Introducción a Wavelets y Procesamiento Multiescala

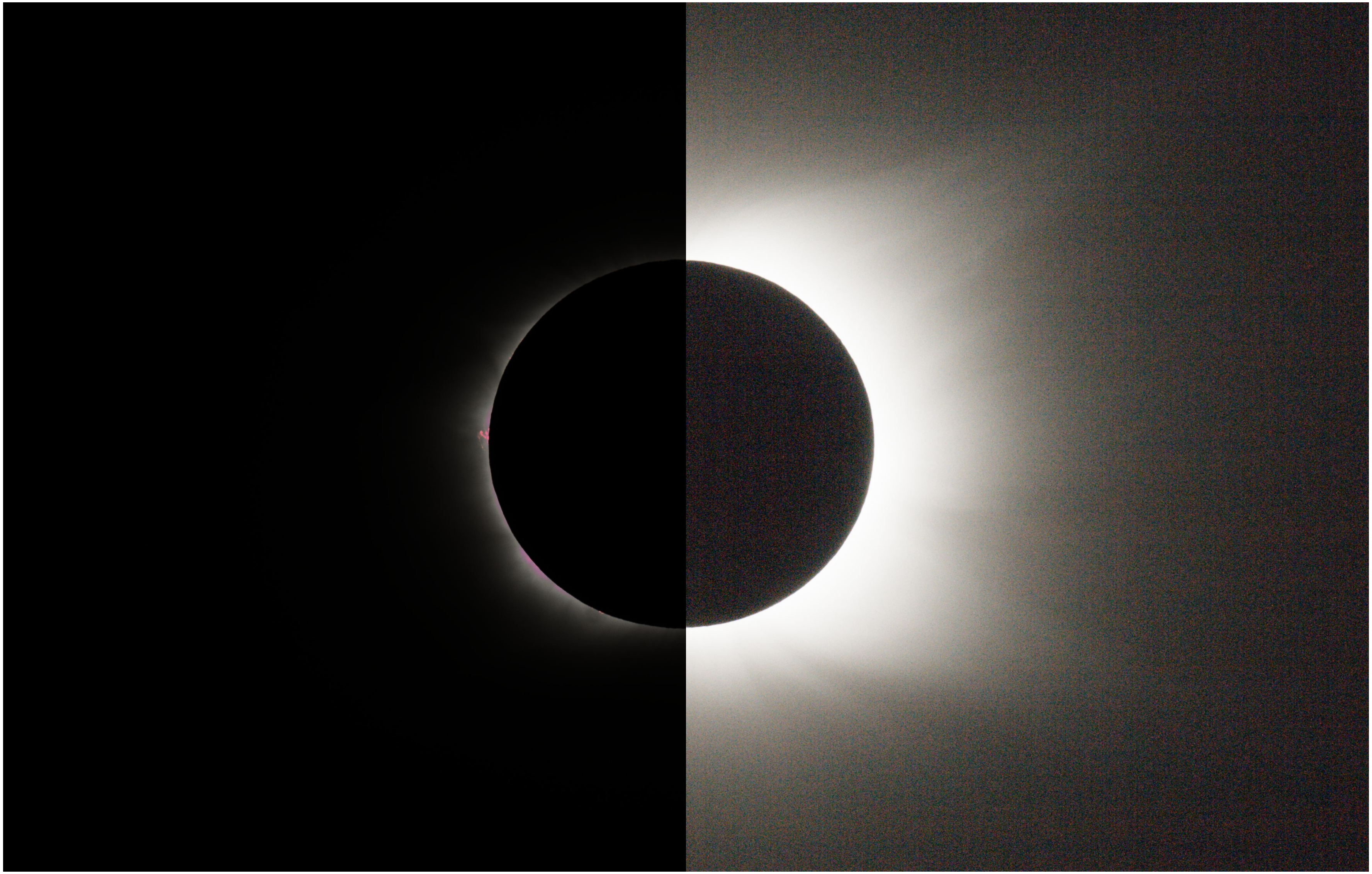
Carlos Milovic F.[1,2]
CAA 2011

1: PixInsight Development Team, Pleiades Astrophoto
2: Centro de Imágenes Biomédicas, Pontificia Universidad Católica de Chile

Las imágenes tienen ruido...

