

Técnicas de integración de imágenes

Aumento de la relación S/R y rechazo de *outliers* con PixInsight[©]

VI Seminario de Astrofotografía Digital de Cielo Profundo, Madrid 20 de Noviembre 2010

Texto aproximado de la presentación oral de la ponencia

1. Un poco de teoría

- Consideraciones previas
- Media
- Mediana
- “*Sigma Clipped*”

2. La herramienta *ImageIntegration* de PixInsight

- Descripción de la interface
 - Integración de imágenes
 - Métodos de combinación
 - Normalización de imágenes
 - Ponderación de imágenes
- Algoritmos de rechazo
 - Tipos de algoritmos
 - Mini guía de aplicación
 - Principios de funcionamiento
- Información obtenida tras la aplicación
 - Consola de proceso
 - Evaluación del ruido
 - Mapas de rechazo y *slope map*
- Estrategia de aplicación

3. Ejemplos de integración de imágenes

- Integración de un número *medio* de imágenes (la situación habitual.....)
- Integración de un número *elevado* de imágenes (en busca de la excelencia.....)
- Integración de un número *reducido* de imágenes (casos desesperados.....)

Lista de referencias

Texto de la ponencia

Diapositiva 2:

La astrofotografía es una especialidad de fotografía de la naturaleza que caracteriza desde el punto de vista de la fotografía convencional, porque todas sus imágenes tal y como son adquiridas están profunda y sistemáticamente subexpuestas.

En estas condiciones de partida, está claro que la relación señal ruido de la imagen pasa a tener una importancia capital. El objeto de esta presentación es mostrar cómo, empleando herramientas y técnicas adecuadas en el proceso de integración de imágenes, con un poco de esfuerzo, se pueden conseguir aumentos significativos en la relación Señal/Ruido (que en adelante indicaremos por las siglas “**SNR**”) en la imagen integrada, y eso de forma compatible con un nivel adecuado de rechazo de lo que denominaremos *outliers*

He dividido la presentación en 3 partes, una introducción con un poco (poco) de teoría, una descripción de la herramienta *ImageIntegration* de PixInsight, que es la que conozco mejor y utilizo habitualmente y por último unos ejemplos prácticos de integración de imágenes

Diapositiva 4

Naturalmente antes de llegar a la etapa de integración de imágenes, debemos habernos esforzado en emplear todo el arsenal de técnicas disponibles para aumentar la señal y disminuir el ruido durante la etapa de adquisición de datos.

Especialmente importante para el empleo de técnicas estadísticas avanzadas es el empleo de *dithering*

También son importantes el proceso de calibrado y de alineado de imágenes. Para buenas prácticas en todos estos procesos una buena recomendación es revisar el trabajo de Antonio Fernández sobre SNR y el de Ignacio de la Cueva sobre calibración del seminario del año pasado.

Diapositiva 5

Bien, volvamos a la escuela, el algoritmo de la media es bien sencillo: Suma de valores de pixel dividido por el número de imágenes. Pues con esa sencillez, la media nos proporciona el mayor incremento de relación señal ruido posible.

Veamos un ejemplo con 6 imágenes. Aquí tenemos las 6 imágenes individuales y aquí la imagen integrada con la media. Seleccionamos ahora un mismo punto de fondo de cielo en cada imagen y medimos su valor. Se aprecia claramente el carácter aleatorio de la señal de fondo.

Por último medimos (más adelante explicaremos cómo) el incremento de relación Señal/ruido de la imagen integrada. Es importante notar que debido a que el ruido no es constante a lo largo de la totalidad del conjunto de datos, es imposible alcanzar con imágenes reales el valor teórico indicado arriba.

