 0.011 1.116207
1 2454977.46670602 16.354 16.339 16.325 0.006 0.009 0.012 1.118462
1 2454977.4676898 16.359 16.345 16.335 0.006 0.009 0.012 1.12068
1 2454977.46866201 16.353 16.339 16.331 0.007 0.009 0.012 1.122912
1 2454977.46963421 16.351 16.341 16.338 0.006 0.009 0.012 1.125184
1 2454977.47061799 16.359 16.339 16.332 0.006 0.008 0.011 1.127523
```

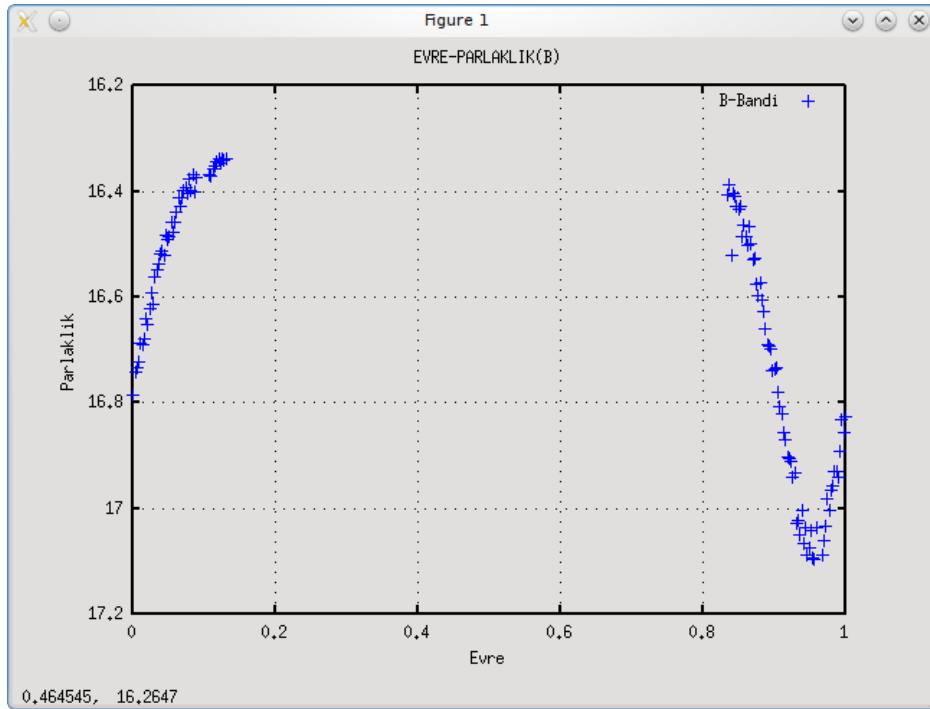
Burada 1. sütun Değişen veya Mukayeseyi, 2. sütun HJD'yi, 3., 4. ve 5. sütun Parlaklığı(Çünkü üç adet **aperture** değeri girmiştik. **Bkz. Şekil 23.**) , 6-7-8 Parlaklıklardaki hataları, 9. sütun ise Hava Kütlesini göstermektedir.

23. Şimdi Grafiğe dökelim.

Bilindiği üzere Evre-Parlaklık grafiği çizilememiz için bir T_0 değerine ihtiyacımız var. Bunun için [SAO/NASA Astrophysics Data System \(ADS\)](#) 'de yayın tarayarak şu T_0 ve P değerlerini yayınlardan aldım. [1992IBVS.3802....1L](#)² yayınından Period değerini **0.38650124** ve [2007IBVS.5791....3](#) yayından da T_0 değerini **2453215.3608** olarak aldım ve bu değerlere göre oluşturulan hesap tabloları(excel formatında) ve grafikler ekle ayrıca sunulmuştur. Burada grafik kalitesi açısından *GNU/Octave* ile hesaplanmış ve *gnuplot* ile çizilmiş grafikler(pdf) ekte sunulmuştur.

GNU/Octave;

