

# NTOFOCUS

Rafael González Fuentetaja

21 de marzo de 2006



# Índice general

<b>1. Introducción</b>	<b>9</b>
1.1. Motivación . . . . .	9
1.2. Objetivo . . . . .	10
1.3. Instalación . . . . .	11
<b>2. Proceso de calibración</b>	<b>13</b>
2.1. Inicialización . . . . .	14
2.2. Movimiento del enfocador . . . . .	14
2.3. Toma de imagen . . . . .	14
2.4. Establecer zonas de análisis . . . . .	14
2.5. Cálculo del HFD . . . . .	15
2.6. Acumular medición al vector de medidas . . . . .	16
2.7. Ajuste de la $V$ . . . . .	17
2.8. Controlando el ajuste . . . . .	17
2.8.1. Puntos discordantes . . . . .	18
2.8.2. Tolerancia del ajuste . . . . .	18
2.8.3. Nivel de “Near Focus” . . . . .	19
2.8.4. Examinando el ajuste . . . . .	19
2.8.5. Un ejemplo . . . . .	20
2.9. Guardando la calibración . . . . .	21
2.9.1. El fichero de calibraciones . . . . .	21
2.9.2. Otras operaciones sobre el fichero de calibraciones . . . . .	22

<b>3. Proceso de re-enfoque</b>	<b>23</b>
3.1. Inicialización . . . . .	23
3.2. Filosofía de operación . . . . .	24
3.3. Alejamiento del foco de la posición actual . . . . .	25
3.4. Comprobación del lado de la curva . . . . .	26
3.5. Acercamiento al punto de nivel “Near Focus” . . . . .	26
3.6. En el punto de “Near Focus” . . . . .	27
3.7. En el punto de enfoque óptimo . . . . .	27
<b>4. Auditoría</b>	<b>29</b>
4.1. Registro de actividad . . . . .	29
4.2. Datasets . . . . .	30
<b>5. Análisis</b>	<b>31</b>
5.1. Análisis de la secuencia de calibración . . . . .	31
5.2. Análisis de imágenes . . . . .	34
5.2.1. Donut demasiado grande . . . . .	35
5.2.2. Sólo fondo de cielo . . . . .	37
5.2.3. Reguerillos de luz . . . . .	37
<b>6. Algoritmos</b>	<b>39</b>
6.1. El algoritmo HFD . . . . .	39
6.2. Otros algoritmos . . . . .	40

# Índice de cuadros

3.1. Acciones de re-enfoque . . . . .	25
4.1. Niveles de trazado . . . . .	29



# Índice de figuras

5.1. Secuencia de calibracion 1 . . . . .	32
5.2. Calibracion . . . . .	33
5.3. Imagen correcta . . . . .	34
5.4. Marginales recortadas . . . . .	35
5.5. Fondo de cielo . . . . .	36
5.6. Distribuciones de fondo de cielo . . . . .	37
5.7. Reguerillos de luz . . . . .	38





# Capítulo 1

## Introducción

### 1.1. Motivación

En el objetivo de proporcionarnos nuestras propias herramientas de control de telescopios y CCD, era lógico que tarde o temprano se abordara el problema de automatizar el enfoque. Los métodos que habíamos probado eran o bien manuales (poner y quitar máscaras de enfoque) o bien algo burdos (medir el brillo de un pixel), y por ello, imposibles de incorporar en un control remoto.

Entonces, allá por el año 2001, apareció el método de enfoque HFD, que parecía bastante sofisticado pero automatizable. La aplicación FocusMax estaba disponible de forma gratuita. Por desgracia para nosotros, ésta se controlaba con ASCOM y además funcionaba con enfocadores que cumplan con la interfaz ASCOM.

Por ello, Cristóbal García - conocido gurú español de las CCDs - me animó a “hincarle el diente” al algoritmo HFD para poder integrarlo en su aplicación ACOR. Empecé a tirar las primeras líneas de código en Octubre del 2004 y, para las Navidades de 2004, ya tenía un primer prototipo. Por supuesto, cascaba por doquier con la primera serie de pruebas reales. El año 2005 ha sido un año de hibernación para NTOFOCUS, puesto que el principal “Beta Tester”, Cristóbal, estaba muy ocupado con otros proyectos. Sin embargo, a principios de 2006, se han reanudado las pruebas, he corregido bastantes errores y he aprendido bastante al analizar las imágenes obtenidas.

No se puede decir que NTOFOCUS esté terminado, porque las cosas del software nunca se terminan, pero ya está bastante usable. A este respecto, no tengo más remedio que citar la obligada letanía:

*El software no está libre de fallos. Por ello, el uso de NTOFO-*

*CUS es bajo riesgo propio. No se admitirán responsabilidades derivadas del uso del mismo.*

Un motivo para escribir este manual es dar la sensación de madurez a NTOFOCUS, pero la verdad es que ha sido más bien para aprender L<sup>A</sup>T<sub>E</sub>X. Así que es posible que este manual se quede pronto desfasado respecto al desarrollo.

## 1.2. Objetivo

NTOFOCUS es una librería dinámica para el enfoque automático del telescopio, siguiendo el método HFD de Larry Weber y Steve Brady. El capítulo 6 ofrece una pequeña descripción de este algoritmo, pero sin duda, la fuente más adecuada del mismo es el artículo [Imaging the Sky 2001](#) que se presentó en Oregon el 2 de Noviembre de 2001.

Esta librería no actúa sobre ningún dispositivo de enfoque ni tiene interfaz gráfica. Es simplemente el motor de cálculo de dos procesos bien diferenciados:

- Proceso de calibración.
- Proceso de re-enfoque.

Por ello, creo que es bastante flexible a la hora de integrarla con una aplicación. Cualquier aplicación puede hacer uso de esta librería. En la actualidad, sólo [la aplicación ACOR de Cristóbal García](#) hace uso de ella. En un futuro muy, muy lejano, pienso rehacer esta librería para Linux. Al ser un motor de cálculo, no utiliza mecanismos específicos de Windows y debería ser muy portable. Pero antes, *tendré que disponer de un observatorio* y además, que éste funcione con Linux.

El tipo de dato del píxel manejado por NTOFOCUS es un entero de 16 bits con signo. Por ello, sólo puede emplearse directamente en CCDs compatibles con este tipo de dato como la Audine.

Amigo lector: Antes de pasar al resto del documento, sólo quiero pedirte que si incorporas esta librería a tu desarrollo<sup>1</sup>, se agradecería una breve reseña tanto a su uso como al autor.

En los capítulos siguientes, se explican con más detalle estos procesos.

---

<sup>1</sup> ¿ Y por que si nó ibas a estar leyendo este documento?