Diapositiva 6

Desgraciadamente la vida del astrofotógrafo/a es dura y el enemigo acecha. Acecha en forma de lo que a falta de mejor traducción he denominado “*outliers*”:

Acecha en forma de píxeles calientes y fríos (aquí podemos ver unos cuantos calentitos)

Acecha en forma de rayos cósmicos (aquí podemos ver uno en un dark)

En forma de trazas de satélites y aviones, y aunque se me haya acabado el espacio, también en forma de columnas defectuosas, asteroides, etc

Diapositiva 7

Veamos cómo se comporta la Media frente al ataque de un “enemigo”: Sustituimos una de las 6 imágenes por otra que contiene la traza de un avión

Aquí vemos como sube el valor de pixel en el punto de paso de la traza y claro, aquí tenemos el valor que nos proporciona la media. Acabamos pues de descubrir (mejor debería decir recordar) la principal desventaja de la media: se le atragantan los “*outliers*”

Diapositiva 8

Como la media no nos soluciona nuestros problemas, sigamos nuestro camino por la olvidada estadística y recordemos la mediana: el algoritmo es también de lo más sencillo y lo tenemos aquí: ordenar los valores de pixel y escoger el que está en el medio

La ventaja es también aquí categórica: ofrece el mejor rechazo de *outliers*.

Veamos de nuevo nuestras 6 imágenes: El resultado es muy bueno! Ha desaparecido totalmente la traza de avión!

¿Podemos dejar ya de sumar imágenes e irnos a restar con Ignacio?: Todavía no.

Si medimos el incremento de relación S/R conseguido veremos que es mucho menor que el que habíamos obtenido con la media, concretamente un 16,4% menor.

Diapositiva 9

Entrando en aplicaciones con base estadística más profunda, las que aquí he agrupado bajo el nombre Sigma Clipped, se pueden conseguir mejores resultados.

En cuanto al algoritmo, aunque como aquí digo hay unas cuantas variantes, lo habitual es calcular la media y la desviación standard de la distribución de pixels y a continuación únicamente promediar los pixels que se encuentren dentro de un rango alrededor de la media definido por un factor sigma, excluyendo los que están fuera de este rango.

Aquí vemos como se comporta uno de estos algoritmos con nuestras 6 imágenes: el resultado es muy bueno, no se aprecia el dichoso avión y además, el valor de incremento de relación S/R es muy superior al conseguido con la mediana y se queda tan solo a un 6% del valor de la media

Diapositiva 10

Las principales desventajas son que necesita bastantes imágenes para dar lo mejor de sí (el ejemplo anterior es casi un caso límite) y sobre todo que como en toda herramienta que elimina información, debe aplicarse con cuidado pues puede producir rechazo de pixels correctos

En cualquier caso, la conclusión es hacer servir tanto como sea posible estos algoritmos pues son los que presentan la mejor combinación de aumento de relación S/R y rechazo de outliers

Sin embargo incluyo aquí una recomendación: empleando aplicaciones fiables y/o que tengan documentados los algoritmos que emplean. Al preparar este trabajo he encontrado:

Alguna aplicación con una implementación deficiente que proporciona un resultado mediocre en rechazo de *outliers*

Alguna aplicación "milagro" que proporciona una imagen integrada *Sigma Clipped* con un nivel de ruido inferior a la misma imagen integrada únicamente con *Average* (lo cual es, matemáticamente imposible)

Diapositiva 11

Comienza la segunda parte de la presentación donde veremos la herramienta *ImageIntegration*, describiendo su interface, sus algoritmos de exclusión, la información que se obtiene tras la integración, con especial mención aquí al sistema de medición de ruido que incorpora, y por último una propuesta de metodología de aplicación

Diapositiva 12

Esta herramienta realiza la integración de un número ilimitado de imágenes prácticamente de cualquier tamaño en formato FITS. (no las carga en RAM.....)

Sus principales características son:

- o Posibilidades de combinación de imagen con operadores media, mediana, máximo y mínimo.
- o Múltiples opciones de algoritmos de rechazo de *outliers*, para poder adaptarse a cualquier situación.
- o Normalización y escalado automático de imágenes, definible por separado tanto para las operaciones de combinación, como para las de rechazo de pixel
- o Exhaustiva información adicional proporcionada al final del proceso, como veremos en su momento

Diapositiva 13

El interface de la aplicación puede imponer un poco la primera vez que nos enfrentamos a él, o sea que lo vamos a ir viendo por secciones.

La primera es la sección en la introducimos las imágenes a integrar y no plantea mayores problemas

13-1 A continuación tenemos la sección que realiza la combinación de imágenes. Permite hacer una normalización del fondo de las imágenes y también asignar una ponderación a cada una de ellas.