```
>>> load auser_sonucB
>>> format long E hjd
>>> hjd=auser_sonucB(:,2);
>>> format free M
>>> M=auser_sonucB(:,4);
>>> T0=2453215.3608
T0 = 2453215.3608
>>> P
P = 0.38650124
>>> plot(((hjd-T0)/P)-floor((hjd-T0)/P),M,"+g;B-Bandi;")
>>> xlabel("Evre")
>>> ylabel("Parlaklik")
>>> title('EVRE-PARLAKLIK(B)');
>>> grid
>>> axis("ij")
>>> print ("B.pdf.ps", "-color", "-dps")
system("ps2pdf \"B.pdf.ps\" B.pdf")
```



² The Variation of the Light Curves of AU Ser - [Li, Zong-Yun; Zhan, Zhou-Sheng; Li, Yu-Lan - 11/1992](#)

³ Minima Times of Some Eclipsing Binary Stars [Guroi, B.; Derman, E.; Muyesseroglu, Z.; Gurdemir, L.; Gokay, G.; Ozbek, N.; Sagir, U.; Kalci, R.; Salman, G.; Coker, D.; Eminoglu, B.; Demircan, Y.; Terzioglu, Z.](#) Information Bulletin on Variable Stars, 5791, 1.

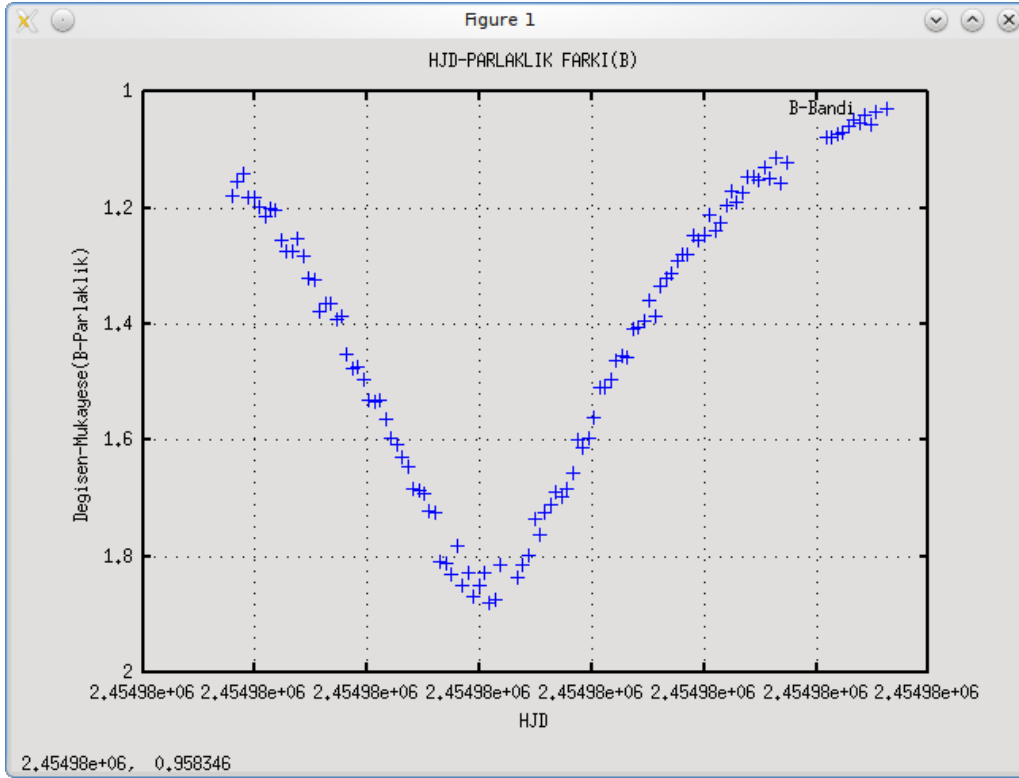
Şekil 25. Evre-Parlaklık Grafiği(B)

Değişen-Mukayese Parlaklık;

hjd, *M*, *P*, *To* değişkenlerini long olarak tanımlamıştık bir önceki çizimde, tekrar tanımlamayalım;

