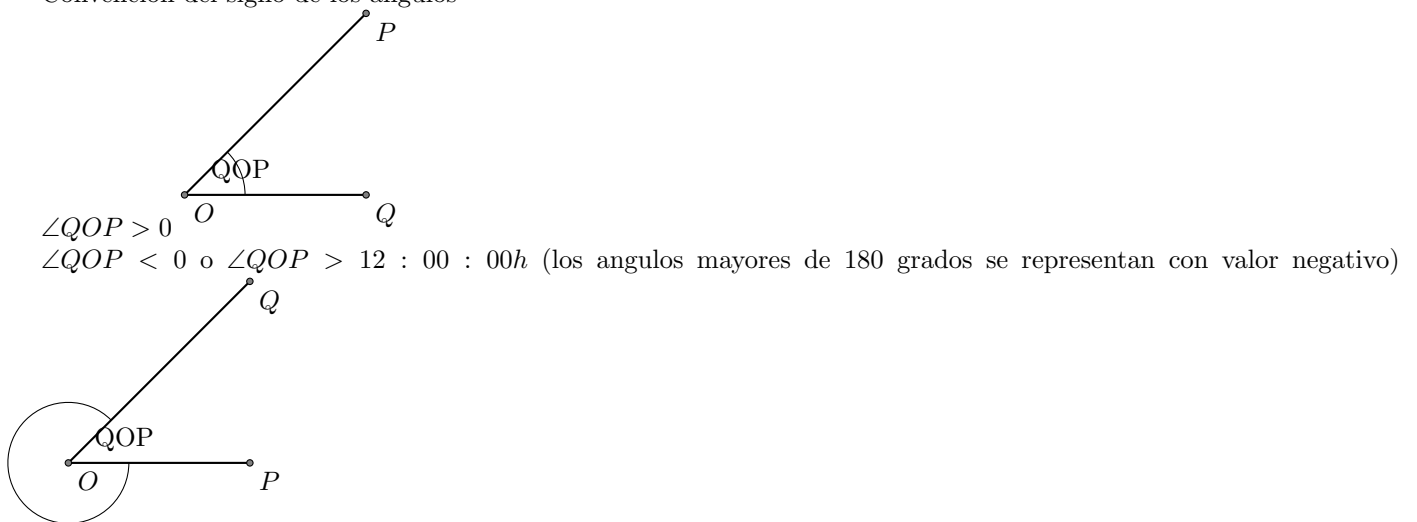
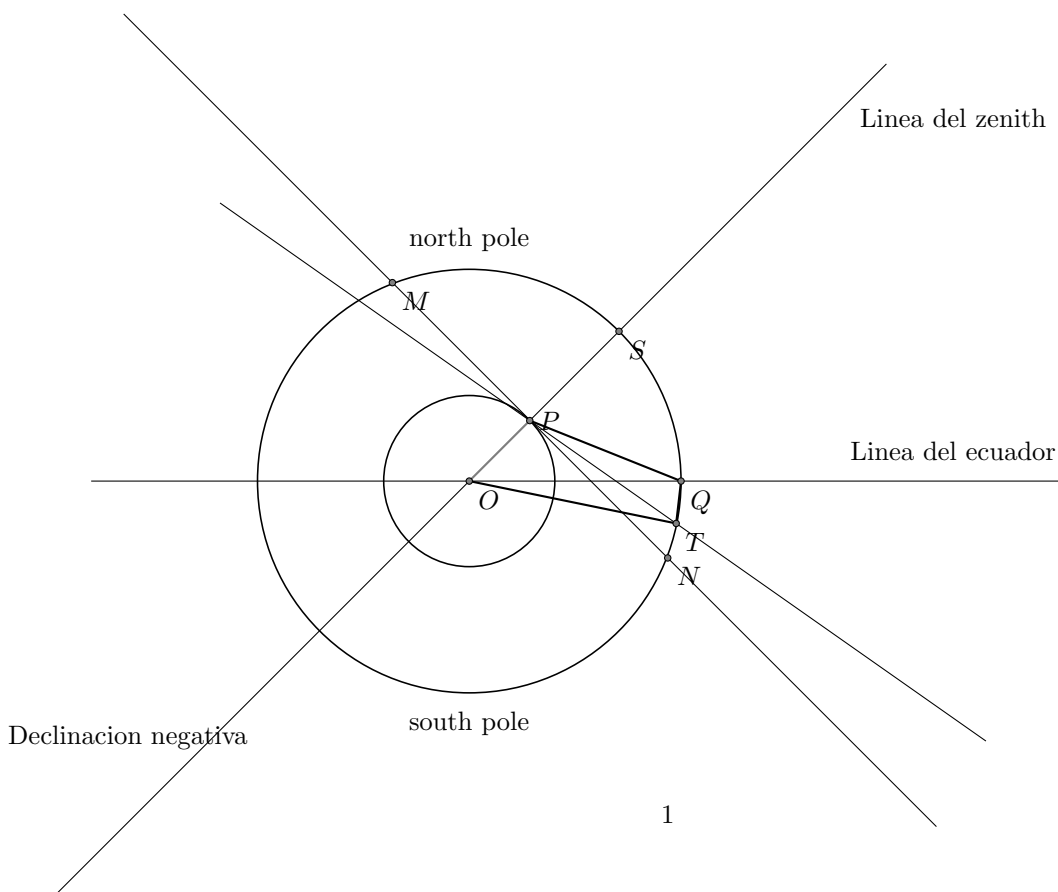