13-2 La siguiente sección es la que nos permite seleccionar el algoritmo de rechazo de pixel que vamos a emplear

13-3 Y por último, la última sección es la que nos permite ajustar los límites de rechazo para optimizar la imagen integrada. Ahora que ya hemos visto la herramienta en su conjunto vamos a meternos en cada una de las secciones

Diapositiva 14

En la sección de integración de imágenes es necesario especificar los siguientes parámetros: combinación, normalización y ponderación, que vamos a ver por separado a continuación. En la mayor parte de los casos no es necesario tocar nada más (es especialmente importante que esté marcada la casilla evaluación de ruido)

Diapositiva 15

La combinación de imágenes permite seleccionar entre media, mediana, mínimo y máximo. De lo que hemos visto en la primera parte se deduce que la mejor opción para imágenes astronómicas, salvo casos especiales, como trazos de estrellas o cometas, es emplear la media, *average*, y complementar con un buen algoritmo de rechazo de pixels.

Diapositiva 16

Dado que vamos a promediar valores de pixel, la normalización de imágenes es una necesidad esencial: podemos por ejemplo tener imágenes tomadas en diferentes noches con niveles de fondo distintos. Si las promediáramos sin normalizar, es decir sin igualar los fondos de cielo, obtendríamos un resultado erróneo. Dado que hemos dicho que solo promediaremos imágenes correctamente calibradas, la normalización aditiva será la que deberemos aplicar

Diapositiva 17

Las imágenes se pueden ponderar en función de varios parámetros: en general la mejor opción, es la ponderación en función de la medición del ruido de las imágenes, ya que acostumbra a dar una mejor relación S/R al final.

Diapositiva 18

Una vez definidos los parámetros de la sección de combinación de imágenes, pasamos a la sección de rechazo de pixels. En ésta debemos definir el algoritmo de rechazo y la normalización a emplear.

Diapositiva 19

Esta es una de las selecciones más importante pues el resultado final depende mucho de ella. Por lo que vamos a dedicarle un cierto tiempo. El programa presenta muchas opciones (y el diseñador del programa me ha comentado que está trabajando todavía en otra más, que a priori parece muy interesante). En general lo que se puede decir es esto: que

cada algoritmo está indicado especialmente para un tipo de problema a resolver, pero también que hay solapes en el uso óptimo de algoritmos, por lo que si queremos obtener el mejor resultado habrá que experimentar. De cualquier modo aquí tenemos una guía

Diapositiva 20

Esta tabla está confeccionada en base a la información del programa y a mi propia experiencia.

Para emplear algoritmos *sigma clipped*, los más potentes, es condición *sine qua non* disponer de una buena estimación de la desviación standard de la distribución de valores de pixel y para eso es necesario tener un número mínimo de imágenes.

Si disponemos de un número muy reducido de imágenes digamos entre 3 y 5 ó 6, no es posible obtener una buena estimación y por tanto la mejor opción en ese caso es *Percentile Clipping*, que es un algoritmo de una sola pasada en él que se excluyen los pixels que quedan fuera de un rango hábilmente fijado sobre la mediana (y no sobre la media).

Si disponemos de algunas imágenes más, digamos entre 5 y 10 la mejor opción es *Averaged Sigma Clipping*. Aquí estamos ya con algoritmos iterativos, mucho más potentes.

Si superamos las 10 imágenes *Winsorized Sigma Clipped* acostumbra a dar un resultado muy bueno, superior al *Sigma Clipping* clásico, y por último si tenemos muchas imágenes, *Linear Fit Clipping* puede dar todavía mejores resultados.

Vamos a ver a continuación una explicación del funcionamiento de los dos algoritmos de rechazo más potentes

Diapositiva 21

Para ello haré servir un ejemplo gráfico sacado, de forma autorizada, de la información del programa:

Supongamos que tenemos N imágenes para integrar. En la figura hemos representado un stack de N pixels situados en las mismas coordenadas de cada imagen, con sus valores ordenados en orden ascendente. El eje horizontal es el número de orden de la muestra y el eje vertical el rango de valores de pixel.

En la figura el símbolo m significa la mediana de la distribución y σ su desviación tipo.

