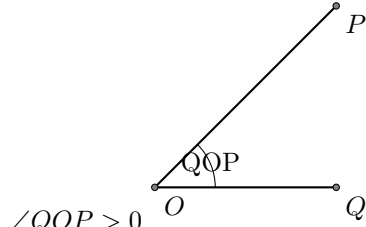
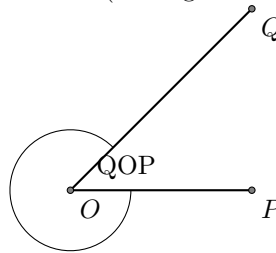


Convención del signo de los angulos



$\angle QOP > 0$

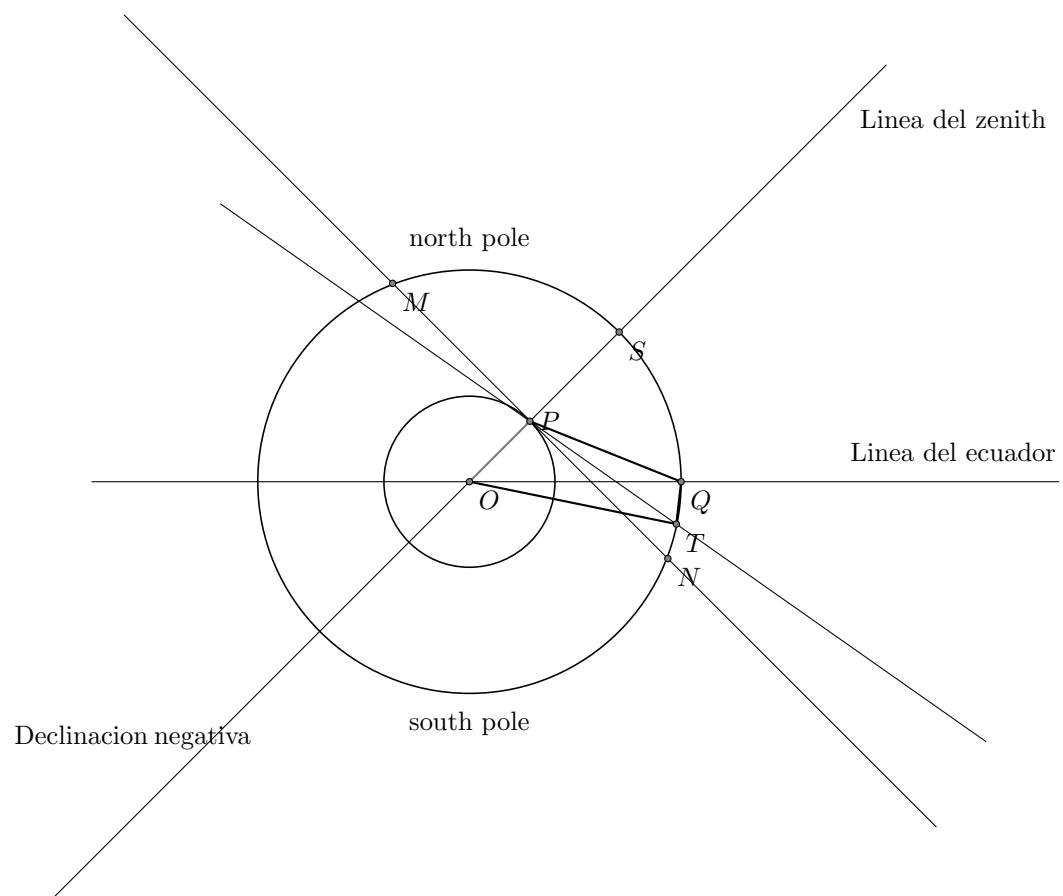
$\angle QOP < 0$ o $\angle QOP > 12 : 00 : 00h$ (los angulos mayores de 180 grados se