## 1.3. Instalación

La instalación es muy sencilla: simplemente desempaquetar el fichero .zip en la aplicación que desea usar NTOFOCUS. El paquete tiene los siguientes ficheros:

- ntofocus.h
- ntofocus.dll
- ntofocus.lib
- ntofocus.pdf
- VShape.R
- Aperture.R
- Sky.R
- R/foco9\_001.txt
- R/foco9\_002.txt
- R/foco9\_... .txt
- R/foco9\_019.txt
- R/foco9\_020.txt
- R/MarginalX.txt
- R/MarginalY.txt
- R/FeaturesX.txt
- R/FeaturesY.txt
- R/Centroid.txt
- R/Sky.txt
- R/VShapePoints.txt

La librería se ha desarrollado en *Borland Builder C++ 6*, pero *en teoría*<sup>2</sup> es utilizable en cualquier otro entorno de desarrollo.

---

<sup>2</sup>Ya sabemos que en teoría, la teoría y la práctica es lo mismo, pero en la practica no lo es.



## Capítulo 2

# Proceso de calibración

El proceso de calibración trata de sacar la curva en forma de V de todo el conjunto telescopio-CCD-enfocador. La forma de esta “V” es constante e independiente de la temperatura, mientras se mantenga la configuración óptica. Para más detalles, conviene leer el fundamento teórico del algoritmo.

El proceso de calibrado de la V emplea las siguientes funciones:

- *void AF\_InitCalibration(void);*
- *void AF\_SetImage(pixel\_t\* img, int width, int height);*
- *void AF\_SetSkyRegion(int x1, int y1, int x2, int y2);*
- *void AF\_SetCentroidRegion(int x1, int y1, int x2, int y2);*
- *double AF\_HFD(const char\* file);*
- *void AF\_AddMeasurement(int focuserPos, double hfd);*
- *int AF\_FitParams();*
- *int AF\_GetFitResult(FitResult\* fr);*
- *void AF\_Save(const char\* label);*

Tras la inicialización, una vez centrada una estrella adecuada, la aplicación debe realizar un proceso repetitivo de:

1. Movimiento del enfocador.
2. Toma de imagen (y guarda opcionalmente a fichero FITS.)
3. Establecer zonas de análisis.

4. Calcular el HFD en esas condiciones.
5. Acumular medición al vector de medidas.
6. Si la aplicación no ha acabado de tomar medidas, volver a 1.)

Cuando ya se han acabado de hacer medidas, entonces se realiza el ajuste de parámetros y se guardan los resultados a un fichero. A continuación, se describen estos pasos.

## 2.1. Inicialización

La función *AF\_InitCalibration()* inicializa a la librería para el proceso de calibrado. La aplicación deberá llamarla al principio del mismo.

```
AF_SetLogLevel(4);    /* nivel de log = depuracion */  
AF_EnableDataSets(1); /* exporta data sets de análisis */  
AF_InitCalibration();
```

El uso de *AF\_SetLogLevel()* y *AF\_EnableDataSets()* está descrito en el capítulo 4.

## 2.2. Movimiento del enfocador

Este punto es de total responsabilidad de la aplicación. NTOFOCUS no está preparado para tratar errores de posición del enfocador, así que hay que mover el enfoque evitando las temidas holguras (backlash).

## 2.3. Toma de imagen

Este punto es de total responsabilidad de la aplicación.

## 2.4. Establecer zonas de análisis

La aplicación debe proporcionar a NTOFOCUS el puntero y dimensiones de la imagen recién adquirida, así como delimitar las zonas de medición para el fondo de cielo y del centroide de la estrella o “donut”.

Si la aplicación es interactiva y permite al usuario seleccionar recuadros, se pueden establecer zonas separadas. Sin embargo, la opción más sencilla es la de establecer un recuadro común para ambas zonas. Ese recuadro podría abarcar la zona central de la imagen.

Es importante tener en cuenta varias cosas:

- Que haya una sola estrella en el área de interés.
- Que la exposición produzca una estrella brillante, pero sin blooming.
- Que la exposición sea lo suficientemente larga como para promediar la turbulencia atmosférica.
- Si la zona del fondo de cielo es la misma que la de la estrella, que haya bastantes más píxeles de fondo de cielo que de estrella.

La siguiente función auxiliar - a definir por la aplicación - permite hacer la simplificación del recuadro común. Notar que, por sistema, se desechan la primera y última fila y la primera y última columna del área a analizar.

```
void AF_SetRegion(pixel_t* img, int width, int height)
{
    AF_SetImage(img, width, height);
    AF_SetSkyRegion(1, 1, width-2, height-2);
    AF_SetCentroidRegion(1, 1, width-2, height-2);
}
```

## 2.5. Cálculo del HFD

Una vez establecida la región de análisis, la función `AF_HFD()` realiza el cálculo del HFD de la estrella o “donut”. Esta función devolverá el HFD calculado. Este valor será negativo si se falla en el cálculo.<sup>1</sup>

Es bastante normal que para una imagen dada, el cálculo del HFD falle. Existen bastantes razones para ello:

- Fallos de precisión interna.
- Falta de una estrella.
- Estrella en una esquina.

---

<sup>1</sup>En realidad, devuelve -1.0, pero es mejor no confiar en el valor exacto de un número de coma flotante, por los errores de conversión. Cristóbal me ha comentado que ha visto valores de -0.9999.

- Imágenes turbulentas.
- Imágenes con blooming.
- Imágenes con “reguerillos”, propios de no utilizar obturador.
- Imágenes en blanco de la CCD Audine.
- Imágenes con sólo fondo de cielo.
- etc.

Para averiguar la causa del fallo, podemos recurrir al análisis de la imagen y el registro de actividad, descritos en el capítulo 4. Si a la función *AF\_HFD()* se le pasa un puntero no nulo, entonces en el fichero de registro de actividad se imprimirá el nombre del fichero FITS afectado en la medición.

La calibración (y el re-enfoque) necesita un número suficiente de imágenes. Por ello, para una posición dada, si el cálculo del HFD de una imagen falla, la aplicación debería tomar nuevamente otra imagen e intentarlo de nuevo. Por supuesto, es recomendable establecer un límite de reintentos. Si se pasa el número de reintentos, es síntoma de que algo extraño sucede.

## 2.6. Acumular medición al vector de medidas

Esta librería mantiene internamente un vector de medidas en el que va acumulando las parejas (*posicion, hfd*) medidas.

El uso típico es realizar este paso y los dos anteriores como en el ejemplo siguiente:

```
AF_SetRegion(imagen, width, height);
hfd = AF_HFD(nomFich);
AF_AddMeasurement(position, hfd);

if(hfd < 0) {

    /* decrementar el contador de reintentos por imagen */
    /* si el contador llega a cero, senalar error grave */
    /* y si no, volver a tomar otra imagen y medir */

} else {

    /* mover el telescopio hasta la siguiente posición */
    /* resetear el contador de reintentos por imagen */
```



```

/* y comenzar las mediciones en ese nuevo punto */
}

```

Es interesante señalar que se pueden (y se deben) realizar varias medidas de HFD con la misma posición del enfocador para tratar de promediar el valor y sobre todo evitar errores espúreos. Se recomienda realizar al menos 5 medidas por posición. NTOFOCUS internamente calcula la mediana de todos los HFDs de una misma posición.

