Reducción Polarimétrica de CASPOL con IRAF

Ricardo Gil-Hutton
CASLEO - CONICET

Octubre 2013

1 Introducción:

CASLEO dispone de una unidad polarizadora llamada CASPOL que puede ser usada en el telescopio "Jorge Sahade" junto con un detector CCD para realizar polarimetría de área.

El CASPOL fue construído en CASLEO con finaciamiento propio y su diseño es similar al IAGPOL (Magalhaes et al. 1996, ASP Conf. Series 97, 118) o al polarímetro del telescopio de 2.2 m. de la Universidad de Hawaii (Masiero et al. 2007, PASP 119, 1126). El instrumento consiste en una unidad que contiene un obturador mecánico, una rueda de filtros, una regleta de filtros neutros, una lámina retardadora y una placa de Savart, y que se instala delante de un detector CCD. La lámina retardadora puede rotar en pasos de 22.5 grados, y la placa de Savart produce dos imágenes polarizadas ortogonalmente separadas 0.9 mm (10.2 segundos de arco en el cielo). Tanto la retardadora como el Savart poseen coating antireflectante para el rango de 400 a 800 nm.

El software de control permite controlar todas las piezas móviles de CASPOL. La versión actualmente en uso es una versión de prueba pero que mantiene todas las funcionalidades necesarias para su operación (ver Fig. 1). El software muestra una pantalla con cuatro regiones: la primera permite posicional la lámina retardadora entre 0 y 360 grados en pasos de 22.5 grados, la segunda controla la rueda de filtros UBVRI, la tercera controla la regleta de filtros neutros y la cuarta la posición del Savart.

La adquisión de imágenes se maneja con el programa de control del detector en uso y el procedimiento a seguir es el mismo que el utilizado para adquirir imágenes con CCD directo.

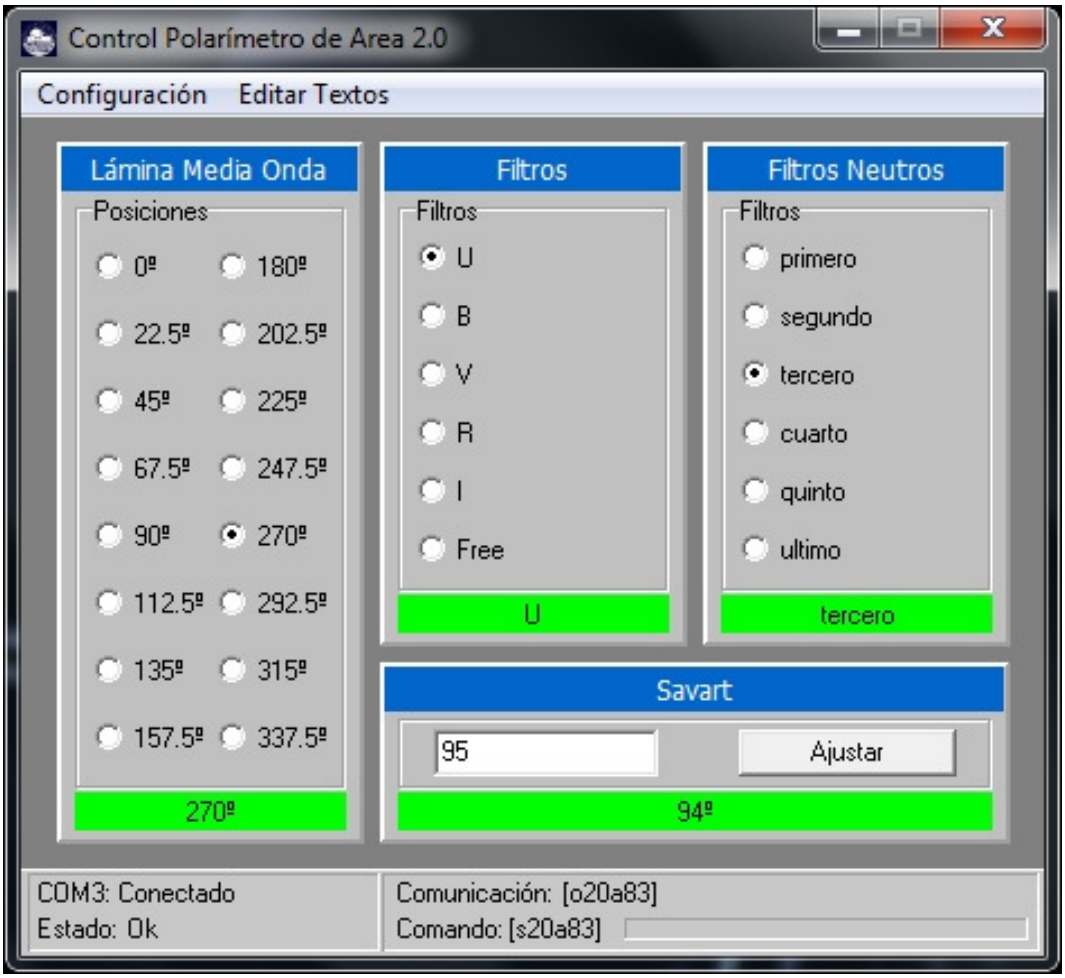

Figure 1: Pantalla de control de CASPOL

En este instructivo se presenta una breve descripción de cómo realizar la reducción de las imágenes para extraer información polarimétrica utilizando IRAF y una serie de tareas disponibles en el paquete *CASLEO*.