```
>>> load auser_muk_sonucB
>>> format free mukM
>>> mukM=auser_muk_sonucB(:,4)
>>> plot(hjd,M-mukM,"+b;B-Bandi;")
>>> axis("ij")
>>> grid
>>> xlabel("HJD")
>>> ylabel("Degisen-Mukayese(B-Parlaklik)")
>>> title('HJD-PARLAKLIK FARKI(B)');
>>> print ("fark.pdf.ps", "-color", "-dps")
system("ps2pdf \"fark.pdf.ps\" fark.pdf")
```

Esasında bu grafiğik mukayesemizin denetimi içindi ve herşeyin yolunda olduğunu gösteren denetim grafiği.

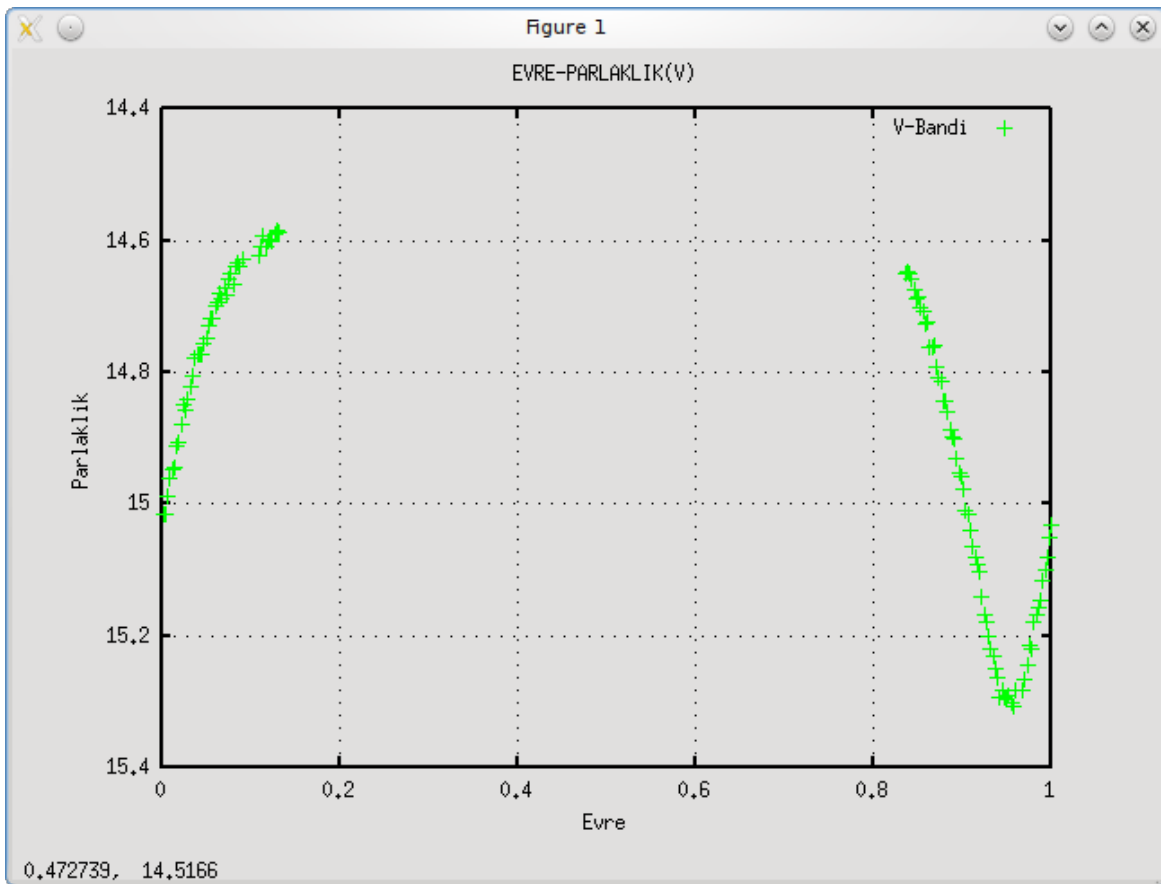


Şekil 26. HJD-PARLAKLIK FARKI