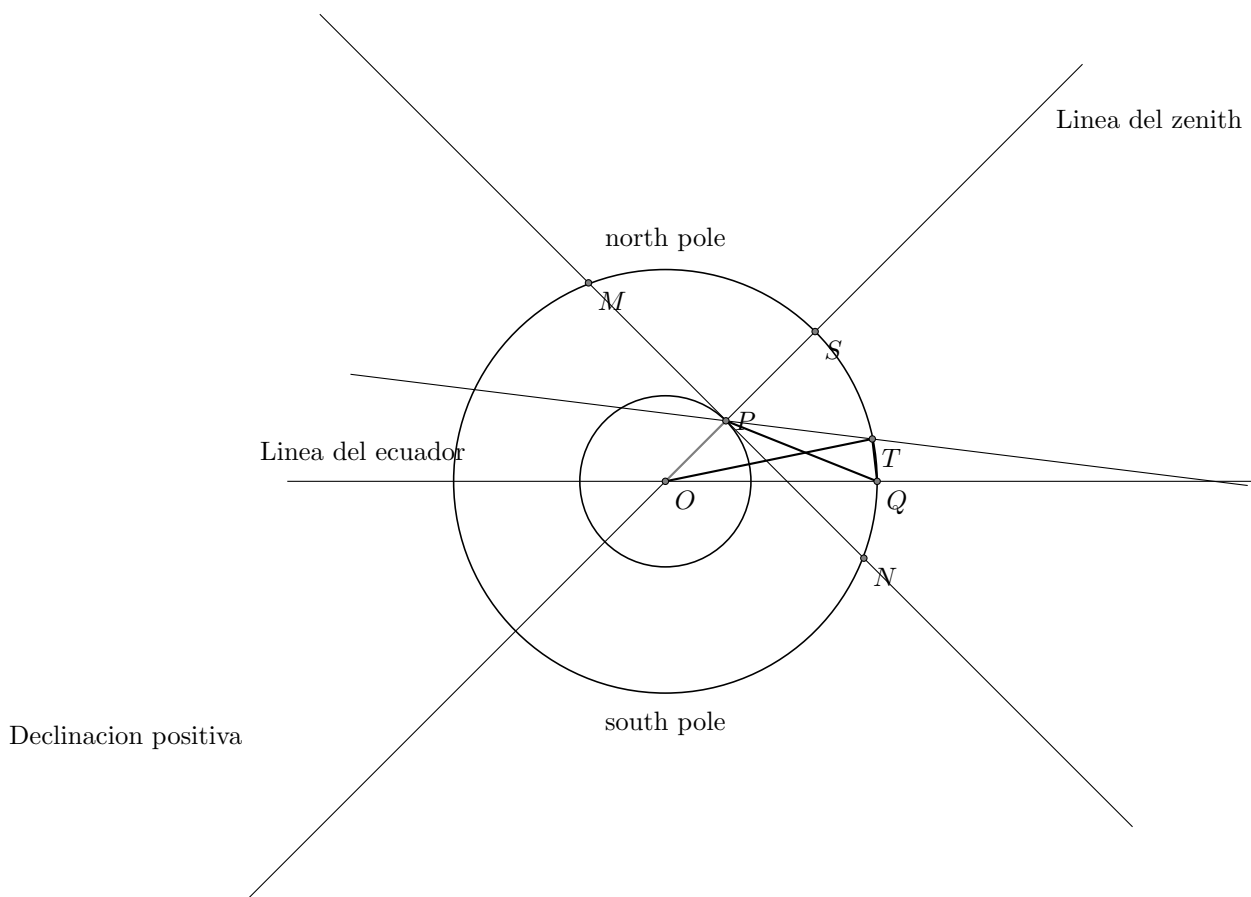
Convención del signo de los angulos



- 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: línea del ecuador
- OP: línea del zenit del observador(la línea que une el observador con el centro de la tierra) perpendicular en la línea 