La función *AF\_AddMeasurement()* descarta toda pareja (*posicion*, *hfd*) cuya *hfd* sea negativa por lo que no hay problema en invocarla sin comprobar el valor de *hfd*.

## 2.7. Ajuste de la V

Una vez terminada la acumulación de medidas, llega el momento de realizar el cálculo de la V. La curva característica es muy lineal en sus extremos, con pendientes simétricas, pero con un suave acodamiento en el centro.

Para hacer el ajuste óptimo de las dos zona lineales por mínimos cuadrados, es imprescindible conocer el nivel de HFD donde la curva deja de ser lineal para comenzar a acodarse. A este nivel se le conoce como nivel de “Near Focus”.

NTOFOCUS cuenta con un algoritmo para intentar establecer este nivel de “Near Focus” automáticamente. Si estamos de acuerdo en usar los valores por defecto, la aplicación deberá llamar simplemente a *AF\_FitParams()*. La sección 2.8.5 muestra un posible ejemplo de uso.

*AF\_FitParams()* devolverá 0 si todo ha ido bien o -1 si existe algún error (demasiados puntos discordantes o RSE no inferior al objetivo).

## 2.8. Controlando el ajuste

Es posible influir sobre el proceso de ajuste estableciendo otros valores distintos a los valores por defecto antes de llamar a *AF\_FitParams()*. Para ello, se pueden utilizar las siguientes funciones:

- *int AF\_SetMaxOutliers(int nrejected);*
- *void AF\_SetMaxRSE(double rse);*

- *void AF\_SetNearFocusHFD(double level);*

Una aplicación interactiva podría cambiar estos ajustes y llamar a *AF\_FitParams()* tantas veces como sea necesario hasta que el examen de los resultados sea satisfactorio.

### 2.8.1. Puntos discordantes

Es posible que durante el proceso de calibración, se produzcan algunos puntos en la parte exterior de la curva en V que, siendo erróneos, no hayan sido detectados. Se pueden reconocer inmediatamente que son discordantes porque su nivel de HFD es anormalmente bajo. NTOFOCUS examina por defecto los tres puntos exteriores de cada lado de la curva para detectar este efecto. Es posible variar el número de puntos examinados mediante una llamada a *AF\_SetMaxOutliers()*. Naturalmente, no todos los puntos examinados son rechazados.

Los puntos rechazados por discordantes no formarán parte del proceso de ajuste de las rectas de regresión.

Esta función devuelve -1 si el tamaño del vector de medidas es menor que  $2 * (\text{numero de puntos discrepantes}) + 1$

### 2.8.2. Tolerancia del ajuste

Una vez desechados los puntos extremos erróneos, la siguiente fuente de error en el ajuste de las rectas de regresión la constituye la parte central de la curva, que no es lineal. El algoritmo automático de ajuste de NTOFOCUS intenta descartar los puntos de la zona central que contribuyen a decrementar la calidad del ajuste. El parámetro elegido para medir la calidad del ajuste es el *error estándar de los residuos*, que se mide en píxeles.

Para ello:

- Determina el punto “codo” para delimitar la parte izquierda y derecha de la curva.
- Para cada lado de la V, y desde la posición del codo, se hacen sucesivos ajustes de mínimos cuadrados hasta que el error estándar de los residuos es menor o igual al límite pre-establecido. Se van despreciando puntos centrales en el proceso. Este método iterativo es bastante insensible a la posición inicial del “codo”.

NTOFOCUS utiliza por defecto un valor de 0.5 píxeles - determinado experimentalmente - como máximo error estándar de los residuos admisible

para el ajuste. Con valores más pequeños como 0.25 se corre el riesgo de despreciar muchos o todos los puntos<sup>2</sup>. Con valores mayores, no se rechazarán los puntos.

La tolerancia del ajuste se determina con la función

```
void AF_SetMaxRSE(double rse).
```

Como subproducto de este algoritmo, se obtiene además el nivel de “Near Focus”, que es el HFD del punto más bajo incluido en el ajuste de la recta de regresión. El nivel de “Near Focus” global es el máximo de los dos niveles de “Near Focus” calculados para cada lado.

### 2.8.3. Nivel de “Near Focus”

Si no estamos contentos con el nivel elegido automáticamente, la aplicación deberá llamar a *AF\_SetNearFocusHFD()*. El script VShape.R es capaz de leer un fichero auxiliar de mediciones producido y pintar un gráfico con las parejas (*posicion, hfd*). En esta gráfica se muestra cual es el nivel de “Near Focus HFD” elegido por NTOFOCUS. Esta misma gráfica nos es de ayuda para determinar un nivel a ojo.

Notar que el parámetro de tolerancia del ajuste sigue estando en vigor. Si queremos asegurar de que no se desprecia ningún punto más por encima de un valor dado elegido a ojo, se deberá ajustar la tolerancia a un valor muy elevado, como por ejemplo 10 píxeles.

### 2.8.4. Examinando el ajuste

Aunque los resultados del ajuste pueden obtenerse examinando el registro de actividad de NTOFOCUS o bien los data sets generados (ver 4) la aplicación también puede obtener de manera directa parte de estos datos para uso propio. Para ello, se utilizará la función *AF\_FitResult()*

```
FitResult result;
AF_GetFitResult(&result);
```

Esta función rellena una estructura *FitResult*, que contiene los siguientes campos:

```
typedef struct {
    int    bfp;    /* best focus position */
```

---

<sup>2</sup>menos tres

```

double pid;      /* position intercept difference */
double nfHFD;    /* near focus HFD */
struct {
    double slope;      /* estimated left slope */
    double slope_dev;  /* estimated left slope standard deviation */
    double x0;         /* estimated left x0 */
    double x0_dev;     /* estimated left x0 standard deviation */
    double rse;        /* residual standard error */
    int npoints;       /* points used for fitting the left side */
    int nreject;       /* points rejected on the left side */
} left;
struct {
    double slope;      /* estimated right slope */
    double slope_dev;  /* estimated right slope standard deviation */
    double x0;         /* estimated right x0 */
    double x0_dev;     /* estimated right x0 standard deviation */
    double rse;        /* residual standard error */
    int npoints;       /* points used for fitting the right side */
    int nreject;       /* points rejected on the right side */
} right;
} FitResult;

```

Se debe destacar el campo *bfp* que devuelve la posición del enfoque óptimo. Eso permite que el enfocador se sitúe en la posición de mejor foco nada más acabar el proceso de calibración y sin tener que pasar por el proceso de re-enfoque.