2 Instalación del paquete CASLEO:

El paquete de IRAF CASLEO se encuentra disponible en la sección de *Instrumental* de la página web. Este paquete contiene tareas que se utilizan en la reducción de CASPOL pero próximamente se agregarán más tareas útiles para la reducción y análisis de otro tipo de observaciones.

Los pasos a seguir para la instalación del paquete son:

- bajar y descomprimir el paquete. Si asumimos que el subdirectorio donde se va a descomprimir es */home/rgh/casleo/iraf/*, al descomprimir se crea un subdirectorio *casleo* que contiene al paquete.
- en el directorio raiz del usuario, agregar al archivo *loginuser.cl*:

```
# subdirectorio donde estan los archivos de
# configuracion instrumental
#
set instrdir = "/home/rgh/casleo/iraf/casleo/instr/"

# subdiretorio donde esta el paquete CASLEO
#
set pkgdir = "/home/rgh/casleo/iraf/casleo/"

# define el paquete
#
task $casleo = pkgdir$casleo.cl
reset helpdb = (envget("helpdb") //
         ",pkgdir$casleodb.mip")
keep
```

- entrar a IRAF y cargar el paquete *casleo*:

```
      NOAO/IRAF PC-IRAF Revision 2.16 EXPORT Fri Apr 13 10:39:09 MST 2012
         This is the EXPORT version of IRAF V2.16 supporting PC systems.


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

    Visit http://iraf.net if you have questions or to report problems.

    The following commands or packages are currently defined:

  Initializing SAMP .... No Hub Available

        dataio.     language.   obsolete.   softools.   vo.
```

```
      dbms.       lists.      plot.       system.
      images.     noao.       proto.      utilities.

vocl>
vocl> casleo

               ****************************************
               *   Complejo Astronomico El Leoncito   *
               *                                      *
               *        v1.0 -- Noviembre 2012        *
               ****************************************


      mascara   mkmasc    ordenar   polext    polred    registra

casleo>
```

Las tareas que por ahora están disponibles en el paquete son:

1. mkmasc – crea una mascara binaria para CASPOL a partir de un flat.
2. mascara – aplica una mascara a imagenes de CASPOL.
3. ordenar – re-ordena en pares ordinaria - extraordinaria el listado de estrellas encontrado.
4. polext – procedimiento para extraer de los archivos de fotometria los valores necesarios para hacer una reduccion con POLRED.
5. polred – procedimiento para ejecutar el programa de reduccion POLRED desde IRAF.
6. registra – encuentra los desplazamientos entre imagenes.

3 Adquisición de las imágenes CCD:

Para reducir las imágenes de CASPOL se deben obtener algunos *bias* y *flat* del mismo modo que cuando se trabaja con CCD directo. Los frames de objetos adquiridas con CASPOL contienen en su cabecera algunos keywords que permiten identificar la posición de la lámina retardadora, el Savart y los filtros. Por ejemplo:

```
casleo> imhea originales/obj128 l+
originales/obj128[1034,1024][short]: hd 208205
No bad pixels, min=0., max=0. (old)
Line storage mode, physdim [1034,1024], length of user area 972 s.u.
Created Wed 15:37:39 07-Nov-2012, Last modified Thu 15:14:24 18-Oct-2012
Pixel file "originales/obj128.fit" [ok]
OBSERVAT= 'CASLEO              '  / observatory
DATE-OBS= '2012-10-19          '  / date (yyyy-mm-dd) of obs.
UT      = 00:33:00.7              / universal time
ST      = 21:47:20.3              / sidereal time
HA      = -00:08:00.3             / hour angle
RA      = 21:55:20.5              / right ascension
DEC     = -01:40:57.3             / declination
EPOCH   =                2012.7   / epoch of ra y dec
ZD      =                  30.2   / zenith distance
AIRMASS =                  1.16   / airmass
EXPTIME =                 10.00   / integration time
TELESCOP= '2.15m               '  / telescope name
DETECTOR= 'TK 1K               '  / detector
GAIN    =                  1.98   / gain, electrons per adu
RDNOISE =                  7.40   / read noise
IMAGETYP= 'object              '  / object, dark, zero, etc.
HWPLATE = '22.5 '
FILTER  = 'V '
NEUTRAL = '0 '
SAVART  = '0 '
OBJECT  = 'hd 208205 '
OBSERVER= 'Para R. GilHutton '
INSTRUME= 'CASPOL '
COMMENT = 'Filtro V, R, I '
casleo>
```

en esta cabecera se indica que la imagen fue adquirida con filtro V, con la lámina retardadora en 22.5 grados, sin filtro neutro y con el Savart en una posición de 0 grados.

Para facilitar la reducción posterior es recomendable adquirir frames de un mismo objeto en por lo menos cuatro posiciones sucesivas de la lámina retardadora. Por ejemplo, 0, 22.5, 45 y 67.5 grados, o 90, 112.5, 135 y 157.5 grados.

La placa de Savart es la responsable de generar las dos imágenes de cada objeto con polarización ortogonal. Si para facilitar la posterior identificación y reducción fuera conveniente que las imagenes ordinaria - extraordinaria aparezcan orientadas de otra manera es posible rotar el Savart en un ángulo arbitrario.