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: línea 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 línea del zenith y la dirección 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 $1 / \cos(\text{ZD})$

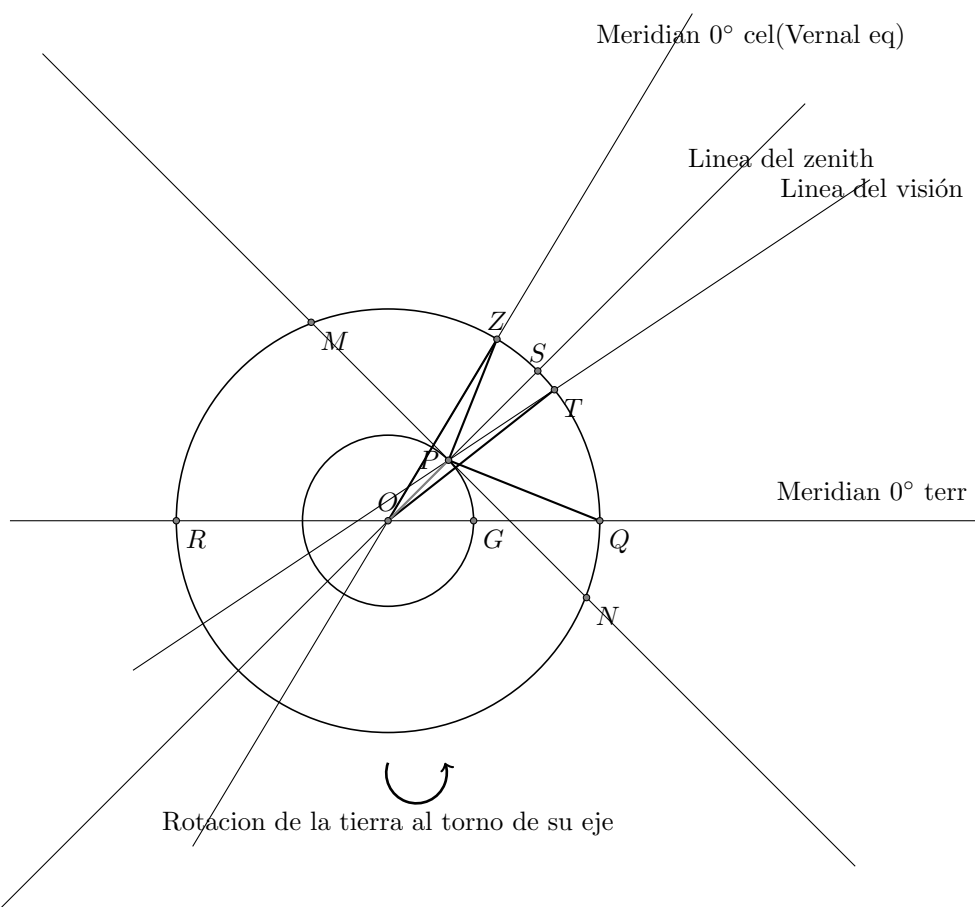
$$\angle POQ = \angle POT + \angle TOQ$$