En los algoritmos clásicos k -Sigma Clipping, m y σ se calculan de la forma habitual. En la figura hay dibujadas dos líneas horizontales que representan los *clipping points*, los puntos de corte o de rechazo: Los valores de pixel que exceden $kH*\sigma$ o $kL*\sigma$ son descartados

En el algoritmo *Winsorized Sigma Clipping*, para el cálculo de m y σ , los valores extremos son sustituidos por sus *nearest neighbors*. Este proceso aplicado de manera iterativa, permite una estimación más *robusta* de m y σ (el término robusto debe entenderse aquí como menos afectado por los *outliers*). Sin embargo una vez determinados m y σ el proceso de exclusión de pixels es el mismo.

En la figura los pixels rechazados están representados como círculos blancos: hay 6 de ellos, 3 en cada extremo de la distribución. La pregunta que deberíamos hacernos es *¿son realmente outliers?*

La respuesta es: *muy probablemente sí, en condiciones normales*. Supongamos que nuestras imágenes tengan un pequeño gradiente y que no todas lo tengan en el mismo sentido. Esto puede suceder por ejemplo si hemos hecho un paso de meridiano en la sesión de adquisición o si hemos hecho la captura de imágenes en diferentes noches.

En estos casos, la distribución de valores de pixel puede mostrar colas o extremos de *outliers aparentes* en ambos extremos, y eso puede llevarnos a que rechacemos pixels válidos.

El algoritmo *Linear Fit Clipping* intenta reducir esta posibilidad

Este nuevo algoritmo ajusta una línea recta al conjunto de datos (es decir intenta encontrar el mejor ajuste en el sentido de minimizar la desviación media). Esta línea da idea de la *tendencia* de los valores de pixel. Este algoritmo es más tolerante a imágenes con gradientes: si la imagen no los tiene (y ha sido correctamente calibrada con flats y normalizada en el paso anterior) el ajuste de la recta tenderá a la horizontal, pero si hay variaciones de iluminación, la pendiente de la recta tenderá a aumentar.

En definitiva lo que importa es que el resultado final, cuando disponemos de número de imágenes importante, es un menor rechazo de *falsos outliers*.

Diapositiva 22

Una vez seleccionado el algoritmo y ejecutado, la herramienta nos ofrece mucha información para ayudarnos a evaluar el proceso y el resultado. Comencemos por la información que nos proporciona la consola de proceso:

Para cada imagen individual nos da su factor de escalado, nivel de ruido y factor de ponderación

Aquí tenemos información sobre modo de integración: combinación, normalización y algoritmo de rechazo

Y aquí los datos sobre los pixel rechazados en cada imagen: con número total pixels y % sobre el total (con detalle de los pixels rechazados en cada extremo de la distribución)

Por último tenemos la **Evaluación de ruido de la imagen integrada**. Esta información obtenida de forma automática mediante técnicas de wavelets es fundamental para evaluar y sobre todo afinar el resultado final. La justificación matemática del cálculo se encuentra en la lista de referencias. La información facilitada es

- Estimación de nivel de ruido de la imagen integrada
- Incremento de SNR de la imagen integrada con respecto a la imagen de referencia (la primera del stack)
- **Incremento medio de SNR de la imagen integrada (sobre el conjunto de las imágenes)**

Como el nivel absoluto de ruido es una cifra abstracta que tiene poco sentido para nosotros, haremos servir el incremento SNR medio, que tiene la ventaja de poder compararse con el valor teórico que esperamos obtener (por ejemplo, si integramos 16 imágenes sabemos muy fácilmente que el incremento teórico de SNR es 4)

Diapositiva 23

Al concluir el proceso de integración la herramienta proporciona, junto con la imagen integrada dos imágenes más: los mapas de rechazo. Los algoritmos *sigma clipped* proporcionan un mapa para los píxeles rechazados en cada extremo de la distribución. Estos mapas tienen varias aplicaciones, siendo la más obvia el comprobar que el rechazo de trazas de avión u otros *outliers* se ha realizado adecuadamente.

El algoritmo *Linear Fit Clipping* proporciona un tercer mapa denominado *Slope map*, literalmente mapa de pendiente, que puede ayudar a visualizar mejor por ejemplo la presencia de gradientes en las imágenes a integrar.