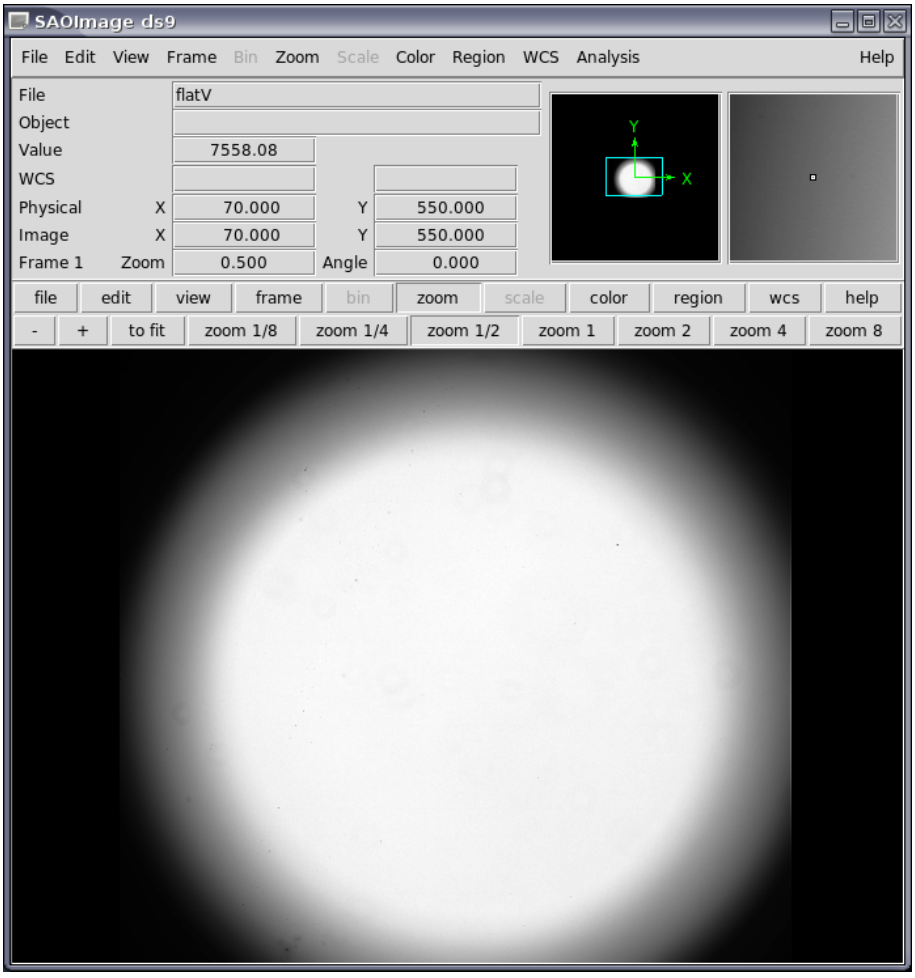

Figure 2: Flat tomado con CASPOL

4 Redución de las imágenes CCD:

Los frames adquiridas se reducen del mismo modo que los correspondientes a CCD directo utilizando el paquete *imred.ccdred*. La unica salvedad es que resulta conveniente identificar el instrumento CASPOL con la tarea *setinstrument*. Si el paquete *casleo* se cargó correctamente, se puede identificar el instrumento mediante:

```
ccdred>
ccdred>
ccdred>
ccdred> setinstrument ? site=casleo-js directory=instrdir$

direct          Current headers for direct CCD at CASLEO 2.15 m. Telescope
```

```
spect           Current headers for spectroscopy at CASLEO 2.15 m. Telescope
echelle         Current headers for Echelle at CASLEO 2.15 m. Telescope
caspol          Current headers for Caspol at CASLEO 2.15 m. Telescope

Instrument ID (type q to quit) (caspol):

...
```

Debido a que el conjunto filtro - regleta neutros - lámina retardadora - Savart produce un fuerte diafragmado de la imagen obtenida en el plano del detector, al finalizar la reducción básica se observa que los frames presentan una zona útil circular más o menos centrada mientras que los rincones se ven afectados por mucho ruido.