### 2.8.5. Un ejemplo

El siguiente fragmento de código muestra un ejemplo simple de uso:

```

int retorno;
FitResult result;

AF_SetMaxRSE(0.5);      /* opcional, 0.5 es el valor por defecto */
AF_SetMaxOutliers(3);   /* opcional, 3 es el valor por defecto */
AF_SetNearFocusHFD(9); /* opcional, es automático por defecto */

retorno = AF_FitResult();
if(retorno != -1) {

    AF_Save("ejemplo");
    AF_GetFitResult(&result);
}

```

```
mover_enfoque(result.bfp);  
  
} else {  
  
    /* se ha producido un error en el ajuste */  
    /* mirar el log */  
}
```

El ejemplo muestra cómo se llaman a las funciones opcionales que controlan el ajuste por mínimos cuadrados de las dos partes de la curva en V.

## 2.9. Guardando la calibración

Una vez satisfechos con el resultado del ajuste con *AF\_FitParams()*, se procederá a guardar los parámetros calculados en un fichero de calibraciones.

La función *AF\_Save()* guardará la calibración recién efectuada en un fichero. A esta calibración se le pasa una etiqueta simbólica de hasta 16 caracteres<sup>3</sup>, que servirá como mnemónico. El hecho de guardar una calibración hace que esta calibración sea considerada la *calibración por defecto*. La calibración por defecto se carga implícitamente en el comienzo del proceso de reenfoque

### 2.9.1. El fichero de calibraciones

NTOFOCUS emplea un fichero de texto plano con una longitud de campos fija para guardar una o más calibraciones. Este fichero se crea en el directorio de trabajo de la aplicación y se llama *VShape.txt*

La aplicación puede leer este fichero para mostrar al usuario un menú de configuraciones a cargar, pero no puede modificar este fichero directamente, sino a través de la API de NTOFOCUS.

En el fichero, cada calibración se compone de dos líneas, una para la parte izquierda de la V y otra para la parte derecha. Cada una de estas líneas comparten además una marca de tiempo con la fecha y hora en la que se produjo la calibración y una etiqueta identificadora con un nombre elegido por la aplicación.

La primera línea del fichero es una cabecera identificadora de las columnas. Las dos líneas siguientes son la calibración por defecto que se cargará en NTOFOCUS en el proceso de re-enfoque si la aplicación no ordena cargar

---

<sup>3</sup>Se cortarán automáticamente los caracteres sobrantes.

otra calibración. La etiqueta reservada para la calibración por defecto es “default”.

El propósito de la etiqueta es mostrar nombres significativos al usuario, cosa que no se lograría con una simple marca de tiempo. Sin embargo, lo que identifica de verdad a una calibración es la marca de tiempo.

### 2.9.2. Otras operaciones sobre el fichero de calibraciones

La función *AF\_Load()* se puede utilizar para cargar la calibración adecuada antes del proceso de re-enfoque. Al contrario que la función *AF\_Save()*, *AF\_Load()* necesita una marca de tiempo <sup>4</sup> para identificar la calibración. El hecho de cargar una calibración hace que ésta sea considerada la *calibración por defecto*. El proceso de re-enfoque cargará siempre la calibración por defecto, con lo que el efecto neto es el deseado.

Se puede borrar una calibración del fichero dada con *AF\_Delete()*. Si se borran todas las calibraciones, el fichero de calibraciones desaparecerá. *AF\_Delete()* también necesita como parámetro la marca de tiempo.

---

<sup>4</sup>de la forma YYYY-MM-DDTHH:MM:SS

## Capítulo 3

# Proceso de re-enfoque

El proceso de re-enfoque utiliza la información de calibración para realizar un enfoque óptimo (punto de BFP) en un momento o día posterior. Es de señalar que con este sistema, no es necesario guardar la información de la posición del enfocador entre sesiones.

En el proceso de re-enfoque se emplearán las siguientes funciones:

- void AF\_InitRefocus(void);
- void AF\_SetImage(pixel\_t\* img, int width, int height);
- void AF\_SetSkyRegion(int x1, int y1, int x2, int y2);
- void AF\_SetCentroidRegion(int x1, int y1, int x2, int y2);
- double AF\_HFD(const char\* file);
- int AF\_NextPosition(int\* position, double hfd);

### 3.1. Inicialización

Cuando la aplicación quiere iniciar el proceso de re-enfoque, llama a la función *AF\_InitRefocus()*. Como antes, se tendrá que seleccionar una zona de cielo adecuada, ya descrita en el apartado 2.4. Es preferible que la zona sea cercana a la posición actual del telescopio para evitar problemas del basculamiento del espejo en los telescopios S/C.

```
/* inicializacion en modo depuracion */  
AF_SetLogLevel(4);
```

```
AF_EnableDataSets(1);
AF_InitRefocus();
```

### 3.2. Filosofía de operación

El *modus operandi* del proceso de re-enfoque es la iteración de los siguientes pasos, hasta que se alcance el punto óptimo:

1. se toma una imagen de una estrella,
2. se mide su HFD,
3. a NTOFOCUS se le pasa pareja (*posicion, hfd*) ,
4. NTOFOCUS sugiere a la aplicación qué acción debe realizar.

En código:

```
int posicion; /* posicion actual */
int nuevaPos; /* nueva posicion propuesta */
int accion; /* accion a realizar */

<... aqui va codigo de toma de imagen ...>

AF_SetRegion(imagen, width, height);
hfd = AF_HFD(nomFichero);
nuevaPos = posicion;
accion = AF_NextPosition(&nuevaPos, hfd);

<... aqui va codigo de analisis/ejecucion de la accion ...>
```

Como puede verse, a NTOFOCUS se le pasa la posición actual y devuelve (no siempre) una posición nueva a la que ir. Además, devuelve un código de acción. La tabla 3.1 resume los códigos y las acciones a realizar:

Al cumplimentar las acciones sugeridas por NTOFOCUS, el proceso de re-enfoque pasa por los siguientes estados:

1. Alejamiento del foco de la posición actual hasta producir un “donuts”.



Código	Acción a realizar
-1	No hacer nada, se ha producido un error.
0	No hacer más movimientos, se ha alcanzado el BFP
1	No mover el enfocador y volver a medir el HFD en la misma posición.
2	Mover el enfocador a 'newPos' y medir un nuevo HFD.
3	Mover el enfocador a discrección y medir un nuevo HFD.

Cuadro 3.1: Acciones de re-enfoque

2. Comprobación de en que lado de la curva en V está la posición de enfoque tras el movimiento.
3. Acercamiento al punto de nivel de “Near Focus”.
4. En el punto de nivel “Near Focus”.
5. En el punto óptimo de foco (BFP).

A continuación se describen estos estados a título informativo. Para usar la librería sólo es necesario codificar el bucle que use las funciones anteriores y ejecutar los códigos de acción que devuelve la librería.

Por seguridad, la librería mantiene un contador interno de iteraciones, que llegado a cero, causará que cualquier llamada a `AF_NextPosition()` devuelva un código de acción -1. <sup>1</sup>

### 3.3. Alejamiento del foco de la posición actual

Tras llamar a `AF_InitRefocus()`, la aplicación hará un gran movimiento del enfocador, en la dirección en la que se no produzcan holguras. La estrella deberá convertirse un “donut”. La cantidad de pasos a mover queda a discrección de la aplicación. <sup>2</sup>

Una vez tomada la imagen y ejecutado el código:

```
AF_SetRegion(imagen, width, height);
hfd = AF_HFD(nomFichero);
nuevaPos = posicion;
accion = AF_NextPosition(&nuevaPos, hfd);
```

<sup>1</sup>En la versión actual, el número de iteraciones permitidas está fijado a 10.