Q(el objeto de referencia) está muy lejos asi 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) Las nueva carpeta donde se van a poner las imagenes se llama M37New (con eso me refiero mas adelante para especificar la carpeta raiz que contiene todas las carpetas con imagenes clasificadas por tipo y filtro y a veces tiene la ruta absoluta)
- 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 imagenes flat para cada filtro y despues se normalizan dividiendo por la media(mean) de cada uno (usando la función iraf imarith) (función createFlatFiles() en red.py)
- Corrección de flat: Todas las imagenes tipo object se dividen con el flat medio normalizado para cada filtro (función flatCorrection() en red.py)

- 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 todas las imagenes tipo object y el fichero mask obtenido antes:

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

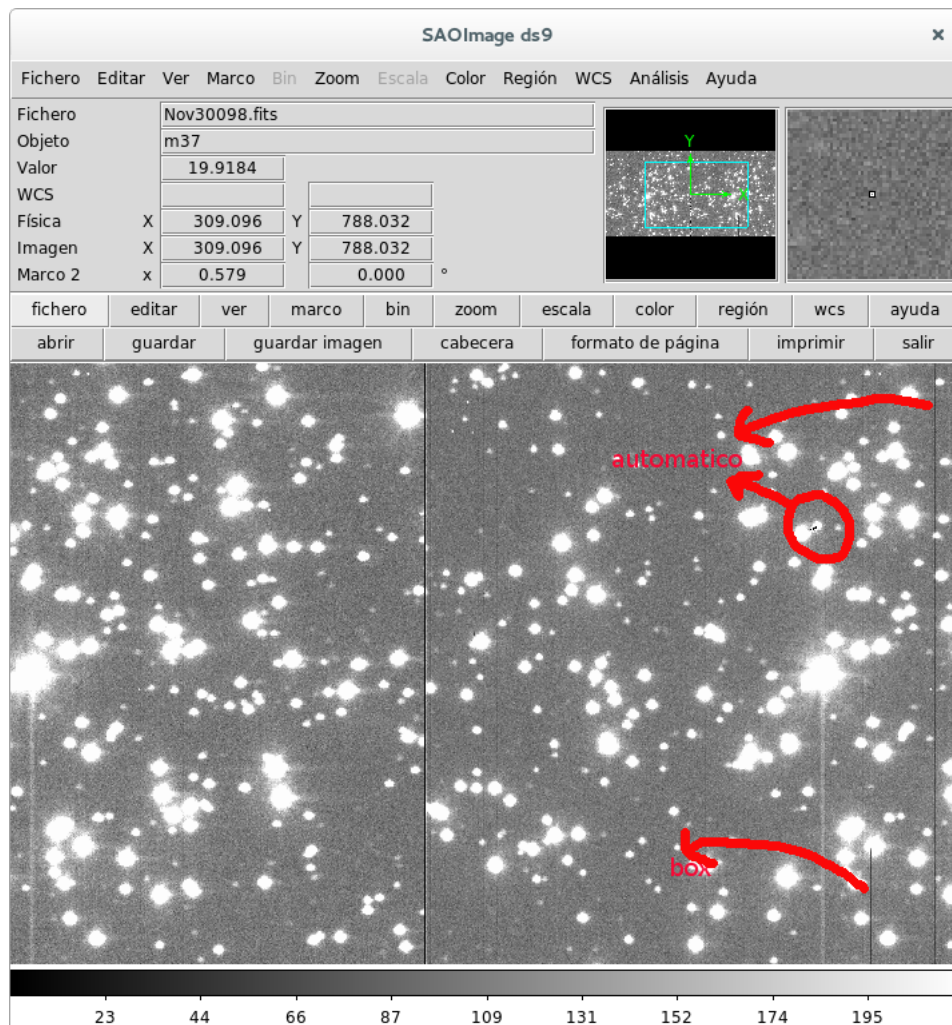


Figure 1: *Perfil radial de las estrellas de referencia para la alineación*

Fotometría

- para la fotometría elegimos solo la parte central del cúmulo (las imágenes que tienen los últimos 3 dígitos antes de la extensión .fits formando un número $\in [96, 105]$)
- **Alinear las imágenes:** Se elige una imagen con buena respuesta (del centro del cúmulo): elegí Nov30098.fits en filtro V y 3 estrellas bien separadas en la imagen que no tengan mucho ruido y que no estén saturadas

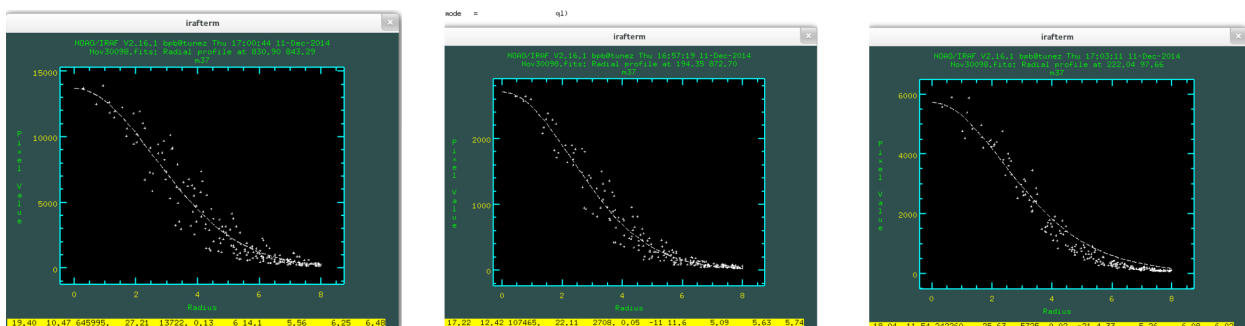


Figure 2: *Perfil radial de las estrellas de referencia para la alineación*