Diapositiva 24

Para acabar el bloque dedicada al conocimiento de la herramienta, aquí os presento un posible proceso de trabajo para aplicar la herramienta *ImageIntegration* a un conjunto de imágenes de cielo profundo:

En primer lugar, como es lógico, cargamos las imágenes

A continuación ajustamos los parámetros de la sección *ImageIntegration* tal y como hemos visto antes

Ejecutamos la aplicación. Obtendremos una imagen sin rechazo de *outliers*, pero con el mejor incremento posible de relación señal/ruido. Guardamos la imagen y anotamos ese valor

Una vez hecho esto seleccionamos el algoritmo de rechazo según número de imágenes, la tabla anterior y la experiencia personal.

Probamos varios algoritmos, analizamos los resultados y seleccionamos el que nos ofrezca mejor resultado en rechazo de *outliers*

Por último, con el algoritmo seleccionado refinamos los valores de corte *low* y *high* de la sección *Pixel Rejection (2)* para conseguir el rechazo de *outliers* deseado con el mejor incremento de relación señal/ruido posible

Diapositiva 25

Comenzamos el **último bloque**, el más práctico. Vamos a ver tres ejemplos, el primero con un número razonable de imágenes a integrar, que debería ser la situación habitual, en el segundo veremos que sucede cuando tenemos muchas imágenes y por último veremos que sucede cuando todo ha fallado y tenemos muy pocas imágenes para integrar

Diapositiva 26

Para los dos primeros ejemplos haremos servir esta imagen. El objeto central es VdB152, el equipo es un TOA-150 con una STL-11000 con exposiciones de 15 minutos. Por lo que respecta al astrofotógrafo, sobre todo en palabras de su sufrida esposa se le podría calificar como de un poco obsesivo ☺☺☺

Diapositiva 27

En el primer ejemplo nuestro astrofotógrafo tomó 15 imágenes. Este es un número que permite hacer una buena estimación de la desviación tipo de la distribución de valores de píxeles y por tanto serán plenamente aplicables los algoritmos de exclusión más potentes.

Siguiendo el plan que nos hemos marcado, en primer lugar realizamos la integración con la media, sin ningún algoritmo de rechazo, y obtenemos este resultado.

Para analizar la imagen y compararla con las que ahora obtendremos, le aplicamos un script que se hace servir para destacar las aberraciones del sistema óptico y que nos proporciona 4 recortes de las esquinas de la imagen y uno del centro. Claramente se aprecia que en la esquina superior derecha tenemos una traza de avión

Diapositiva 28

A continuación vamos a repetir la integración con varios algoritmos. Naturalmente, cuando se conoce la herramienta se puede ir mucho más directamente al algoritmo óptimo. En este caso, con este número de imágenes yo habría ido a probar únicamente dos algoritmos, pero para hacer más didáctica la presentación he incluido también el *Sigma Clipping* clásico que es el que tenemos aquí, con la imagen integrada, los mapas de rechazo y los resultados en la consola de proceso.

Hay que hacer notar que estas integraciones sucesivas no consumen excesivo tiempo, pues los datos de la medición de ruido de la imágenes individuales, que es el proceso que consume más tiempo, se guardan en un caché temporal.

Diapositiva 29

Aquí tenemos los resultados obtenidos con *Winsorized Sigma Clipping*

Diapositiva 30

Y aquí tenemos los resultados de *Linear Fit Clipping*. Recordar que todos estos resultados los hemos obtenido con los parámetros por defecto que nos ofrece la herramienta.

Ahora que ya hemos recogido bastantes datos, vamos a analizarlos y compararlos

Diapositiva 31

¿Cómo lo vamos a hacer?

Pues vamos a comparar el incremento de la relación señal / ruido que hemos obtenido con cada algoritmo, vamos a comprobar la situación en los bordes de la imagen integrada, y por supuesto, vamos a comparar la efectividad en el rechazo de *outliers*

Diapositiva 32

En esta tabla he resumido los datos más significativos que nos proporciona la consola de proceso.

Como hay muchos datos, vamos a concentrarnos en la columna enmarcada en rojo: el incremento medio de relación señal/ruido que hemos obtenido en cada integración.