<sup>2</sup>El script `VShape.R` de análisis de la curva puede ayudar a elegir un valor de salto adecuado.

se procederá a ejecutar la acción, que deberá ser un código 3 (mover a discreción) o un -1 si algo ha ido mal. Tras la llamada, 'nuevaPos' contiene la misma posición que antes de la llamada a *AF\_NextPosition()*

La aplicación nuevamente decide la cantidad de pasos a moverse. Debe ser lo suficientemente lejos como para que se vea un cambio en la imagen. Más no hace falta.

### 3.4. Comprobación del lado de la curva

Tras ejecutar la acción 3 y haber movido una cantidad suficiente de pasos, se toma de nuevo una imagen. Al medir de la estrella en esta imagen, NTOFOCUS decidirá en qué lado de la V esta la posición actual del enfocador.

Tras ejecutar de nuevo:

```
AF_SetRegion(imagen, width, height);
hfd = AF_HFD(nomFichero);
nuevaPos = posicion;
accion = AF_NextPosition(&nuevaPos, hfd);
```

procederemos a examinar la acción recibida. Puede ser un código 2, código 1 o código -1. Recordamos que si recibimos el código 2, la nueva posición del enfocador se devuelve en 'nuevaPos'. <sup>3</sup>

### 3.5. Acercamiento al punto de nivel “Near Focus”

A este estado llegaremos cuando hayamos recibido anteriormente un código 2. En este estado, estamos “resbalando” por la pendiente de la V, hasta llegar al punto de nivel Near Focus.

Sucesivas llamadas a

```
AF_SetRegion(imagen, width, height);
hfd = AF_HFD(nomFichero);
nuevaPos = posicion;
accion = AF_NextPosition(&nuevaPos, hfd);
```

van dando una código de acción 2 ó 1. <sup>4</sup>

<sup>3</sup>En la versión actual de NTOFOCUS no se devuelve código 1 en este estado.

<sup>4</sup>En la versión actual de NTOFOCUS no se devuelve código 1 en este estado.

### 3.6. En el punto de “Near Focus”

Tras resbalar”por la pendiente de la V ejecutando códigos de acción 2 habremos llegado - con un margen de tolerancia - al punto de “Near Focus”. En este punto, hay que medir el HFD muy bien <sup>5</sup> para poder saltar con confianza a la posición BFP de foco óptimo.

Por ello, la ejecución de

```
AF_SetRegion(imagen, width, height);  
hfd = AF_HFD(nomFichero);  
nuevaPos = posicion;  
accion = AF_NextPosition(&nuevaPos, hfd);
```

devuelve varios códigos de acción 1. Tras haber repetido esto varias veces, se devolverá un código 2, que nos dice que movamos al siguiente punto, el punto de enfoque óptimo. <sup>6</sup>

### 3.7. En el punto de enfoque óptimo

Después de estar varias iteraciones en el estado anterior ejecutando códigos 1, se entra en este estado tras la recepción de un código 2.

Cuando se ejecute la acción 2 y se vuelva a llamar a:

```
AF_SetRegion(imagen, width, height);  
hfd = AF_HFD(nomFichero);  
nuevaPos = posicion;  
accion = AF_NextPosition(&nuevaPos, hfd);
```

se recibirá el código de acción 0, con lo que se habrán terminado las iteraciones. Si en este estado se efectúan llamadas a NTOFOCUS en este estado, devolverán un código -1. Esto significa que hay algún bug en la aplicación que le está haciendo iterar de más.

---

<sup>5</sup>En la actualidad, se toman 5 muestras del HFD en esta posición para averiguar el promedio (mediana).

<sup>6</sup>En la versión actual de NTOFOCUS, nos daremos cuenta que hemos llegado a este estado porque recibimos códigos de acción 1. En versiones futuras, en los estados anteriores también podrían llegar códigos 1, por lo que será imposible (e irrelevante) cuándo se ha llegado a este punto.



# Capítulo 4

## Auditoría

### 4.1. Registro de actividad

La ejecución de los dos procesos, calibración y re-enfoque, es bastante laboriosa y muchas cosas pueden ir mal. Como ayuda al diagnóstico de problemas, NTOFOCUS incorpora un fichero de registro de actividad de sus acciones llamado *ntofocus.txt*. Este registro de actividad tiene 5 niveles de información. Cada nivel incluye información de todos los niveles inferiores. La tabla 4.1 muestra dichos niveles.

Nivel	Etiqueta en el log	Uso típico
5	VERBO	Trazas dentro de bucles. No utilizar.
4	DEBUG	Nivel de depuración. Abundantes trazas informativas.
3	INFO	Nivel por defecto. Muestra pasos significativos del proceso.
2	AVISO	Mensajes que requieren cierta atención.
1	ERROR	Errores que detiene el flujo natural del proceso.
0	-	No saca mensajes.

Cuadro 4.1: Niveles de trazado

El nivel por defecto de NTOFOCUS es 3 (INFO) y se puede cambiar momento llamando a la función *AF\_SetLogLevel()*, preferiblemente justo antes del principio de cada proceso de calibración o re-enfoque. Este fichero se machaca con nuevo contenido por cada comienzo de calibración o re-enfoque.

## 4.2. Datasets

La segunda gran herramienta de que dispone NTOFOCUS es la generación de “datasets”, ficheros con información de texto en formato adecuado para importar en alguna aplicación de análisis.

La ejecución de NTOFOCUS produce los siguientes “datasets”:

**R/MarginalX.txt** - Fichero con la distribución marginal X de una imagen durante el proceso de calibración. Se utiliza para el análisis de las distribuciones marginales. Se machaca al comienzo de cada proceso de calibración o re-enfoque.

**R/MarginalY.txt** - Idem.

**R/FeaturesX.txt** - Fichero con información del tamaño de la estrella a partir de las distribuciones marginales. Se machaca al comienzo de cada proceso de calibración o re-enfoque.

**R/FeaturesY.txt** - Idem.

**R/Centroid.txt** - Fichero con la información de centroide y tamaños finalmente seleccionados. Se machaca al comienzo de cada proceso de calibración o re-enfoque.

**R/Sky.txt** - Fichero con información de valor y desviación del fondo de cielo. Se machaca al comienzo de cada proceso de calibración o re-enfoque.

**R/VShapePoints.txt** - Fichero de texto con todas las medidas realizadas en todas las calibraciones hechas. Se utiliza para analizar calibraciones en R. No se machaca, siempre se añade.