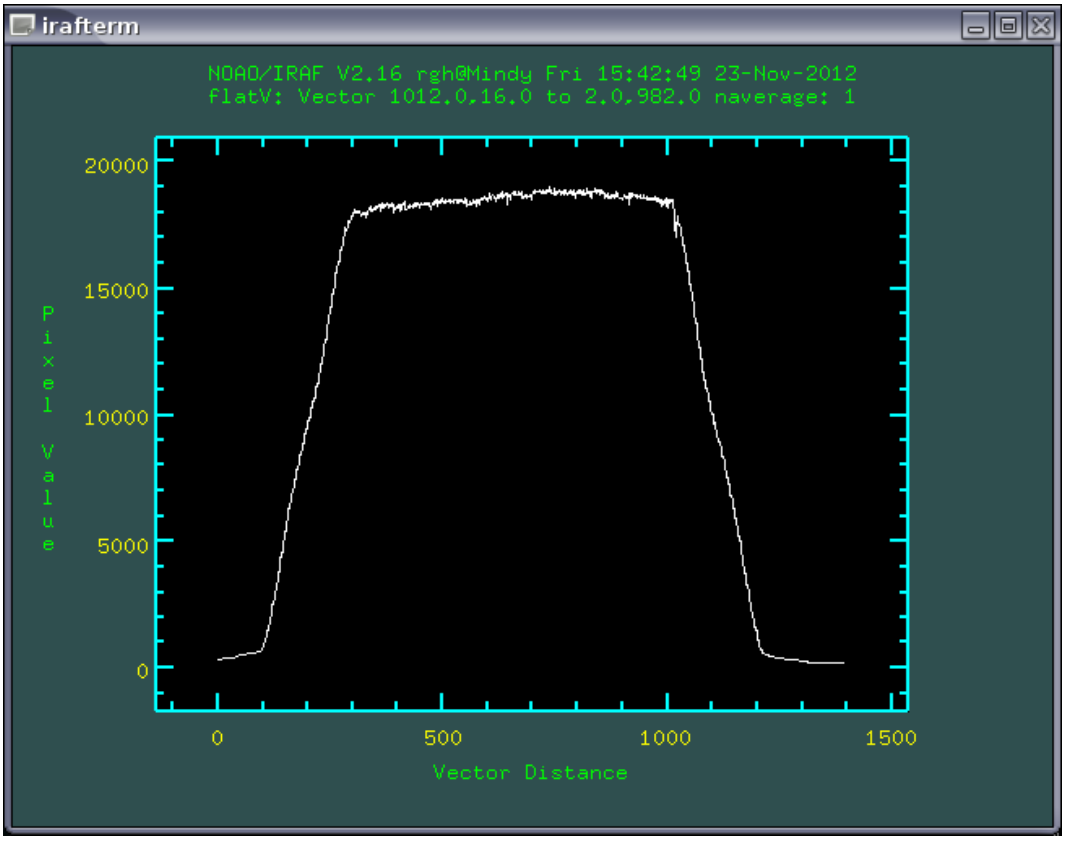

Figure 3: Corte diagonal del flat mostrado en la Figura 2

En la Figura 2 se muestra un flat donde se puede ver que la región utilizable es circular de unos 4 minutos de arco de diámetro (900 pixels si se utiliza el Tek1024) y cuyo centro esta corrido del centro de la imagen, y en la Figura 3 se muestra un corte de la imagen desde el rincón superior izquierdo hasta el inferior

derecho donde se puede ver que fuera de la zona central la imagen presenta una rápida caída a cero que al hacer la reducción por flat de los frames de objeto introduce ruido. Si bien esta región ruidosa no contiene información útil es conveniente eliminarla para facilitar la búsqueda e identificación de objetos y su reducción. Para esto se debe construir una máscara circular que defina la región útil y permita reemplazar la zona ruidosa por un valor constante (usualmente, cero).

Para construir la máscara se utiliza la tarea *mkmasc* del paquete *casleo*. Como frame de referencia se utiliza un flat y se debe definir el radio de la máscara, las coordenadas del centro aproximado y los lados de la región rectangular donde se buscará el centro real. Entonces:

```
ccdred>
ccdred> lpar mkmasc
       archin = "flatV"         reference flat image
      archout = "mask"          output mask
         (rad = 450.)           radius of useful region in image
          (cx = 512)            approx. mask center in X
          (cy = 512)            approx. mask center in Y
       (ladox = 10)             X side of the search area
       (ladoy = 10)             Y side of the search area
       (lista = "")
        (mode = "q")
ccdred>
ccdred>
ccdred> mkmasc flatV mask1 rad=450 cx=450 cy=530 ladox=10 ladoy=10
10% 20% 30% 40% 50% 60% 70% 80% 90% 100% - done
445 525 2160.887 100000000.
10% 20% 30% 40% 50% 60% 70% 80% 90% 100% - done
445 526 2167.829 2160.887
10% 20% 30% 40% 50% 60% 70% 80% 90% 100% - done
...
...
...
10% 20% 30% 40% 50% 60% 70% 80% 90% 100% - done
455 533 2119.125 2058.857
10% 20% 30% 40% 50% 60% 70% 80% 90% 100% - done
455 534 2127.084 2058.857
10% 20% 30% 40% 50% 60% 70% 80% 90% 100% - done
455 535 2135.135 2058.857
10% 20% 30% 40% 50% 60% 70% 80% 90% 100% - done

Centro de la mascara: X=455  Y=525
ccdred>
```

La tarea busca en un rectángulo de *ladox x ladoy* centrado en *cx* y *cy* la

máscara de radio *rad* que aplicada al frame de referencia (un flat) maximiza la señal.