Pueden estar difusas...

O no podemos desplegar su información de manera simple...

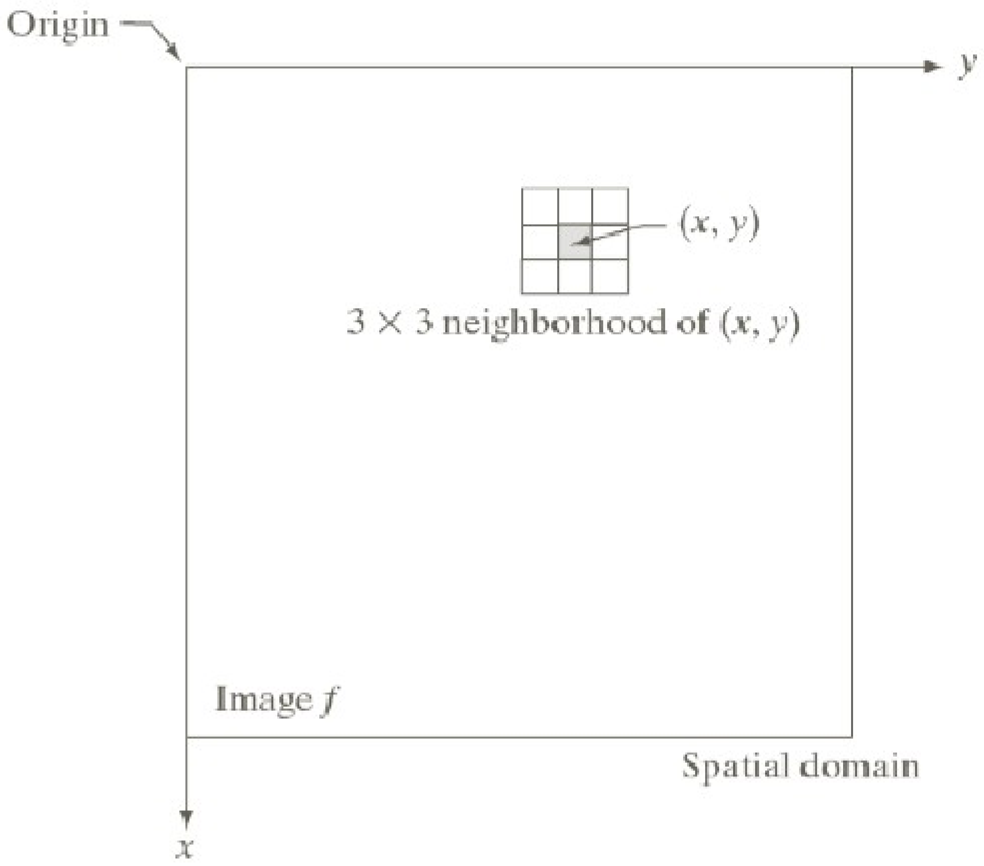


En resúmen, las imágenes son imperfectas y complejas

- Necesitamos hacer más que ajustar la intensidad de pixeles individualmente.
- Funciones que tomen en cuenta pixeles vecinos, y operen entre ellos.
- Tomar decisiones inteligentes en base a información espacial e intensidades.

Filtrado en el espacio

- Promediemos pixeles vecinos...