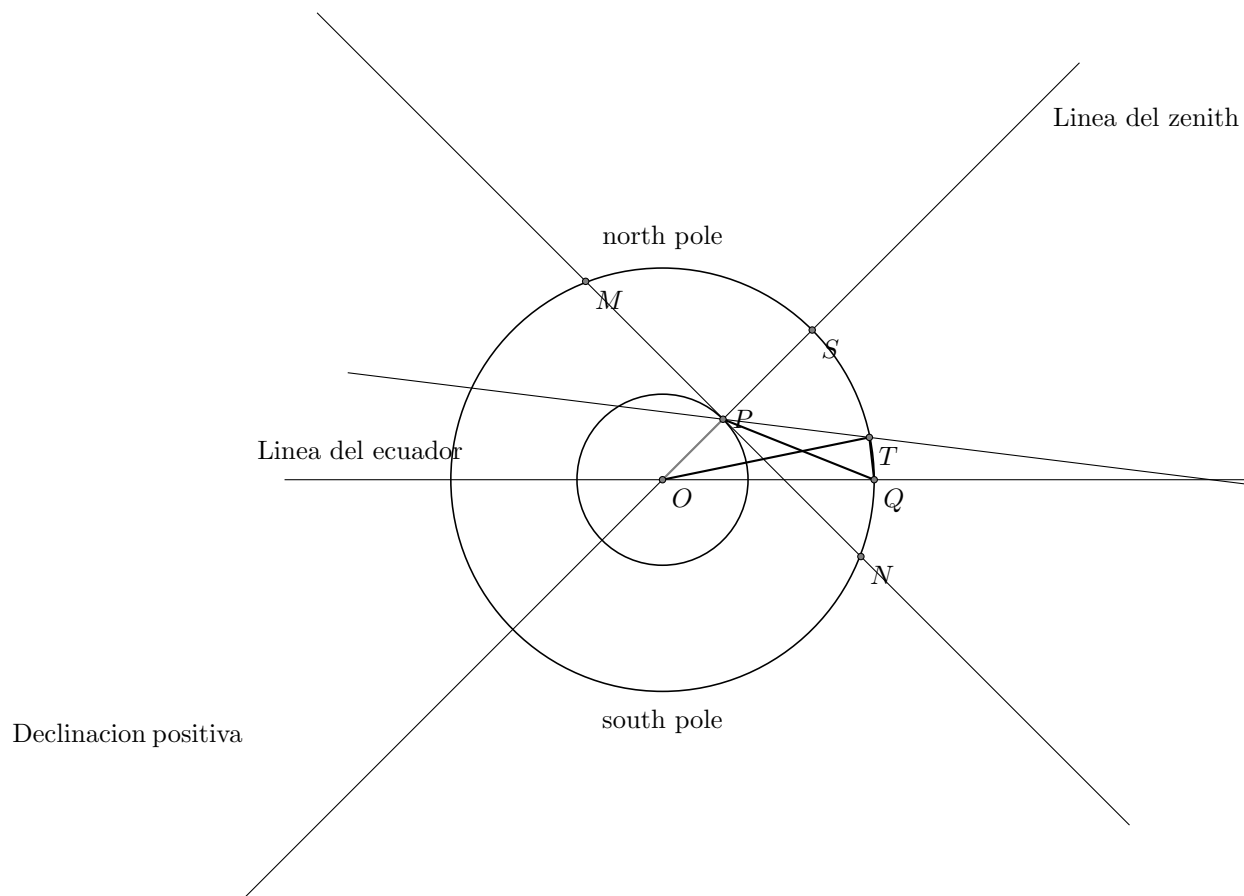


representan con valor negativo)

- O: centro de la tierra y de la esfera celeste (esfera que contiene la esfera de la tierra, el plano ecuatorial terrestre se extiende al plano ecuatorial celeste y la comparte igual que la tierra en 2 hemisferios: norte y sur y el plano del meridiano 0 se puede extender para toda la esfera celeste). La correspondiente de la latitud en la tierra es la declinación para la esfera celeste (negativa si el objeto a observar está en hemisferio sur celeste) y para la longitud la declinación recta(positiva desde el meridiano 0 en el sentido de la rotación dela tierra en torno a su propio eje y negativa en el otro sentido)
- P: punto del observador
- T: posicion del objeto observado