```

>>> load auser_sonucV
>>> hjd=auser_sonucV(:,2);
>>> M=auser_sonucV(:,4);
>>> P=0.38650124
P = 0.38650124
>>> T0=2453215.3608
T0 = 2453215.3608
>>> title('EVRE-PARLAKLIK(V)');
>>> plot(((hjd-T0)/P)-floor((hjd-T0)/P),M,"+g;V-Bandi;")
>>> xlabel("Evre")
>>> ylabel("Parlaklik")
>>> axis("ij")
>>> title('EVRE-PARLAKLIK(V)');
>>> grid
>>> print ("V.pdf.ps", "-color", "-dps")
system("ps2pdf \"V.pdf.ps\" V.pdf")

```



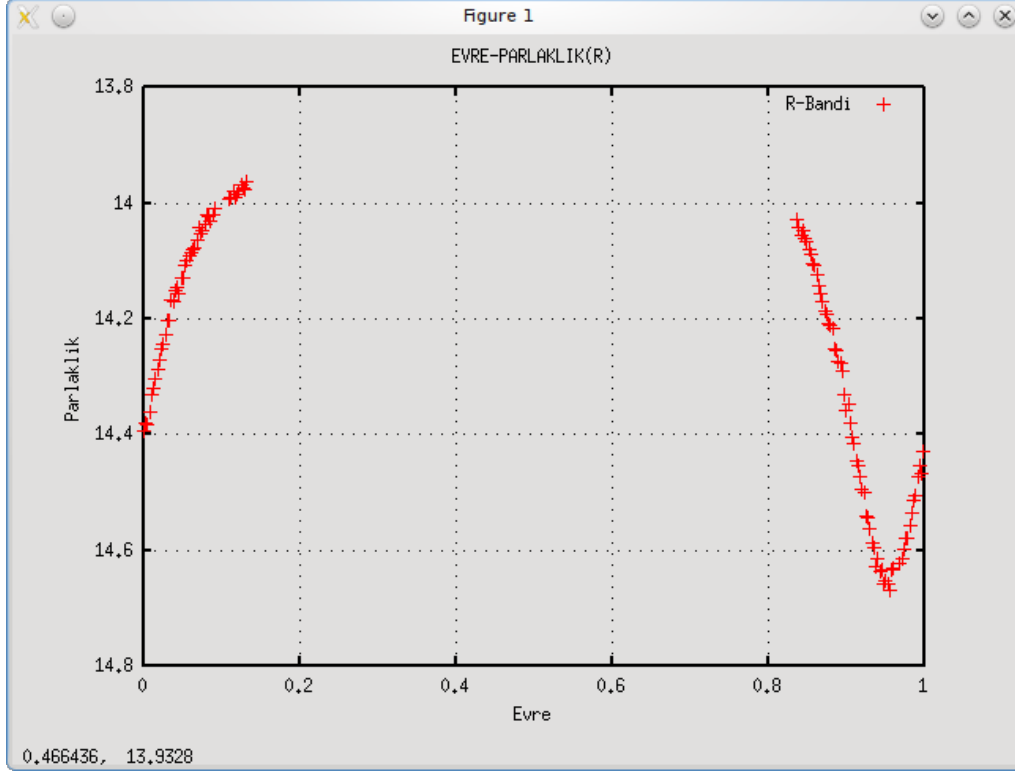
Şekil 27. Evre-Parlaklık Grafiği(V)

```

>>> load auser_sonucR
>>> hjd=auser_sonucR(:,2);
>>> M=auser_sonucR(:,4);
>>> P=0.38650124
P = 0.38650124
>>> T0=2453215.3608
T0 = 2453215.3608

```

```
>>> plot(((hjd-T0)/P)-floor((hjd-T0)/P),M, "+r;R-Bandi;")
>>> xlabel("Evre")
>>> ylabel("Parlaklik")
>>> title('EVRE-PARLAKLIK(R)');
>>> axis("ij")
>>> grid
>>> print ("R.pdf.ps", "-color", "-dps")
system("ps2pdf \"R.pdf.ps\" R.pdf")
```



Şekil 28. Evre-Parlaklık Grafiği(R)

Evet çalışmamız burada sona erdi. Bu zevkli çalışmayı sizlerle paylaşabilmemizi sağlayan değerli hocamız **Doç. Dr. Birol GÜROL**'a teşekkürü bir borç biliriz.

Yücel KILIÇ

Not: Bu çalışma tamamıyla Pardus İşletim Sistemi üzerinde yapılmıştır.