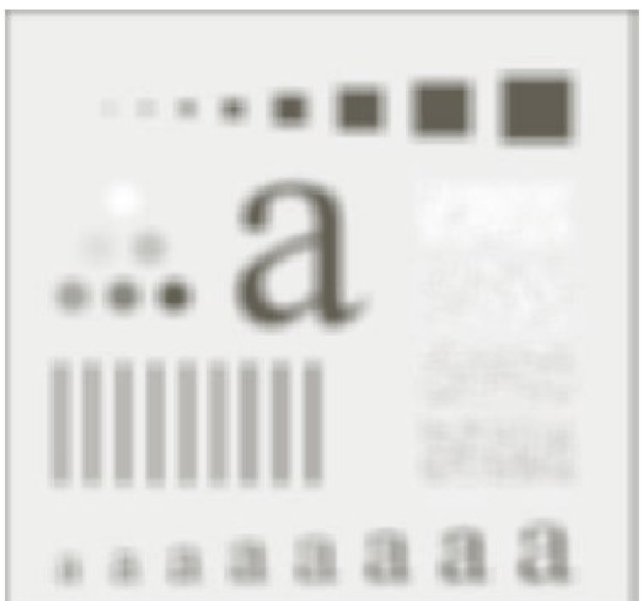


Filtro Promedio

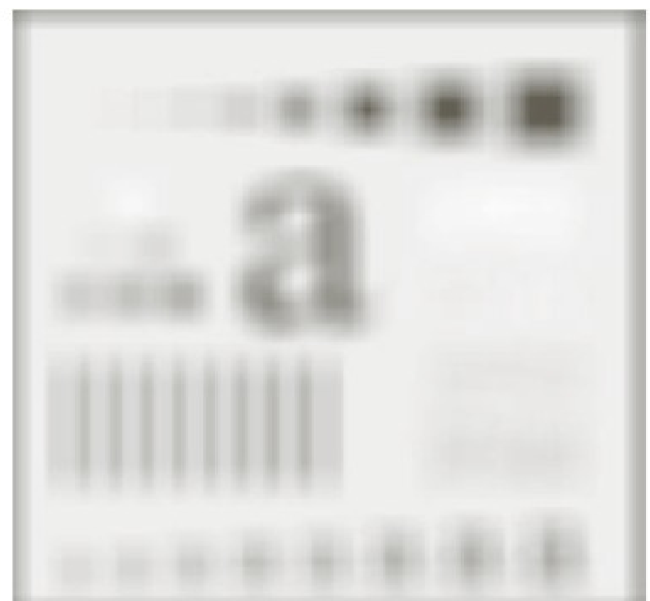
Original

Promedio
3x3

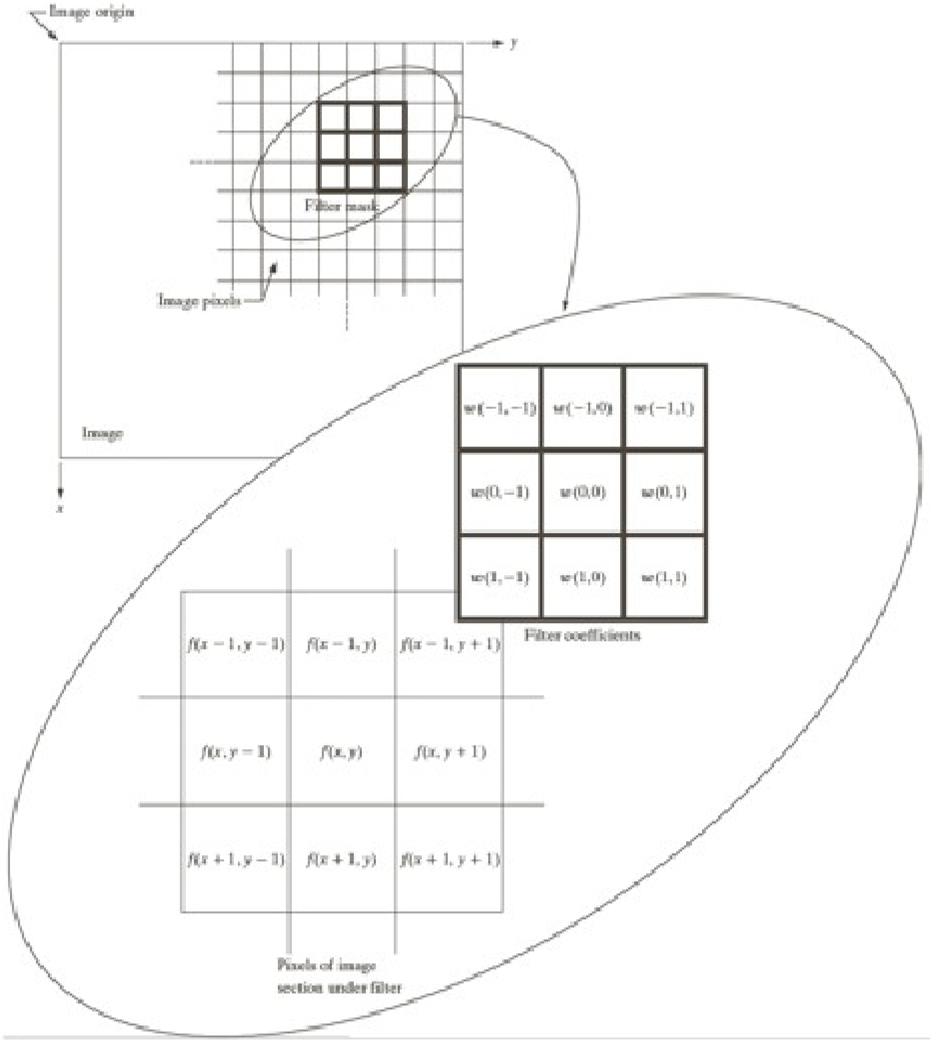
Promedio
15x15



Promedio
25x25



Filtros espaciales en general



Promedio ponderado

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \Rightarrow \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

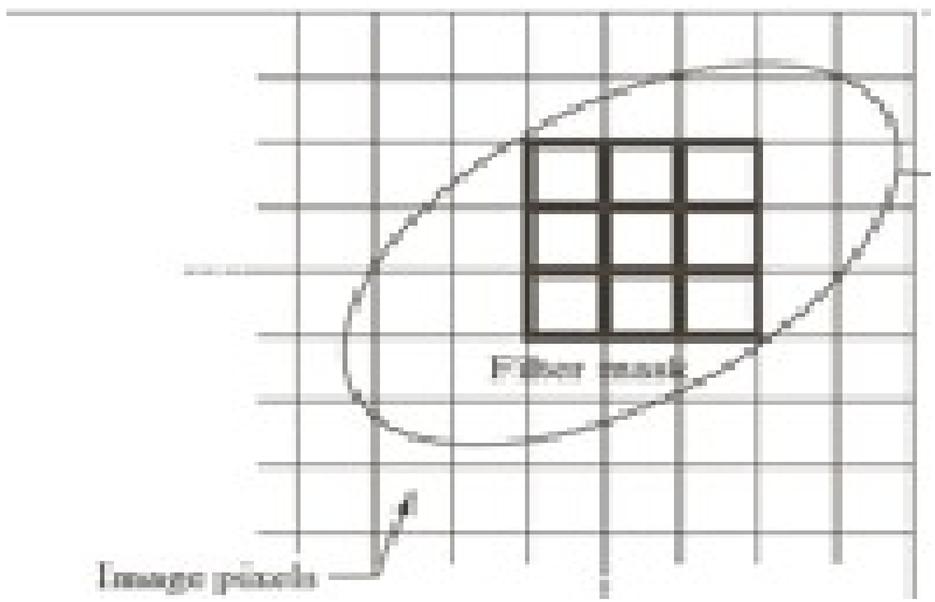
Promedio simple 3x3

Promedio ponderado 3x3

Convoluciones

- Matemáticamente, el filtrado con una máscara w(x,y) la podemos escribir:

$$w(x,y) * f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t)$$



Filtros de suavizado (o desenfoque)

- Son promedios ponderados.
- Se usan funciones para encontrar valores de las máscaras.
    - Lineal
    - Desenfoque gaussiano
- Eliminan variaciones locales en las imágenes.

Filtros de agudizado (o enfoque)

- Veamos el filtro “Laplaciano”:

0	−1	0
−1	4	−1
0	−1	0

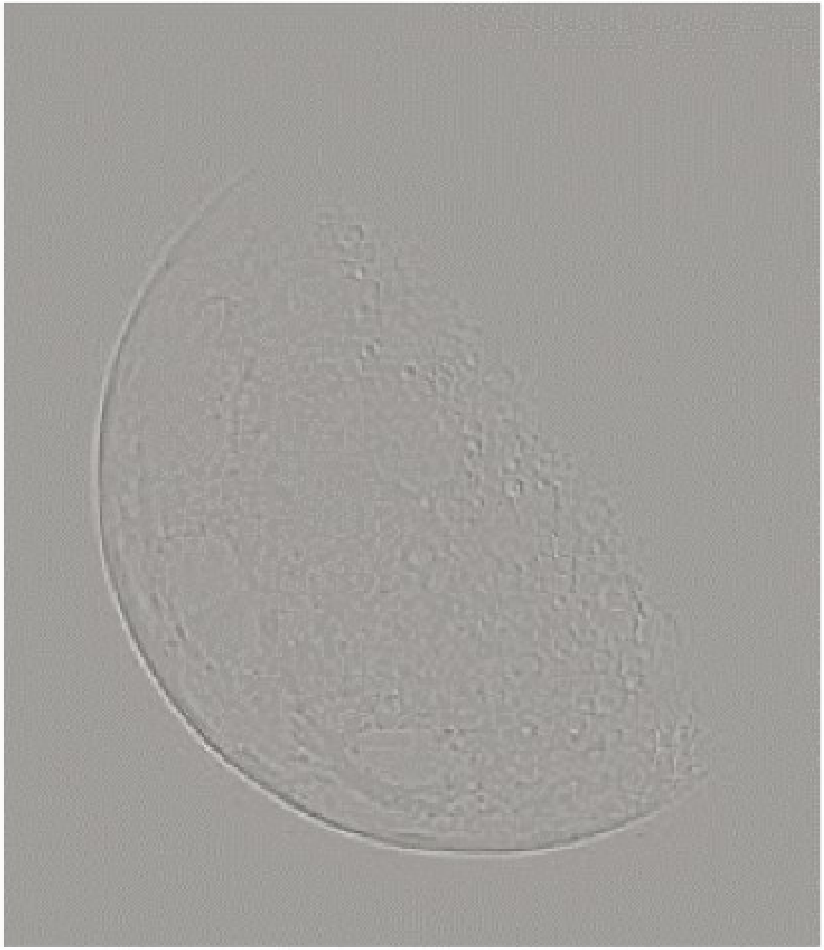
−1	−1	−1
−1	8	−1
−1	−1	−1

- Equivale a comparar un pixel con sus vecinos.

Filtro Laplaciano

Imagen original

Laplaciano escalado



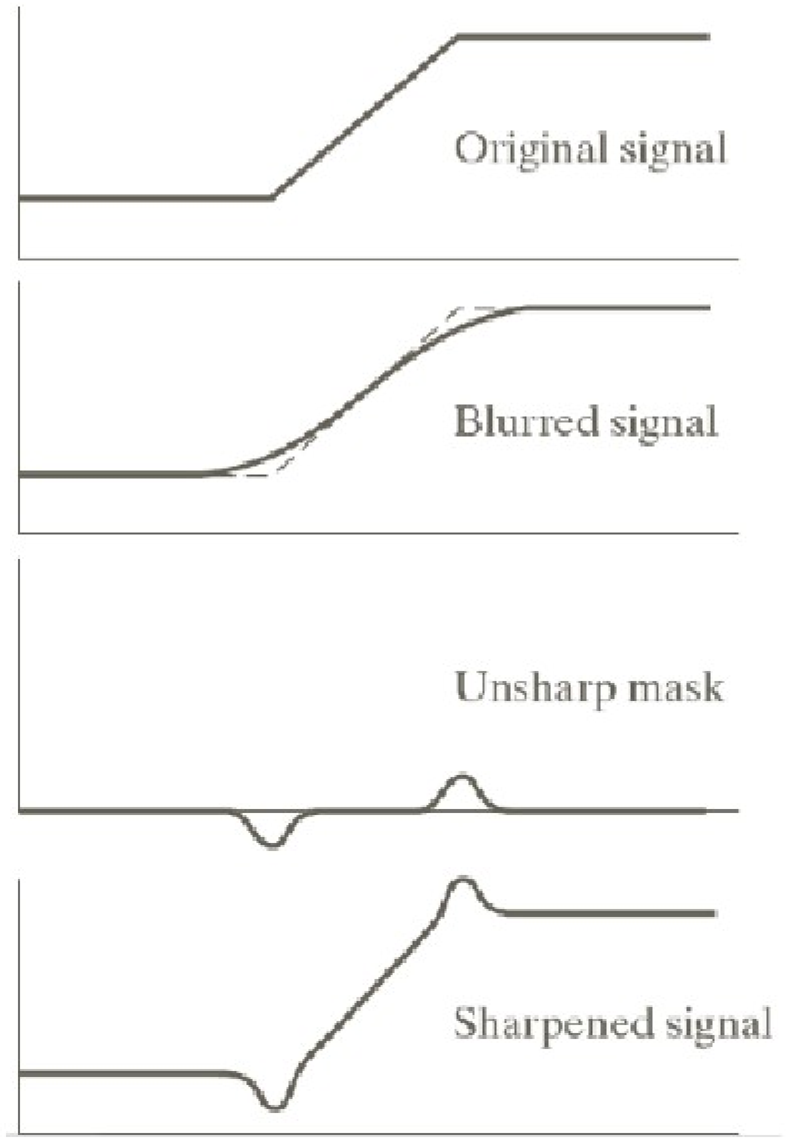
Filtro de Agudizado (Sharpen)