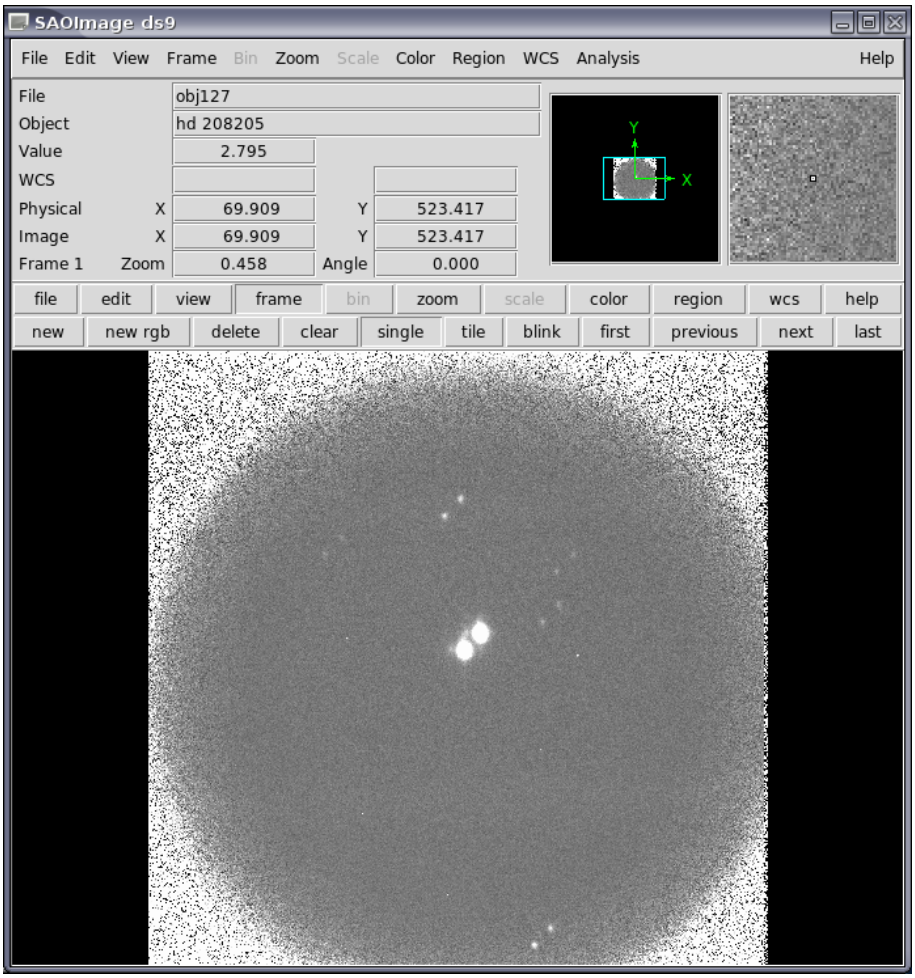

Figure 4: Imagen de objeto adquirida con CASPOL antes de aplicar la máscara.

Luego de encontrar la máscara más conveniente, hay que aplicarla a los frames de objeto mediante la tarea *mascara*. Para aplicar esta tarea es conveniente hacer una lista de entrada con los frames a procesar, otra de salida con los nombres que se le quiere asignar a los frames procesadas, y un archivo con el valor para cada imagen con el cual se quiere reemplazar las zonas recortadas. Particularmente, si se quiere utilizar un mismo valor para todas las imágenes se puede ingresar este valor directamente. Entonces:

```
ccdred>
ccdred> lpar mascara
       archin = "@lista"        list of images to apply the mask
```

```
      archout = "@mslista"      list of output images
         mask = "mask"          mask name
        (value = 0.)            constant pixel value to use out of the mask
      (lvalue = )               list of constant pixel values to use out of the
        (flist = no)            do you use a list of constant pixels for each i
        (lista = "")
      (lista1 = "")
      (lista2 = "")
        (mode = "q")
ccdred>
ccdred> mascara @lista @mslista mask1 value=0. flist-
10% 20% 30% 40% 50% 60% 70% 80% 90% 100% - done
...
...
ccdred>
```

En la figura 4 se muestra un frame de objeto luego de la reducción básica y en la figura 5 se observa el mismo frame después de aplicar la corrección por la máscara.

Al llegar a este punto los frames de objeto adquiridos con CASPOL están listos para que se extraiga la información polarimétrica.

5 Reducción polarimétrica:

Debido a que el Savart produce dos imágenes con polarización ortogonal por cada estrella el tratamiento que se le debe dar a los frames producidos por CASPOL es diferente dependiendo del número de objetos que aparezcan en ellas. En lo que sigue daremos un ejemplo de procesamiento para un frame que contiene más de un objeto de interés y se necesita obtener la polarimetría para todos ellos. Los casos de objetos únicos (standards, asteroides, etc.) pueden ser tratados de la misma manera o procesados manualmente.

Para poder automatizar el proceso de reducción es necesario obtener las coordenadas de los objetos en el frame. Con un número de $2 * n$ imágenes en el frame para n estrellas, lo más práctico es utilizar la tarea *daofind* del paquete *digiphot.daophot* para encontrar sus coordenadas (X,Y). Si bien es posible realizar esta búsqueda para todas las imágenes utilizando *daofind* con una lista como "imagen de entrada", se puede aprovechar que las estrellas mantienen sus posiciones relativas en imágenes sucesivas y realizar la búsqueda solo en una de ellas.