Proyeccion longitudinal P está en el hemisferio norte(latitud positiva)





- OQ : linea del ecuador
- OP : linea del zenit del observador (la linea que une el observador con el centro de la tierra) perpendicular en la linea del horizonte MN (solo se pueden ver objetos en el arco MSN : objetos con declinación entre $-(90^\circ - \text{latitude})$ y $180^\circ - (90^\circ - \text{latitude})$)
- PT : linea de vision del observador
- $\angle QPT$ Para la proyeccion longitudinal es declinacion medida por el telescopio (negativa si la estrella está debajo del plano ecuatorial) - el angulo en el cielo entre la dirección del telescopio orientado hacia el objeto que observamos y la dirección del telescopio orientado hacia un objeto que está en el ecuador celeste.

- $\angle QOT$ declinacion de la estrella (negativa si la estrella está debajo del plano ecuatorial) - medida desde un punto que esta en el ecuador de la tierra
- $\angle QOP$ latitud
- $\angle TPS$ proyeccion en este plan del angulo zenith distance(formado por la linea del zenith y la direccion en cual apunta el telescopio)
- En práctica esta es zenith distance (el azimuth no cuenta) ZD medida en grados (dd:mm:ss) con dd entre 00 y 90 y Airmass se aproxima como $1 / \cos(ZD)$

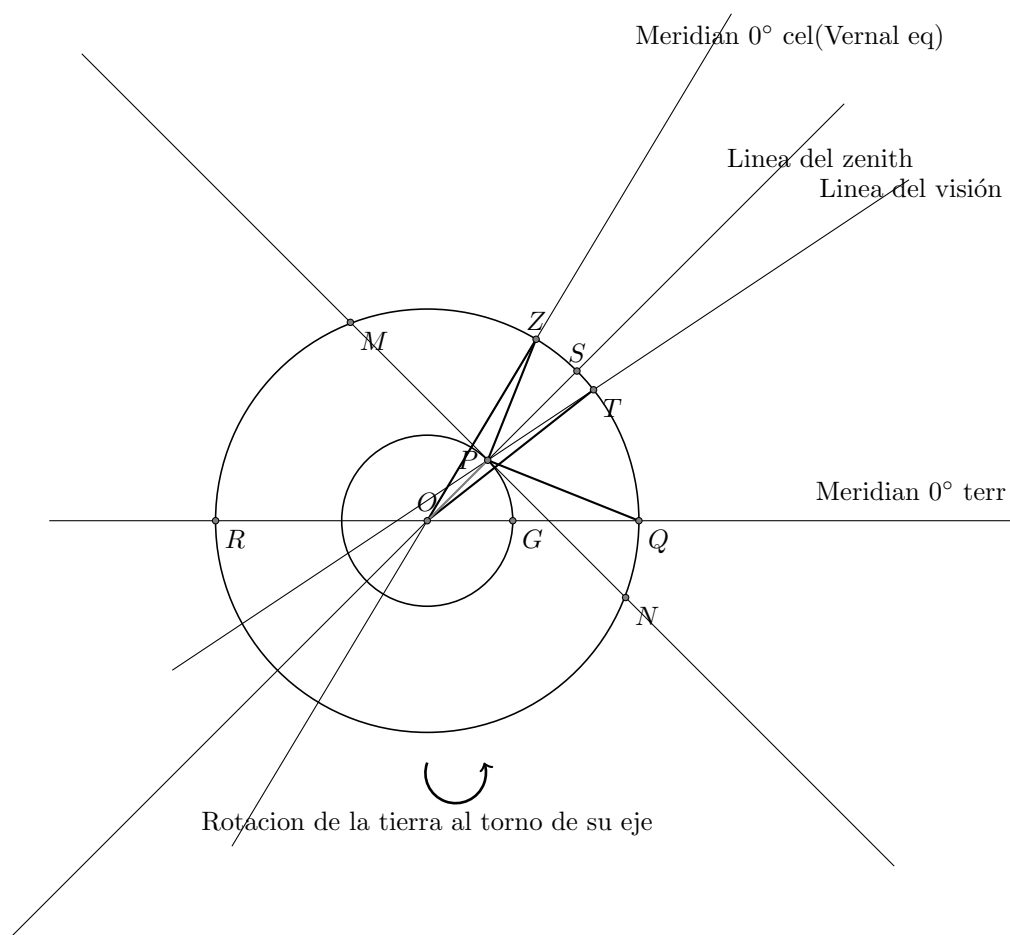
$$\angle POQ = \angle POT + \angle TOQ$$