**R/Refocus.txt** - Ficheros con los puntos (*posicion, hfd*) producidos en el re-enfoque. Se utiliza para analizar la secuencia del último re-enfoque. Se machaca con cada nuevo re-enfoque.

**R/imagen.txt** - Volcado en ASCII en filas y columnas de la imagen “imagen.fit”

Para habilitar la generación de “datasets”, se ha de utilizar la función *AF\_EnableDataSets()*. Al invocarla con un parámetro de 1, se habilita la exportación de dichos datos. Con un valor de 0, se deshabilita. Por defecto, no se habilita la exportación.

Como en el caso del registro de actividad, es recomendable llamarla justo antes del de calibración o re-enfoque.

## Capítulo 5

# Análisis

Para ayudar al desarrollo de la librería, se ha recurrido al paquete de software libre de análisis estadístico 'R'.

R Development Core Team (2005). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org>.

Hay varios scripts R de utilidad:

**Aperture.R** - dibuja las distribuciones marginales, centroide y diámetro recomendado de medida para la fotometría posterior. Opcionalmente, muestra las aperturas sobre la imagen.

**VShape.R** - analiza y pinta la gráfica en V.

**Refocus.R** - analiza y pinta el proceso de re-enfoque.

Las secciones siguientes muestran cómo interpretar las gráficas producidas por estos scripts. Antes de ejecutar dichos scripts, es necesario establecer el directorio de trabajo de R en el mismo sitio donde se ha dejado la librería *ntofocus.dll*.

### 5.1. Análisis de la secuencia de calibración

Antes de pasar al análisis particular de imágenes, conviene echar un vistazo a los resultados de la secuencia de calibración, utilizando el script *VShape.R*.

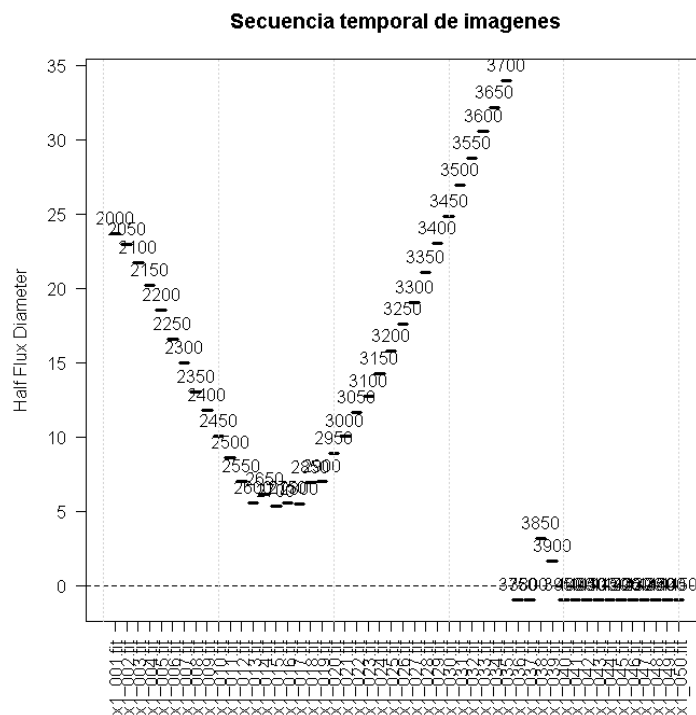


Figura 5.1: Secuencia temporal de calibración. Notar cómo a partir de la imagen 036 se han producido errores.



La figura 5.1 muestra en el eje X la secuencia temporal de imágenes y en el eje Y el HFD medido. En cada barra aparece la posición del enfocador. Del examen de la figura, se deduce que algo raro pasa a partir de la imagen *foco9\_036.fit*. Esta imagen y la siguiente muestra un nivel de HFD anormalmente bajo. El resto, con un HFD de -1.0, han sido identificadas por NTOFOCUS como mediciones erróneas.

El menú de R nos invita a ver la curva de calibración obtenida por NTOFOCUS.

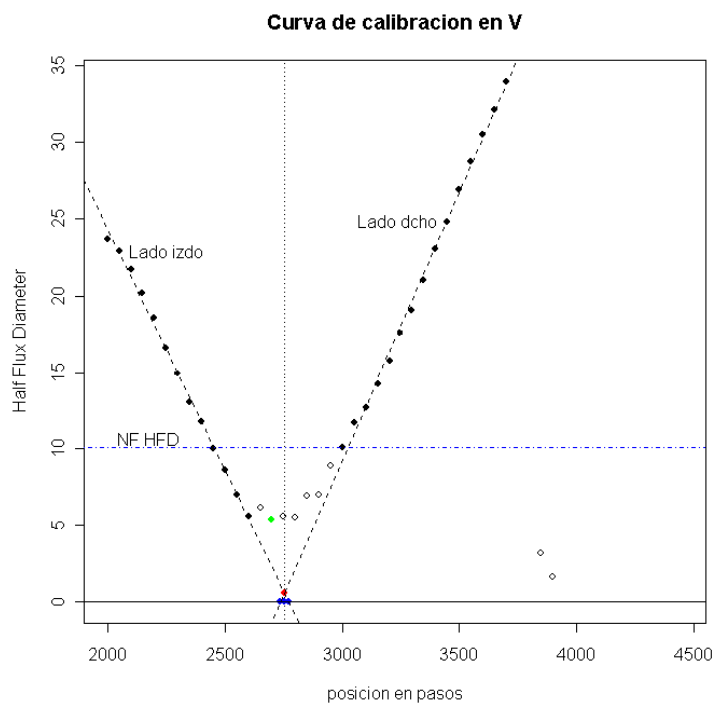


Figura 5.2: Calibración producida. En negro, los puntos considerados. En hueco, los puntos descartados.

En la figura 5.2 se pueden distinguir varios elementos:

**En color negro,** los puntos utilizados para el ajuste por mínimos cuadrados.

**En color verde,** el codo que separa el lado izquierdo del derecho.

**En forma hueca,** los puntos extremos que han sido rechazados por discordantes (outliers) y los puntos centrales que han sido rechazados porque contribuían a exceder la tolerancia del ajuste lineal.

**En color rojo,** el punto de mejor foco en este momento <sup>1</sup>.

**En color azul,** las intersecciones de las rectas de regresión con  $y = 0$  y una aproximación a la abscisa del punto de mejor foco.

**En color azul,** la línea del nivel “Near Focus”, que pasa en este caso por el punto de la derecha, porque ha sido calculada *a posteriori*.

## 5.2. Análisis de imágenes

Estas gráficas e imágenes se obtienen al ejecutar el script *Aperture.R*. La figura 5.3 muestra el aspecto de las distribuciones marginales y aperturas para un donuts medido correctamente. Lo deseable es que las distribuciones marginales sean simétricas respecto al centroide y que no haya estrellas o brillos cercanos que dificulten el delimitar los límites de la estrella.