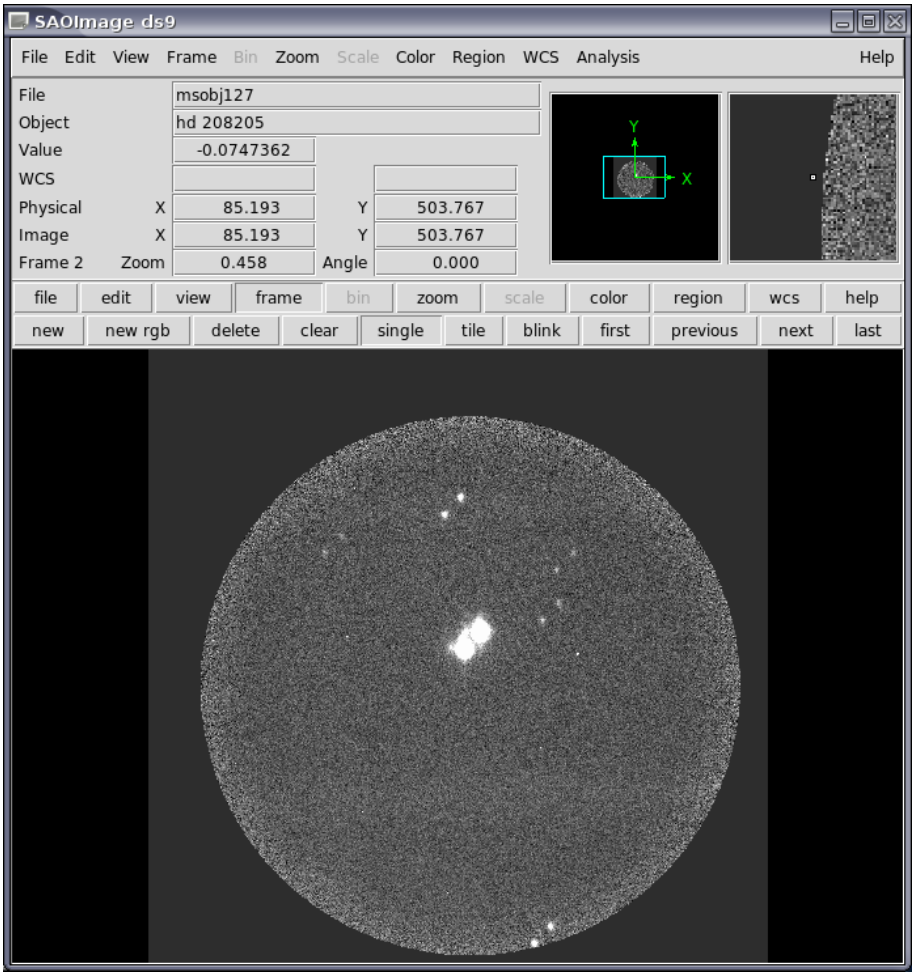

Figure 5: el mismo frame de la figura 4 después de aplicar la máscara.

Por ejemplo, si se quiere buscar objetos en *msobj127.fit* se debe:

1. desplegar el frame y elegir una región del background libre de objetos.
2. encontrar el valor medio de esta región mediante la tarea *imstatistics*:

```
daophot> imstat msobj127[300:400,200:300]
#               IMAGE           NPIX      MEAN     STDDEV       MIN       MAX
 msobj127[300:400,200:300]     10201     3.482      1.992    -2.137     90.58
daophot>
```

3. luego de obtener el valor medio para el cielo y conocido el readout noise y ganancia del detector (están en la cabecera de cada imagen) se puede

calcular el sigma del background con:

$$\sigma = \frac{\sqrt{ganancia * cielo + rdnoise^2}}{ganancia}$$

4. utilizando *imexamine* encontrar el FWHM de una imagen estelar (9 pixels para *msobj127*).

5. finalmente, utilizar *daofind* para valores del threshold razonables, por ejemplo:

```
daophot>
daophot> daofind msobj127 fwhm=9 ccdread=rdnoise gain=gain exposure=exptime filter=filter
         sigma=3.96 threshold=5.
Output coordinate file(s) (default: image.coo.?) (default):

FWHM of features in scale units (9.) (CR or value):
        New FWHM of features: 9. scale units  9. pixels
Standard deviation of background in counts (3.96) (CR or value):
        New standard deviation of background: 3.96 counts
Detection threshold in sigma (5.) (CR or value):
        New detection threshold: 5. sigma 19.8 counts
...
...
Image: msobj127.fit  fwhmpsf: 9.  ratio: 1.  theta: 0.  nsigma: 1.5

  494.97   749.20   -1.243   0.406   0.097  0.308      1
  521.97   778.10   -1.008   0.528   0.195  0.576      2
  527.71   525.84   -6.844   0.434   0.140  0.351      3
  554.70   554.65   -6.648   0.420   0.185  0.568      4
  644.69    33.74   -1.396   0.539  -0.183  0.148      5
  658.02   572.67   -0.113   0.464   0.054  0.386      6
  671.56    61.86   -1.105   0.809  -0.099  0.305      7
  682.10   656.50   -0.199   0.484   0.016  0.450      8
  708.78   685.61   -0.048   0.495   0.025  0.361      9

threshold: 19.8 relerr: 0.523  0.2 <= sharp <= 1.  -1. <= round <= 1.
daophot>
```

Los resultados se guardan en un archivo *msobj127.coo.1*. Se puede observar que *daofind* encuentra un número impar de objetos, cosa que no es posible debido a que cada estrella debe mostrar dos imágenes. Esto puede ser el resultado de alguna componente que cae fuera de la región útil del frame, ruido que es detectado por *daofind* como un objeto, etc.

Para poder realizar la reducción es necesario que las imágenes ordinaria y extraordinaria de cada estrella aparezcan una a continuación de la otra. Para lograr

esto es necesario reordenar el listado de coordenadas encontrado por *daofind*. Para ello se sigue el siguiente procedimiento:

1. con la imagen desplegada se identifican las imágenes ordinaria y extraordinaria de algún objeto y se lo identifica en el listado de coordenadas. Particularmente, las imágenes del objeto más brillante en *msobj127* son las identificadas con los números 3 y 4 en la salida mostrada más arriba.