Q(el objeto de referencia) está muy lejos así que las distancias $OQ, PQ, PT, OT \gg OP$ (el radio de la tierra) y $\angle TPQ \approx \angle TOQ$ (la declinacion medida por el telescopio es aprox la declinacion del objeto observado medida desde un punto del ecuador terrestre si el objeto está bastante lejos)

Ademas si se apunta el telescopio cerca al zenith $\angle SPT \approx \angle POT \approx 0$ así que $\angle POQ \approx \angle TPQ$ (latitud \approx declinación)

Miramos las imagenes del flat del cielo (cuando el telescopio esta orientado aprox hacia el zenith) y determinamos la latitud 28:17:48 (grados)

Proyeccion transversal P está al este del meridiano 0 terrestre(longitud positiva)



- Z vernal equinox
- G punto en la tierra en el meridiano 0
- $\angle ZOQ$ UniversalTime

- $\angle ZPT \approx \angle ZOT$ RightAscension
- $\angle ZPS \approx \angle ZOS$ SiderealTime
- $\angle SPT \approx \angle SOT$ object hour
- $(ST = RA + h)$, cuando el objeto a observar está justo arriba($h=0$) $ST = RA$
- $\angle QOP$ longitud

Cuando Sidereal Time es aprox 0 UT representa la longitud

longitud 22:21:45(hours) = -24:34:45(degrees)

Instrumento

<http://www.ing.iac.es/astrophysics/instruments/wfc/index.html>

Noise

En los headers de las imagenes(y en la especificacion de la pagina web para exposiciones largas(48 sec) - exposiciones cortas(29 sec)):

GAIN = 2. / gain, electrons per adu (1.26 - 2.5)

RDNOISE = 5.4 / read noise, electrons (4.6 - 9)

En la pagina web:

Noise(ADU) (3.7 - 3.6)

Bias (2030 - 1830)

Field of view and detector size

En la pagina web:

Field of View 34x34arcmin

Pixel scale 0.333 arcsec / pixel

pixel size = 13.5 microns

Numero de pixeles del detector = FOV / pixel scale = 6126 x 6126, diferente al numero de pixeles por filas y columnas del header de las imagenes fits