En la figura 5.3 podemos distinguir en las distribuciones:

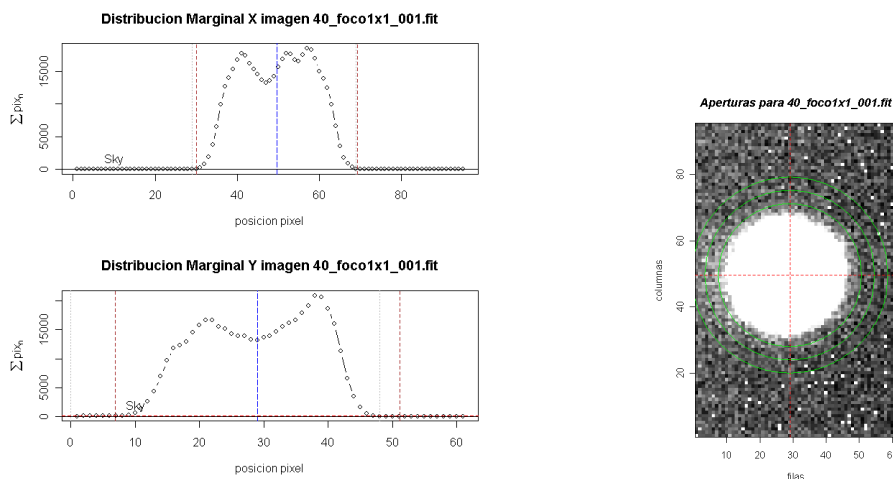


Figura 5.3: Distribuciones marginales que muestran el centroide (azul), los radios para fotometría (marrón) y el límite hallado para la estrella (gris).

**En color azul,** el centroide calculado.

<sup>1</sup>este punto es, en realidad, otra aproximación a la verdadera BFP

**En color rojo**, el umbral  $sky + 3\sigma_{sky}$  donde acaba el fondo de cielo y comienza la estrella.

**En color gris claro**, y apenas perceptibles, dos líneas verticales que delimitan los límites de la estrella.

**En color marron**, dos líneas simétricas con el centroide que marcan el diámetro recomendado en ese eje (X o Y) para la fotometría de apertura.

Y en la imagen:

**En color rojo**, el centroide y la apertura para la fotometría.

**En color verde**, el anillo para calcular el fondo de cielo local, según el procedimiento clásico de la fotometría de apertura.

Sin embargo, por muchos factores, la cosa puede no salir bien en ocasiones. A continuación se muestran algunos casos problemáticos. NTOFOCUS no es capaz de tratar bien muchos de estos casos y calcula incorrectamente el centroide o la apertura recomendada para la fotometría.

### 5.2.1. Donut demasiado grande

Procedemos a examinar la imagen *40\_foco\_1x1\_036.fit*, cuyas distribuciones marginales  $M_x$  y  $M_y$  podemos observar en la figura 5.4. Las distribuciones de esta imagen parecen recortadas. Y el cálculo del centroide y los anillos de fotometría total mente erróneos.

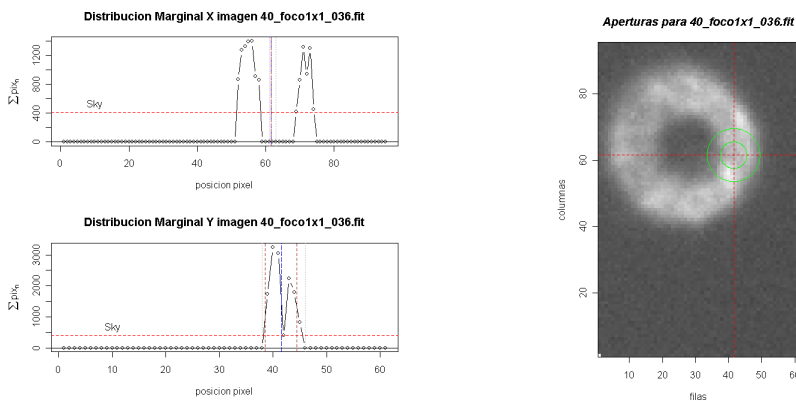


Figura 5.4: Distribuciones marginales fuertemente recortadas por un umbral de cielo alto

La respuesta la tiene la figura 5.5

Esta gráfica muestra cómo el algoritmo de estimación de fondo de cielo ha estado dando valores consistentes de promedio y desviación típica durante toda la serie de calibración hasta que se ha “roto”. El fondo de cielo y sobre todo la desviación típica se disparan. Como consecuencia, el umbral que fija tanto el cálculo del centroide como los límites del “donuts” es demasiado alto y el resultado es completamente erróneo.

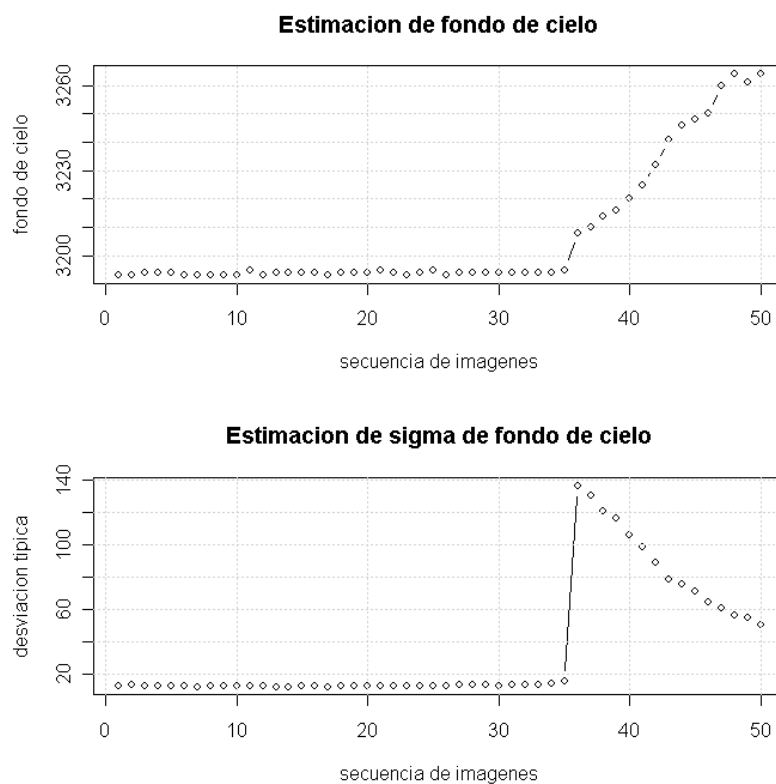


Figura 5.5: Evolución del fondo de cielo en la serie de imágenes.

La causa de semejante rotura la tiene la proporción de píxeles de “donut” respecto a píxeles de fondo de cielo. Ambas poblaciones resultan comparables y el histograma es casi bimodal.

Es una suerte que el algoritmo se “rompa” de un modo brusco porque se generan mediciones discordantes que se detectan en la etapa de ajuste. Sin embargo, la mejor opción es aumentar el tamaño de la imagen.