En primer lugar he incluido el valor obtenido empleando la media: 3.434. Este es el valor máximo que se puede alcanzar con este juego de imágenes, es nuestro techo, pero es un techo que, recordemos, nosotros no podremos alcanzar porque queremos excluir los *outliers* y cualquier algoritmo de exclusión (no milagroso) al descartar pixels, incrementa indefectiblemente el nivel de ruido en la imagen integrada. Pero de cualquier forma, es nuestra referencia y el valor al que debemos intentar aproximarnos tanto como podamos.

A continuación he puesto el valor obtenido con la Mediana: observar que diferencia más notable, es un 20% inferior.

Queda claro que la mediana no es una opción a considerar si disponemos de herramientas más sofisticadas.

Esto nos deja los tres algoritmos *Clipped*. El que tiene mejor relación S/R es el *Linear Fit* y el que tiene peor el clásico *Sigma Clipping*. Observemos además que el *Sigma Clipping* no ha descartado ningún pixel por la banda baja. Esto no es un problema en si mismo, simplemente quiere decir que en esta imagen, los límites por defecto de este algoritmo no son lo más adecuados. Pero lo que si nos dice este valor es que vamos a tener que ampliar el límite de rechazo inferior, con lo que el valor 3.125 empeorará más.

Diapositiva 33

Continuemos nuestro análisis. ¿Por qué digo de ver la situación en los extremos de la imagen?.

Pues porque diferentes algoritmos tienen diferentes comportamientos en los extremos, en los que el uso de *dithering* hace que se dispongan de menos imágenes para integrar. Con sensores grandes como el que yo utilizo, esto no acostumbra a ser problema, pero con sensores pequeños, si el encuadre está algo justo puede serlo.

Diapositiva 34

Esto se puede apreciar mejor si aplicamos a las imágenes el script que hemos visto antes y que nos presenta cuatro recortes de las esquinas y uno del centro. Tenemos aquí las cuatro imágenes, la de la media arriba a la izquierda, la integración *Sigma Clipping* abajo a la izquierda, *Linear Fit* arriba a la derecha y *Winsorized* abajo a la derecha.

Desde este peculiar punto de vista se aprecia que *Winsorized* se comporta mejor.

Pero esto no deja de ser una cuestión bastante accesoria, lo que nos interesa más, es el rechazo de *outliers*. Para ello vamos a hacer un zoom en la zona marcada con un círculo azul, en la que como puede apreciarse tenemos una magnífica traza de avión en la imagen integrada con la media.

Diapositiva 35

Aquí tenemos los 4 zooms al 300%. En la media podemos ver casi de todo: tenemos la traza del avión, se aprecia una segunda traza de avión más débil, casi horizontal, pixeles calientes aquí y aquí y también pixeles fríos.

Si analizamos las otras 3 imágenes vemos que las trazas de avión y los pixeles calientes han desaparecido en las 3. Sin embargo en lo que respecta a los pixeles fríos, la imagen *Sigma Clipped* conserva muchos (de hecho todos, ya que hemos visto que no había recortado), la imagen *Linear Fit* está mucho mejor, pero muestra dos puntitos negros que son bastante visibles. La imagen *Winsorized*, tiene por el contrario el fondo de cielo muy suave y no presenta ningún cold pixel.

Por tanto (y tras haber hecho esta comprobación en otros puntos clave de la imagen) selecciono el algoritmo *Winsorized Sigma Clipping*.

Diapositiva 36

Una vez el elegido el algoritmo, ¿hemos acabado el trabajo? Está claro que lo podríamos dejar así, pero ese resultado lo podemos afinar, lo podemos mejorar. Vamos a ver como

Definimos una o más previsualizaciones, las necesarias para poder ver al menos un *outlier* de los más oscuros y uno de los más claros (un hot pixel o una traza de avión por ejemplo). Para hacer esto, utilizamos la imagen obtenida al principio, sin rechazo de pixels

A continuación comenzamos a ampliar el límite inferior. Al hacerlo obtendremos una imagen integrada con menos pixels rechazados y por tanto con un nivel menor de ruido. Sin embargo llegará un momento en el que nos empezarán a aparecer los píxeles oscuros que queremos evitar. Volver un poco atrás, justo para que desaparezcan otra vez: ese es nuestro valor de límite inferior.