2. se calcula las diferencias en coordenadas X e Y de estas dos imágenes restando las coordenadas del objeto más a la izquierda de la posición del objeto más a la derecha ($\Delta X = 26.99$; $\Delta Y = 28.81$).

3. se ejecuta la tarea *ordenar* del paquete *casleo* sobre el archivo de coordenadas creado por *daofind* o cualquier otra tarea que liste coordenadas X e Y en las dos primeras columnas:

```
daophot>
daophot> lpar ordenar
       archin = "msobj127.fit.coo.1" coordinate file
      archout = "msobj127.ord"  output file
          (sx = 38.387)         x-axis offset
          (sy = 0.)             y-axis offset
          (dx = 2.)             x-axis error
          (dy = 2.)             y-axis error
       (lista = "")
        (mode = "q")
daophot>
daophot> ordenar msobj127.coo.1 msobj127.ord sx=26.99 sy=28.81
Pair: 1  2
Pair: 3  4
Pair: 5  7
Pair: 8  9
Total pairs: 4
Total stars: 9

Lonely stars:
658.018  572.675  -0.113  6

daophot>
```

La tarea *ordenar* encuentra los pares de imágenes correspondientes a cada estrella, identifica las estrellas sin par detectado y crea un nuevo archivo con el orden correcto. Como este archivo es válido para cualquier frame si se lo corrige

por el offset correspondiente, es posible utilizarlo para todos los frames si se encuentra el corrimiento de un frame a otro.

Para eso se utiliza la tarea *registra* del paquete *casleo* que toma como entrada la lista de frames para los que se necesita archivos de coordenadas, una lista con los nombres de los archivos de coordenadas que se van a crear para cada frame, y el archivo encontrado en el paso anterior por *ordenar* y que contiene en el orden correcto las coordenadas para las imágenes de las estrellas en **el primer frame de la lista de entrada**. Entonces:

```
daophot>
daophot> cat mslista
msobj127.fit
msobj128.fit
msobj129.fit
msobj130.fit
...
...
daophot>
daophot> cat coolista
msobj127.coo.2
msobj128.coo.2
msobj129.coo.2
msobj130.coo.2
...
...
daophot>
daophot> registra @mslista @coolista msobj127.ord
z1=-3.541501 z2=15.62198

Mark a reference star...

<r> plots the radial profile
<a> save the coordinates
<q> to exit
```

La tarea despliega el primer frame y solicita que se marque una estrella cualquiera como referencia presionando **a**. Luego de marcar la estrella elegida, se sale presionando **q** y la tarea muestra las coordenadas del objeto y despliega el segundo frame solicitando que se marque la misma estrella:

```
Image: msobj127.fit Coord.: 527.73 525.94
z1=-3.814749 z2=15.91672

Mark the same reference star...
```

```
<r> plots the radial profile
<a> save the coordinates
<q> to exit
```

Al finalizar la tarea creó un archivo de coordenadas por cada frame donde los objetos están ordenados de tal modo que sus imágenes ordinaria y extraordinaria aparecen una a continuación de la otra. Para verificar que los objetos fueron bien identificados y que el archivo de coordenadas es correcto se puede hacer:

```
daophot>
daophot> display msobj130 1
z1=-2.653793 z2=14.84566
daophot>
daophot> tvmark 1 msobj130.coo.2 mark=circle radii=5 color=0
daophot>
```

que marcará un pequeño círculo arriba de las imágenes identificadas.

Como ya se dispone de la posición de cada imagen en el orden correcto es posible obtener valores de flujo para calcular los parámetros polarimétricos utilizando las tareas fotométricas de IRAF. Para ello se puede utilizar la tarea *phot* con los mismos parámetros de entrada que se utilizaron para *daofind*. Es importante considerar que la tarea *phot* también está disponible en el paquete *digiphot.apphot*, pero la configuración de los archivos de parámetros en ambos casos es levemente diferente asi que se sugiere usar siempre el paquete *digiphot.daophot*.

Como el FWHM encontrado es de 9 pixels, para hacer la fotometría se usan aperturas de 7, 9, 11, 13, 15, 17 y 19 pixels de radio y un anillo para estimar el cielo que comienza a 25 pixels del centro de la estrella y tiene un ancho de 10 pixels. Se recomienda no utilizar más de 20 aperturas porque ese es el límite de los programas de reducción polarimétrica. Entonces:

```
daophot>
daophot> phot @mslista fwhm=9 ccdread=rdnoise gain=gain exposure=exptime filter=filter
         sigma=3.96 apertures=7,9,11,13,15,17,19 annulus=25 dannulus=10 int-
Input coordinate list(s) (default: image.coo.?) (default):
Output photometry file(s) (default: image.mag.?) (default):

Centering algorithm (none) (CR or value):
        New centering algorithm: none
Sky fitting algorithm (mode) (CR or value):
        Sky fitting algorithm: mode
```

```
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 (10.) (CR or value):
        New width of the sky annulus: 10. scale units 10. pixels
File/list of aperture radii in scale units (7,9,11,13,15,17,19) (CR or value):
        Aperture radius 1: 7. scale units 7. pixels
        Aperture radius 2: 9. scale units 9. pixels
        Aperture radius 3: 11. scale units 11. pixels
        Aperture radius 4: 13. scale units 13. pixels
        Aperture radius 5: 15. scale units 15. pixels
        Aperture radius 6: 17. scale units 17. pixels
        Aperture radius 7: 19. scale units 19. pixels
Standard deviation of background in counts (3.96) (CR or value):
        New standard deviation of background: 3.96 counts
Minimum good data value (INDEF) (CR or value):
        New minimum good data value: INDEF counts
Maximum good data value (INDEF) (CR or value):
        New maximum good data value: INDEF counts

msobj127    494.97   749.20  3.248925   19.439   19.287   19.204   19.159   19.126   ...  ok
msobj127    521.97   778.10  3.432047   19.526   19.368   19.290   19.260   19.268   ...  ok
msobj127    527.71   525.84  5.212608   13.806   13.666   13.602   13.571   13.555   ...  ok
...
...
```