tamaño del detector = num pixeles * pixel size = 82.7 mm

Reducción de las imagenes

- Separar las imagenes en carpetas por tipo: object y flat y cada una por el filtro(B, R, V) (función `initDirs()` in `red.py`)
- Corrección bias(sustraer bias de la imagen): con la función `colbias` de `iraf`(función `trimAndOverscan()` in `red.py`): se calcula una columna como media de las columnas definidas en la sección bias y se sustrae de todas las columnas de la sección trim. Se hace para todas las imagenes: flat y object
- Crear los ficheros flat que se usan para la corrección de flat: con `imcombine` se hace una media de todas las imaganes flat para cada filtro y despues se normalizan dividiendo por la media(mean) de cada uno (usando la función `imarith`) (función `createFlatFiles()` in `red.py`)
- Corrección de flat: Todas las imagenes tipo object se dividen con el flat medio normalizado para cada filtro (función `flatCorrection()` in `red.py`)
- para la fotometría elegimos solo la parte central del cúmulo (las imagenes que tienen los últimos 3 dígitos antes de la extensión .fits formando un número $\in [96, 105]$)
- corregir pixeles malos (independientes del filtro): de forma aproximadamente automática: elegimos 2 flat con tiempo de exposition largo (EXPTIME keyword in header) `M37New/flat/V/Nov30032.fits` (60) y `M37New/flat/R/Nov30031.fits` (35) y hacemos la media con `imcombine` y 2 con el valor EXPTIME pequeño: `M37New/flat/V/Nov30016.fits` (2) `M37New/flat/V/Nov30015.fits` (2) Luego dividimos los 2 resultados y creamos una mascara con `ccdmask`. Despues ejecutamos `fixpix` para lodas las imagenes tipo object y el fichero mask obtenido antes:

```
ecl> imcombine M37New/flat/R/Nov30031.fits,M37New/flat/V/Nov30032.fits M37New/flat/FlatR
ecl> imcombine M37New/flat/V/Nov30016.fits,M37New/flat/V/Nov30015.fits M37New/flat/FlatV
ecl> imarith M37New/flat/FlatBigExptime / M37New/flat/FlatSmallExptime M37New/flat/FlatMed
ecl> noao
ecl> imred
ecl> ccdred
ecl> ccdmask M37New/flat/FlatBigSmallDivided M37New/flat/MaskFile
ecl> cd M37New/object/V
ecl> fixpix @list /scratch1/tobs/M37New/flat/MaskFile
ecl> cd ../R
ecl> fixpix @list /scratch1/tobs/M37New/flat/MaskFile
ecl> cd ../B
ecl> fixpix @list /scratch1/tobs/M37New/flat/MaskFile
```

- Si queremos definir una mascara:

Si miramos en la imagen Nov30098.fits y queremos definir en la columna $577(x \in [577, 578))$ como pixeles malos la parte de abajo ($y \in [0, 259)$)

```
bpopescu@colibri:/scratch1/tobs$ cat col577Mask
box 577 0 578 259
```

```
ecl> mskregions.regions="col577Mask"
ecl> mskregions.dims="1024,1024"
ecl> mskregions.masks="M37New/flat/Col577Mask"
ecl> mskregions.refimages="M37New/object/V/Nov30098.fits"
ecl> mskregions
The list of region specifications (col577Mask):
The list of output mask images (M37New/flat/Col577Mask):
The list of input reference images (M37New/object/V/Nov30098.fits):
Creating mask M37New/flat/Col577Mask.pl using reference image M37New/object/V/Nov30098.
Using regions file col577Mask
ecl>
```

(Si no queremos que nos pregunte siempre por los parametros hay que modificar el parametro mode del task de 'ql' a 'h')

Se crea un fichero /scratch1/tobs/M37New/flat/Col577Mask.pl, no hace falta especificar la extensión .pl en el paso siguiente cuando se pasa como parametro al task fixpix:

Después hay que modificar todas las imagenes tipo object con fixpix como antes

```
ecl> cd M37New/object/V
ecl> fixpix @list /scratch1/tobs/M37New/flat/Col577Mask
ecl> cd ../R
ecl> fixpix @list /scratch1/tobs/M37New/flat/Col577Mask
ecl> cd ../B
ecl> fixpix @list /scratch1/tobs/M37New/flat/Col577Mask
```

Miramos la diferencia entre las imagenes antes de aplicar esta mascara y despues:

```
cl> display.xsize=0.5
cl> display M37New/object/V/Nov30098.fits fill+ xcen=0.25
cl> display Nov30098.fits erase- fill+ xcen=0.75
```