Repetimos el proceso, ampliando el límite superior fijándonos ahora en los *outliers* brillantes. De esta forma llegaremos a fijar el límite superior.

Esto es lo que hemos hecho con nuestra imagen y hemos obtenido el siguiente resultado:

Diapositiva 37

El límite inferior que antes estaba en 4.0 lo hemos podido ampliar hasta 4.2 y el límite superior que antes que antes era 2.0 lo hemos podido ampliar hasta 2.5, y como podemos ver en el mismo zoom de antes, no se aprecian *outliers* ni blancos ni negros. Naturalmente estos ajustes han tenido un efecto beneficioso en la relación S/R de la imagen integrada. Vamos a verlo en nuestra tabla:

Diapositiva 38

En la última línea tenemos los valores de la imagen optimizada. El incremento de relación S/R que hemos alcanzado es 3.285. Esto significa que hemos aumentado un 5% la relación S/R con respecto a la imagen sin optimizar y que nos hemos quedado a un 96% del máximo teórico. Un buen resultado.

Observando los resultados, alguien se podría preguntar: el resultado inicial de *Linear Fit* es mejor que el que hemos obtenido al final ¿no podría ser que disminuyendo sus límites llegáramos a un punto satisfactorio de exclusión de *outliers* y que este punto tuviera una relación S/R mejor que la que hemos obtenido con *Winsorized*?

Y la respuesta es podría ser, hay que comprobarlo (de hecho yo lo hice y por poco es mejor *Winsorized*, no lo he incluido aquí por falta de tiempo)

Diapositiva 39

Bueno, vamos a ver el segundo ejemplo. Sucede que el astrofotógrafo que tomó estas imágenes es un poco obsesivo y a pesar del resultado que hemos visto no se conformó con 15 imágenes y al fin de semana siguiente tomó 7 más hasta un total de 22. Vamos a ver que mejora obtuvo. En primer lugar integramos las imágenes con la media

Diapositiva 40

A continuación con *Sigma Clipping*

Diapositiva 41

Winsorized Sigma clipping

Diapositiva 42

Y *Linear Fit Clipping*

Diapositiva 43

Y aquí tenemos la tabla resumen. Como antes nos concentramos en los incrementos de relación S/R y de nuevo *Linear Fit* parte con el mejor incremento.

Vamos a analizar las imágenes

Diapositiva 44

Aquí tenemos los recortes, y volvemos a hacer un zoom en la misma zona que antes

Diapositiva 45

Aquí volvemos a tener los zooms al 300%. En la imagen de la media no ha cambiado nada, seguimos viendo a todos nuestros amigos *outliers*. *Sigma clipped* clásico, a pesar de haber integrado 22 imágenes todavía presenta algún pixel frío, por lo que la descartamos (realmente este algoritmo yo ya no lo utilizo porque he comprobado en repetidas ocasiones que *Winsorized* es superior). Sin embargo *Linear Fit* sí que presenta un cambio, el fondo del cielo es muy uniforme y, a diferencia del ejemplo anterior, no presenta ningún pixel frío. Como su valor de relación S/R de partida es el mejor, lo seleccionamos.

Diapositiva 46

Procedemos a afinar el resultado según lo explicado y esto es lo que obtenemos:

El límite inferior que antes estaba en 5.0 lo hemos podido ampliar hasta 5.1 y el límite superior que antes era 2.5 lo hemos podido ampliar hasta 2.9. De nuevo el zoom nos muestra que no se aprecian *outliers*.

Veamos en nuestra tabla los efectos de estos ajustes en la relación S/R de la imagen integrada.

Diapositiva 47

En la última línea tenemos los valores de la imagen optimizada. El incremento de relación S/R que hemos alcanzado es 4.0604. Esto significa que hemos aumentado un 1.4% la relación S/R con respecto a la imagen sin optimizar y que nos hemos quedado en un 97.4% del máximo teórico. Un resultado excelente.

Otra cuestión sería saber si nuestro astrofotógrafo obsesivo se encuentra satisfecho de la operación:

La mejora en relación S/R con respecto a la integración de 15 imágenes ha sido del 23.6%, pero ese incremento se ha conseguido tras incrementar un 46% (7 sobre 15) el tiempo de exposición.