### 5.2.2. Sólo fondo de cielo

La figura 5.6 muestra el aspecto de las distribuciones y de la imagen cuando uno se descuida y no atina a introducir una estrella en el campo. Afortunadamente, NTOFOCUS sí que detecta este caso.

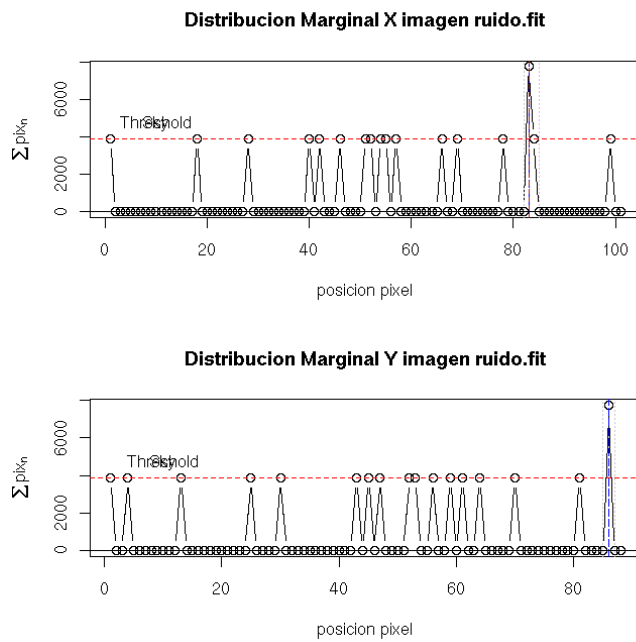


Figura 5.6: Distribuciones de fondo de cielo

### 5.2.3. Reguerillos de luz

La figura 5.7 muestran el aspecto de las distribuciones marginales de una estrella casi enfocada y su imagen correspondiente muestra un reguerillo doble de luz. Estos reguerillos se forman al no utilizar obturador en estrellas brillantes.

Notar cómo la apertura fotométrica se ve algo aumentada por la presencia del reguerillo de luz, aumentando el flujo medido y decreciendo la exactitud de la medida HFD en esta situación.

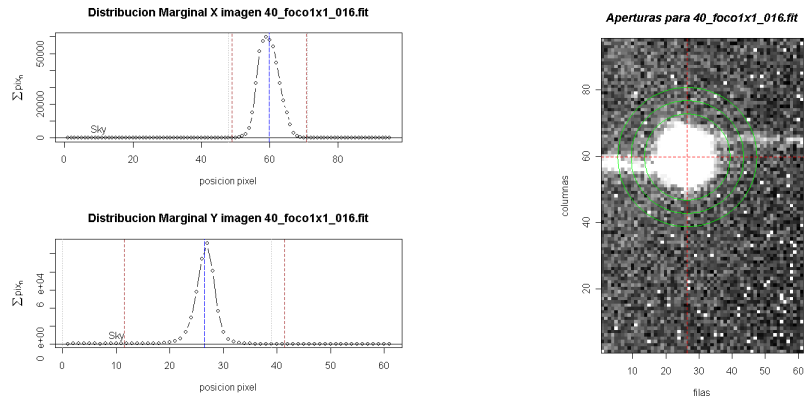


Figura 5.7: Reguerillos de luz

La selección del radio final para la apertura es una media ponderada de los distintos radios mostrados en las distribuciones marginales, con tendencia a favorecer al radio más corto. Aún así, el reguerillo tiene su influencia en el cálculo.

## Capítulo 6

# Algoritmos

### 6.1. El algoritmo HFD

Se define el Half Flux Diameter o HFD como el diámetro de una apertura fotométrica aplicada a la imagen de una fuente de luz estelar tal que la mitad del flujo medido quede dentro de la apertura y la otra mitad quede fuera.

Esta métrica es independiente de la forma que adopte esa fuente de luz bajo condiciones de desenfoque. Esto es una gran ventaja porque no se necesitan formular hipótesis adicionales, salvo el que sea razonablemente simétrica de revolución. Por ejemplo, en los telescopios S/C la PSF se va transformando de un perfil más o menos gaussiano a uno en forma de “donut”.

Otra gran ventaja de esta métrica es que, salvo en la parte central de la gráfica, la relación entre el HFD y la posición es muy lineal, lo que facilita la realización de un algoritmo por ordenador.

Por último, la forma de esta  $V$  (pendientes y puntos de corte con los ejes cartesianos) es independiente de la temperatura. Una vez hecha una calibración se puede reutilizar mientras no cambie la configuración óptica (relación focal).

La obtención de esta métrica HFD plantea retos interesantes como el de saber cómo el del tamaño de una estrella para poder medir su flujo total, dónde está su centro para poder centrar una apertura circular, cómo medir el brillo del fondo del cielo y su dispersión, y por supuesto, la realización de la fotometría de apertura.

Los pasos a seguir para averiguar el HFD de una estrella son. a grandes rasgos:

- Determinar el fondo de cielo y su desviación estándar.

- Determinar el centroide de la estrella.
- Determinar la apertura circular que engloba a la estrella.
- Medir el flujo total en esa apertura.
- Determinar el diametro de otra apertura que hace que el flujo medido sea la mitad que el anterior.

## 6.2. Otros algoritmos

Una breve descripción sobre los algoritmos empleados.

**Fondo de cielo.** Se utiliza un algoritmo de *sigma-clipping*, tal y como se describe en [3], con un umbral de  $3\sigma$ . Es bastante robusto y discrimina bien las estrellas o donuts hasta que se vuelven demasiado grandes.

**Centroide.** Se calculan las distribuciones marginales  $M_x$  y  $M_y$  y se calcula el centroide de cada una de ellas, según el método de los momentos modificados expuesto en [1]. El umbral se fija a la mitad del máximo de la distribución para evitar las influencias de los “reguerillos de luz” propios de no usar el obturador.

**Tamaño de la estrella.** A partir del centroide de una distribución marginal, se miran los dos radio que hacen que el nivel acumulado sea mayor que *cielo* +  $3\sigma$ . En una distribución asimétrica se hace una media ponderada de los dos radios. Como radio final común, se vuelve a realizar otra media ponderada de los radios de  $M_x$  y  $M_y$ .

**Medición del flujo.** Por fotometría clásica de apertura circular con el radio recomendado en el paso anterior. Se coloca un anillo exterior de 4 píxeles de ancho y a una distancia de 4 píxeles de la abertura. Se efectúa corrección de píxeles parciales.

**Cálculo del HFD.** Por un simple algoritmo de bisección descrito en [2].

**Mínimos cuadrados** - Según las fórmulas descritas en [2]



# Bibliografía

- [1] Ronald C. Stone: *A Comparison of Digital Centering Algorithms*. The Astronomical Journal, Volume 97, Number 4.
- [2] William H. Press et al. *Numerical Recipes in C*. Second Edition. Cambridge University Press.
- [3] Philip Massey. *A User's Guide to CCD Reductions with IRAF* 15 Feb 1997.