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

- 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	(0.000,0.000)	

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 imagenes: usando imalign Creamos una lista(fichero listCenter-M37) con las imagenes que hay que alinear(las tipo "object" del centro del cúmulo (con el numero del nombre de 96 a 105)) y otra a cual añadimos 'a' antes de la extensión (fichero listCenterAligned-M37) con createListAlignObject.py (ejecutando align desde pyraf, sale un error, porque?)

```
ecl>imalign.input="@listCenter-M37"
ecl>imalign.reference = "M37New/object/V/Nov30098.fits"
ecl>imalign.coords = "coords-98-aligned.txt"
ecl>imalign.output = "@listCenterAligned-M37"
ecl>imalign.interp_type = "nearest"
ecl>imalign
```

- **Buscar estrellas y obtener las magnitudes** Para poder tener valores de las medias de sky sigma y fwhm para cada imagenes (que usamo después como parametros de entrada en daofind para que busque de forma automática todas las estrellas de la imagen) cogemos algunas estrellas de referencia de cada imagen miramos la distribucion radial de la luz = COUNTS(XCENTER, YCENTER, Rmax) de cuales queremos que iraf calcule sky sigma, fwhm y magnitud. Nos interesan sky sigma y fwhm que definen el perfil de intensidad (intenta a ajustarlo a una gaussiana con estos parámetros)

```
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
```

despues con el cursor encima de la estrella (en la ventana ds9)(las coordenadas donde está el cursor con las columnas :