Phot genera un archivo **.mag.** para cada frame con la fotometría de los objetos cuyas coordenadas se encuentran en los archivos **.coo.** correspondiente a ese frame.

Con la información obtenida con la tarea *phot* se puede realizar finalmente una reducción polarimétrica de los objetos de interés. Para eso se utiliza la tarea *polext* del paquete *casleo* cuya función es extraer de los archivos **.mag.** creados los datos necesarios para hacer la reducción. Entonces:

```
daophot>
daophot> lpar polext
       archin = "*.mag.1"       photometry files
      archout = "hd208205.fot"  output file
       (naper = 1)              number of apertures
       (lista = "")
      (lista1 = "")
        (mode = "q")
daophot>
daophot> polext *.mag.1 salida.fot naper=7
daophot>
```

produce un archivo *salida.fot* que contiene los valores extraídos de los archivos **.mag.** para cada imagen ordinaria o extraordinaria correspondiente a los cam-

pos IMAGE, ID, MSKY, NSKY, RAPERT, SUM y AREA (los tres ultimos para todas las aperturas calculadas), y de la cabecera de los respectivos frames los keywords FILTER, HWPLATE, SAVART, DATE-OBS, y UT.

Este archivo creado por la tarea *polext* es el archivo final de entrada para el programa de reducción en FORTRAN denominado **POLRED.F90** (cuyo codigo se encuentra disponible en el subdirectorio del paquete *casleo*). Para poder ejecutar este programa desde IRAF se encuentra disponible la tarea *polred* que permite armar el archivo inicial de parámetros y correr el ejecutable para finalmente renombrar el archivo de salida con el nombre indicado por el usuario. Verifique siempre que el código *POLRED.F90* fue correctamente compilado antes de ejecutar la tarea *polred*:

```
daophot>
daophot> lpar polred
       archin = "hd208205.fot"  file with the observations
      archout = "hd208205-a.pol" output file
     (rdnoise = 7.4)            readout noise of the detector
        (gain = 1.98)           gain of the detector
       (naper = 4)              number of apertures used
       (qinst = 0.)             Q component of the instrumental polarization
       (uinst = 0.)             U component of the instrumental polarization
      (eqinst = 0.)             error in Q component of the instrumental polari
      (euinst = 0.)             error in U component of the instrumental polari
        (apos = 0.)             correction of the position angle of polarizatio
       (norma = yes)            do you want renormalization?
         (exe = "./polred")     polred executable file
        (mode = "q")
daophot>
daophot> polred salida.fot salida.pol naper=7 exe=/home/rgh/casleo/pol-imagen/reduccion/polred

Making parameter file...

Executing the reduction program...

 ----------------------------------------
 POLRED v1.0 - Reduction program for CASPOL
 November 2012
 ----------------------------------------

 Number of apertures:  7
 Images and HW:
 msobj127                       0.
 msobj128                       22.5
 msobj129                       45.
 msobj130                       67.5
 HW positions:  4
 Stars:  4
```

```
Copying the output file...
Done.
daophot>
```

El archivo de salida generado (*salida.pol*) contiene la siguiente información:

```
# POLRED v1.0 - Reduction program for CASPOL
#
# From: 2012-10-19  UT: 00:31:20.7
# To  : 2012-10-19  UT: 00:36:51.0
#
# Instr. Q (%)   :     0.000    0.000
# Instr. U (%)   :     0.000    0.000
# Instr. TH (deg):    0.000
#
#
#

# Star :   1

...
...

# Star :   2

 Aper    Q%      eQ%     U%      eU%    Pol%    ePol%   TH     eTH    eTeo%
  7.0  -0.328   0.067   0.739   0.067   0.809   0.034  56.95   1.21   0.132
  9.0  -0.263   0.005   0.764   0.005   0.808   0.003  54.50   0.12   0.124
 11.0  -0.244   0.030   0.753   0.030   0.791   0.020  53.97   0.71   0.120
 13.0  -0.233   0.033   0.747   0.033   0.783   0.021  53.66   0.79   0.119
 15.0  -0.224   0.026   0.746   0.026   0.778   0.017  53.34   0.64   0.119
 17.0  -0.213   0.025   0.743   0.025   0.773   0.017  53.01   0.64   0.119
 19.0  -0.205   0.029   0.746   0.029   0.773   0.020  52.68   0.74   0.120

# Star :   3

...
...
```

donde para cada estrella se lista para cada apertura utilizada las componentes Q y U con sus errores, la polarización total y su error, el ángulo de posición y su error y un error teórico estimativo calculado como $(S/N)^{-1}$.