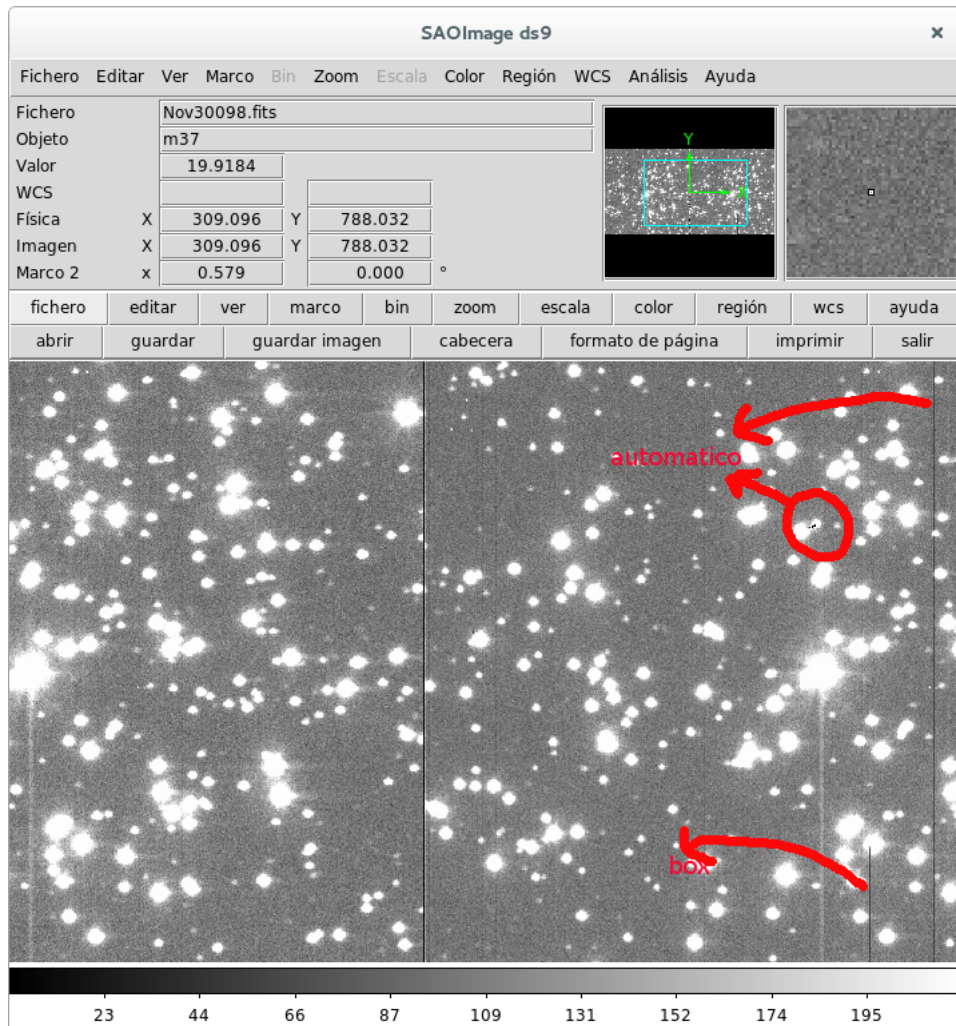



Figure 1: *Perfil radial de las estrellas de referencia para la alineación*

- FOTOMETRIA: Alinear las imagenes: Se elige una imagen con buena respuesta elegí Nov30098.fits en filtro V y 3 estrellas bien separadas en la imagen que no tengan mucho ruido y que no estén saturadas

para ver el perfil radial:

en xgterm abrir el visualizador ds9 (funciona con ximtool, pero no he probado)
en iraf

```
ecl>display image.fits
```

```
ecl>imexam image.fits
```