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

despues 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'

- Para hacer una media de las columnas 4(SKYSIGMA) y 5(FWHM) de cada fichero daoedit podemos ejecutar:

```
python daoeditAvg.py M37New/object/R/Nov30103a.fits-daoedit
```

```
sky sigma mean is 8.19
```

```
FWHMPSF mean is 5.03
```

- 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"
```

- 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 (busca for maximos locales por encima de un valor configurable a través del parámetro findpars.threshold) calcula el perfil de intensidad y los escribe en un fichero. XCENTER y YCENTER se ajustan para cuadrar con el perfil de intensidad definidos por los parametros datapars.fwhmpsf y datapars.sigma

```
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 que son entrada para la funcion phot que calcula las magnitudes teniendo en cuenta también los parámetros que especifican la apertura: photpars y fitskypars que se especifican una vez para todas las imagenes de la forma siguiente:

Aperture photometry

Cogemos una media de todos los valores medios fwhm para establecer el valor del parametro apertures que es recomendable no variar entre las imagenes

```
$ find M37New/ -iname *-daoedit -exec python daoeditAvg.py {} \; | grep FW | cut -d' ' -f4
5.91
5.90
6.13
6.01
5.90
5.83
5.36
5.64
5.95
5.97
```

La media de estos valores es 5.853 y la colocamos en photpars.apertures 17.56 (3 * 5.853) Los valores recomendados de annulus es el valor de apertures + 3 y dannulus 5

```
daophot> photpars.apertures = 17.56
daophot> fitskypars.annulus=20.56
daophot> fitskypars.dannulus=5
```

- Ejecutamos phot para cada imagen (teniendo los parametros datapars.fwhmpsf y datapars.sigma establecidos con el valor correspondiente)

```
daophot> phot M37New/object/R/Nov30103a.fits coords=M37New/object/R/Nov30103a.fits-daofind output=M3
```

- Los pasos anteriores (buscar estrellas y obtener magnitudes) se hacen de forma automática :

```
python daofind.py listCenterAligned-M37
```

que genera los ficheros *-daofind de todas las imagenes alineadas y los ficheros *.mag con las magnitudes

- 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"
```

- **Obtener la imagen PSF**

Cogemos una imagen de referencia (yo elegía la misma que he utilizado para alinear: M37New/object/V/Nov30098a.fits pero no creo que tiene que ser la misma) establecemos los valores de datapars.fwhmpsf y sigma a los valores correspondientes a esta imagen (como antes: el valor medio obtenido despues de seleccionar un par de estrellas con daoedit en la imagen correspondiente)

```
daophot>datapars.sigma=11.43
daophot>datapars.fwhmpsf=5.83
```

Como vamos a usar la misma psf para todas las imagenes para extraer las estrellas después con allstar, cogemos la media de todas las fwhmpsf de todas las imagenes para el valor del parámetro daopars.fitrad(=v) y 3*v+1 para daopars.psfrad

```
daophot>daopars.fitrad=5.853
daophot>daopars.psfrad=18.56
```

Ejecutamos psf con la opción interactive = yes , así elegimos despues las estrellas de cual calcular la psf(tenemos que tener la imagen en ds9 con display y despues ejecutar psf, pulsando 'a' dibuja el perfil radial de la estrella, después pulsando 'a' aceptamos la estrella , con 'd' la rechazamos: la estrella debería tener un perfil radial parecido a una gauss y no debería tener vecinos : los vecinos son las estrellas a una distancia menor de 1.5 * psfrad / scale + 2 * fitrad / scale + 1)

```
daophot>psf.image="M37New/object/V/Nov30098a.fits"
daophot>psf.photfile="M37New/object/V/Nov30098a.fits.mag"
daophot>psf.psfimage="M37New/object/V/Nov30098a.fits.psf"
daophot>psf.opstfile="M37New/object/V/Nov30098a.fits.pst"
daophot>psf.groupfile="M37New/object/V/Nov30098a.fits.psg"
daophot>psf.interactive=yes
daophot>psf
```

- Para todas las imagenes ejecutamos allstar como se describe en el siguiente paso para la imagen que elegí como referencia: (usamos la misma psf para todas las imagenes, pero cambiamos sigma y fwhmpsf con los valores medios correspondiente para cada imagen)