La respuesta a esta pregunta a mi modo de ver es muy subjetiva y cada uno debe responderla en función de sus objetivos personales.

Diapositiva 48

Último ejemplo. Seguimos con nuestro astrofotógrafo obsesivo. Ahora estamos con otro objeto. El pobre hombre tuvo una mala noche y le pasó de todo:

No paraban de pasar aviones, tuvo un problema de guiado..... y al final, se nubló. En definitiva él considera que tiene solamente 4 imágenes decentes para integrar. Las que aquí tenemos.

Veamos en primer lugar que puede conseguir integrando 4 imágenes:

Diapositiva 49

Repetimos los pasos ya vistos y cómo vamos mal de tiempo pasamos directamente a la tabla.

Dado el reducido número de imágenes de que disponemos hemos atacado directamente con *Percentile Clipping* y tras afinar los límites obtenemos lo que indica la tabla:

Bueno, hemos conseguido una pequeña mejora, pero ¿es eso todo lo que podemos conseguir?

En la papelera de reciclaje tenemos 2 imágenes descartadas, una con una traza de avión y otra con un claro error de guiado.

Bueno, la del error de guiado está claro que no la vamos a hacer servir, pero la de la traza del avión ¿Por qué no?

Y no penséis que esta es una pregunta retórica, no: en Àger veo todavía a compañeros que descartan estas imágenes, y se puede extraer información válida de ellas. En la introducción ya vimos que con 6 imágenes podíamos salvar una traza de avión. Vamos a ver si ahora podemos hacerlo con tan solo 5

Diapositiva 50

Aquí tenemos las 5 imágenes. Repetimos, abreviadamente, los pasos anteriores y obtenemos lo siguiente:

Diapositiva 51

De nuevo el mejor algoritmo es *Percentile Clipping* y gracias a él obtenemos una imagen sin la traza del avión y con una mejora del 6% en la relación señal / ruido con respecto a la integración de 4 imágenes.

Nuestro astrofotógrafo se siente contento, pero piensa, ¿puedo hacer algo más?

¿Cómo está la imagen con error de guiado??

Diapositiva 52

Bueno, la imagen tiene un claro defecto de guiado, éste ha dado un “salto”, aquí lo podemos ver a tamaño 1:1.

Nuestro astrofotógrafo, que ya empieza a estar entrado en años y empieza a notar que su memoria ya no es la que era, se estruja el cerebro y recuerda haber leído a principios de los 2000 un artículo en *Sky & Telescope* con un método para aprovechar imágenes como ésta. Pero era muy complicado, había que restar una imagen “buena” a la que tenía el defecto de guiado para obtener ese defecto y luego restar ese “defecto” a la imagen original para eliminarlo, y luego combinar las imágenes. Muy complicado. ¿Hará falta hacer todo eso?, se pregunta nuestro hombre, o ¿quizás uno de los algoritmos *Sigma Clipped* pueda llevárselo por delante?

Vamos a verlo

Diapositiva 53

En la tabla podemos ver que la inclusión de la sexta imagen nos ha permitido utilizar un algoritmo de rechazo más potente (*Averaged Sigma Clipping*) y ello no ha llevado a obtener un incremento de la **SNR** de 1,8930 y una mejora del **24,2%** con respecto a la integración de 4 imágenes.

Diapositiva 54

El análisis de las imágenes muestra que todo ello se ha conseguido con un buen rechazo de outliers: a la izquierda podemos ver las integraciones de 4, 5 y 6 imágenes, todas ellas sin traza de avión y a la derecha, arriba un detalle de la integración de 4 imágenes y abajo, de 6 imágenes. La diferencia en ruido de fondo es muy evidente aunque quizás la proyección no permite apreciarlo.

Por último para asegurarnos que la inclusión de la 6ª imagen no ha degradado el resultado, además de la inspección visual obtenemos los siguientes valores con *CCDInspector*, que muestran que tanto el *FWHM* como el *aspect ratio* no se han visto perjudicados (de hecho no solo no se han visto afectados sino que ambos han mejorado ligeramente)