hacer click en el punto donde quieres ver el perfil radial y pulsar "r" después

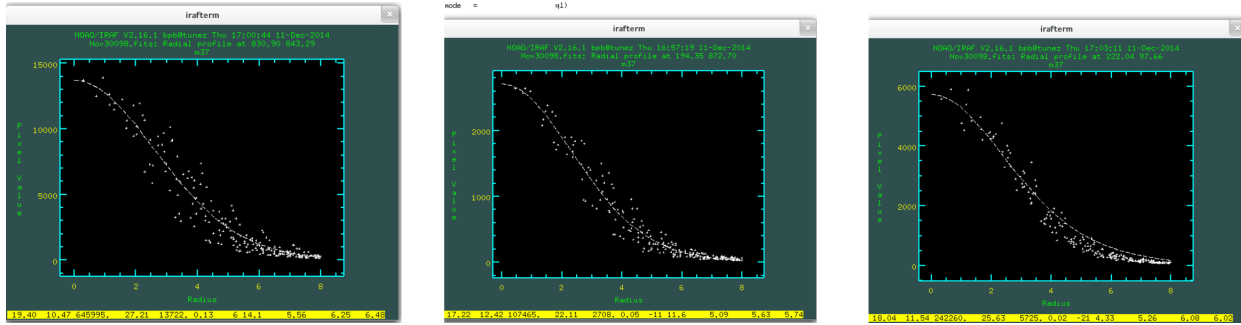


Figure 2: *Perfil radial de las estrellas de referencia para la alineación*

- escribir las coordenadas en un fichero

```
[bpopescu@colibri tobs]$ cat coords-98.txt
194 872
831 843
222 98
```

- Usamos imcentroid para obtener las coordenadas exactas usando como input la misma imagen de referencia

```
ecl> imcentroid.input = "M37New/object/V/Nov30098.fits"
ecl> imcentroid.reference = "M37New/object/V/Nov30098.fits"
ecl> imcentroid.coords = "coords-98.txt"
ecl> imcentroid
```

salen las coordenadas:

#Coords	Image	X-center	Err	Y-center	Err	Num	
/scratch/M37New/object/V/Nov30098.fits	194.519 (0.028)	872.795 (0.024)	1				
/scratch/M37New/object/V/Nov30098.fits	830.926 (0.015)	843.310 (0.010)	2				
/scratch/M37New/object/V/Nov30098.fits	222.031 (0.018)	97.730 (0.019)	3				

#Refcoords	Reference	X-center	Err	Y-center	Err	Num	
/scratch/M37New/object/V/Nov30098.fits	194.519 (0.028)	872.795 (0.024)	1				
/scratch/M37New/object/V/Nov30098.fits	830.926 (0.015)	843.310 (0.010)	2				
/scratch/M37New/object/V/Nov30098.fits	222.031 (0.018)	97.730 (0.019)	3				

#Shifts	Image	X-shift	Err	Y-shift	Err	N	Internal	
/scratch/M37New/object/V/Nov30098.fits	0.000 (0.017)	0.000 (0.015)	3	(0.				

miramos los nuevos valores de x-center y y-center y los ponemos en otro fichero:

```
[bpopescu@colibri tobs]$ cat coords-98-aligned.txt
194.519 872.795
830.926 843.310
222.031 97.730
```

- Alinear todas las imágenes: usando imalign Creamos una lista(fichero listCenter) con las imágenes que hay que alinear(las tipo "object" del centro del cúmulo (con el número del nombre de 96 a 105)) y otra a la cual añadimos 'a' antes de la extensión (fichero listCenterAligned) con alignImages.py (ejecutando align desde pyraf, sale un error, porque?)

```
ecl> imalign.input="@listCenter"
ecl> imalign.reference = "M37New/object/V/Nov30098.fits"
ecl> imalign.coords = "coords-98-aligned.txt"
ecl> imalign.output = "@listCenterAligned"
ecl> imalign.interp_type = "nearest"
ecl>imalign
```