```
daophot>allstar.image="M37New/object/V/Nov30098a.fits"
daophot>allstar.photfile="M37New/object/V/Nov30098a.fits.mag"
daophot>allstar.psfimage="M37New/object/V/Nov30098a.fits.psf.fits"
daophot>allstar.allstarfile="M37New/object/V/Nov30098a.fits.als"
daophot>allstar.rejfile="M37New/object/V/Nov30098a.fits.arj"
daophot>allstar.subimage="M37New/object/V/Nov30098a.fits.sub"
daophot>datapars.sigma=11.43
daophot>datapars.fwhmpsf=5.83

daophot>allstar
```

- Automatización para todas las imagenes:

```
$ python allstar.py listCenterAligned-M37 /scratch1/tobs/M37New/object/V/Nov30098a.fits.psf.fits
```

listCenterAligned-M37 es la lista de todas las imagenes del centro del cúmulo alineadas y el segundo parámetro es la imagen psf obtenida en el paso anterior

- Para las estrellas de calibración (elegimos de momento "pg2213", "pg2331", "pg0918", "pg0231" porque phl1027 no está en: <http://www.cfht.hawaii.edu/ObsInfo/Standards/Landolt/>) y separamos las imagenes para cada una con

```
$ python createStdLists.py
```

como observamos que hay varias imagenes para el mismo objeto tenemos que rehacer todo el proceso de fotometría como en el caso del cúmulo empezando con elegir una imagen de referencia, centrar, alinear , ...

- elegimos como imagen referencia para pg0231 M37New/object/V/Nov30047.fits y creamos el fichero coords-47.txt
- elegimos como imagen referencia para pg0918 M37New/object/V/Nov30090.fits y creamos el fichero coords-90.txt
- elegimos como imagen referencia para pg2213 M37New/object/V/Nov30055.fits y creamos el fichero coords-55.txt
- elegimos como imagen referencia para pg2331 M37New/object/V/Nov30043.fits y creamos el fichero coords-43.txt

```
ec1>imcentroid input="M37New/object/V/Nov30047.fits" reference="M37New/object/V/Nov30047.fits" coords="coords-47.txt"
ec1>imcentroid input="M37New/object/V/Nov30090.fits" reference="M37New/object/V/Nov30090.fits" coords="coords-90.txt"
ec1>imcentroid input="M37New/object/V/Nov30055.fits" reference="M37New/object/V/Nov30055.fits" coords="coords-55.txt"
ec1>imcentroid input="M37New/object/V/Nov30043.fits" reference="M37New/object/V/Nov30043.fits" coords="coords-43.txt"
```

- ejecutamos imalign para alinear todas las imagenes segun la imagen de referencia para cada objeto pg

```
ec1>imalign.interp_type = "nearest"
ec1>imalign input="@listCenter-pg0231" output="@listCenterAligned-pg0231" reference="M37New/object/V/Nov30047.fits" coords="coords-47-aligned.txt"
ec1>imalign input="@listCenter-pg0918" output="@listCenterAligned-pg0918" reference="M37New/object/V/Nov30090.fits" coords="coords-90-aligned.txt"
ec1>imalign input="@listCenter-pg2213" output="@listCenterAligned-pg2213" reference="M37New/object/V/Nov30055.fits" coords="coords-55-aligned.txt"
ec1>imalign input="@listCenter-pg2331" output="@listCenterAligned-pg2331" reference="M37New/object/V/Nov30043.fits" coords="coords-43-aligned.txt"
```

- python readAlsFiles.py --listFiles=listCenterAligned-M37
- python makeCharts.py --listFiles=listCenterAligned-M37

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
```