- BUSCAR ESTRELLAS display y daoedit para todas las imágenes alineadas de listcenterAligned: por ejemplo M37New/object/R/Nov30103a.fits:

```
ecl> noao
noao> digiphot
digiphot> daophot
```

```
daophot> display M37New/object/R/Nov30103a.fits
frame to be written into (1:16) (2):
z1=8.931553 z2=90.34003
```

```
daophot> daoedit M37New/object/R/Nov30103a.fits
```

después con el cursor encima de la estrella (en la ventana ds9) pulsamos 'a' y en la tecla enter se abre una ventana con las columnas :

```
# XCENTER YCENTER      SKY SKYSIGMA    FWHM  COUNTS      MAG
```

después de seleccionar varias guardamos la salida en un fichero:

```
bpopescu@colibri:/scratch1/tobs$ cat M37New/object/R/Nov30103a.fits-daoedit
```

```
# XCENTER YCENTER      SKY SKYSIGMA    FWHM  COUNTS      MAG
  832.83  575.81      43.9   6.99    4.98  54017.2 -11.831
  418.22  684.58      55.5  12.89    5.05 235331.5 -13.429
  175.20  652.63      51.5   9.59    5.04 266379.4 -13.564
   52.98  456.31      40.7   6.49    4.44  58369.4 -11.915
   89.85  235.86      47.8   9.12    4.97 211897.9 -13.315
```

624.22	843.96	47.2	8.37	5.02	162999.8	-13.030
193.41	872.61	39.7	6.35	4.88	64339.4	-12.021
273.99	859.73	37.5	6.56	4.97	22932.2	-10.901
348.81	133.11	46.2	9.01	5.29	153856.1	-12.968
611.49	559.89	47.6	8.17	5.07	58278.2	-11.914
912.48	143.12	40.0	6.54	5.62	34869.5	-11.356

Para salir de daoedit pulsamos 'q'

- El proceso de generar un fichero con las coordenadas de todas las estrellas con daofind se puede hacer de forma manual: para cada fichero imagen:

Hay que establecer los siguientes parametros una vez para todas las imagenes

```
daophot> datapars.exposure = "EXPTIME"
daophot> datapars.airmass = "AIRMASS"
daophot> datapars.filter = "INSFILTE"
daophot> datapars.ccdread = "RDNOISE"
daophot> datapars.gain = "GAIN"
```

Hacemos una media de las columnas 4(SKYSIGMA) y 5(FWHM) con

```
python daoeditAvg.py M37New/object/R/Nov30103a.fits-daoedit
sky sigma mean is 8.19
FWHMPSF mean is 5.03
```

Antes de ejecutar daofind para cada imagen hay que establecer para cada una los parametros FWHM y SKYSIGMA obtenidos con daoedit en el paso anterior

```
daophot> datapars.fwhmpsf = 5.03
daophot> datapars.sigma = 8.19
```

Ejecutamos daofind:

```
daophot> daofind M37New/object/R/Nov30103a.fits M37New/object/R/Nov30103a.fits-daofind
```

Esto va a crear el fichero M37New/object/R/Nov30103a.fits-daofind con las magnitudes de todas las estrellas encontradas en la imagen

- Teniendo los ficheros *-daoedit con las coordenadas de algunas estrellas elegidas de forma interactiva como se explicó antes podemos ejecutar

```
python daofind.py listCenterAligned
```

que genera los ficheros *-daofind de todas las imagenes alineadas

- Si tenemos la imagen ya cargada en ds9 podemos superponer las coordenadas (el fichero generado por daofind) con tvmark (! el mismo frame) para ver si salen bien (si hay estrellas que no se han encontrado hay que reducir el valor del parametro findpars.threshold (4 por defecto) o ampliarlo si marca estrellas que no son)

```
daophot> tvmark coords=M37New/object/R/Nov30103a.fits-daofind
```

Observaciones

- eliminar pixeles malos: ccdmask De forma adicional quiero aplicar otro mask y quiero definir columnas 660 y 678 como columnas malas(para corregir por ejemplo img 65) y uso un fichero mask de forma:

```
cols (660,660) || cols (678,678) ? 1 : 0
```

pero sale un error: "segmentation violation". Pero funciona con un fichero mask definido:

```
circle (220., 220., 50.) && circle (240., 220., 50.) ? 1 : 0
```