Imagen original

Laplaciano sumado a imagen

Unsharp Mask

Unsharp - mask



El tamaño o “radio” del realce dependerá del tamaño del filtro de suavizado.

Convoluciones

- Herramienta poderosa para procesar las imágenes.
- Sin embargo, son muy caras computacionalmente. Requiere muchos cálculos, y es lento [O(n^2)].
- Filtros grandes se vuelven prohibitivos.
- Esto nos limita el tamaño de las estructuras que queremos detectar, destacar o suavizar.

Transformada de Fourier

- Nos propone una vía para implementar convoluciones grandes, y tener más control de los filtros.
- En vez de trabajar en el dominio del espacio, trabajamos en el dominio de la frecuencia.
- La imagen se modela como una suma de senos y cosenos de distinta frecuencia.

Transformada de Fourier

- Nos propone una vía para implementar convoluciones grandes, y tener más control de los filtros.
- En vez de trabajar en el dominio del espacio, trabajamos en el dominio de la frecuencia.
- La imagen se modela como una suma de senos y cosenos de distinta frecuencia.

Transformada de Fourier

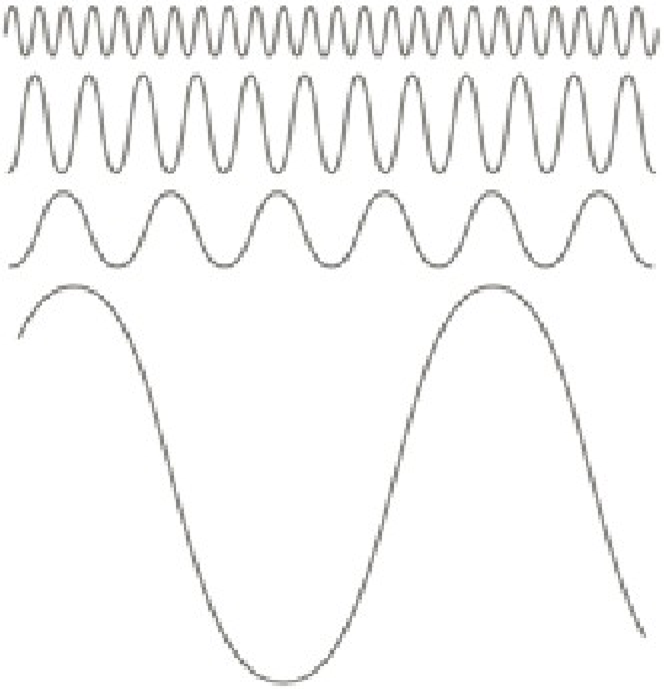
- Caso discreto 2D:

$$F(\omega, \nu) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(\omega x/M + \nu y/N)}$$

- Y su inversa:

$$f(x, y) = \frac{1}{MN} \sum_{\omega=0}^{M-1} \sum_{nu=0}^{N-1} F(\omega, \nu) e^{j2\pi(\omega x/M + \nu y/N)}$$

Transformada de Fourier



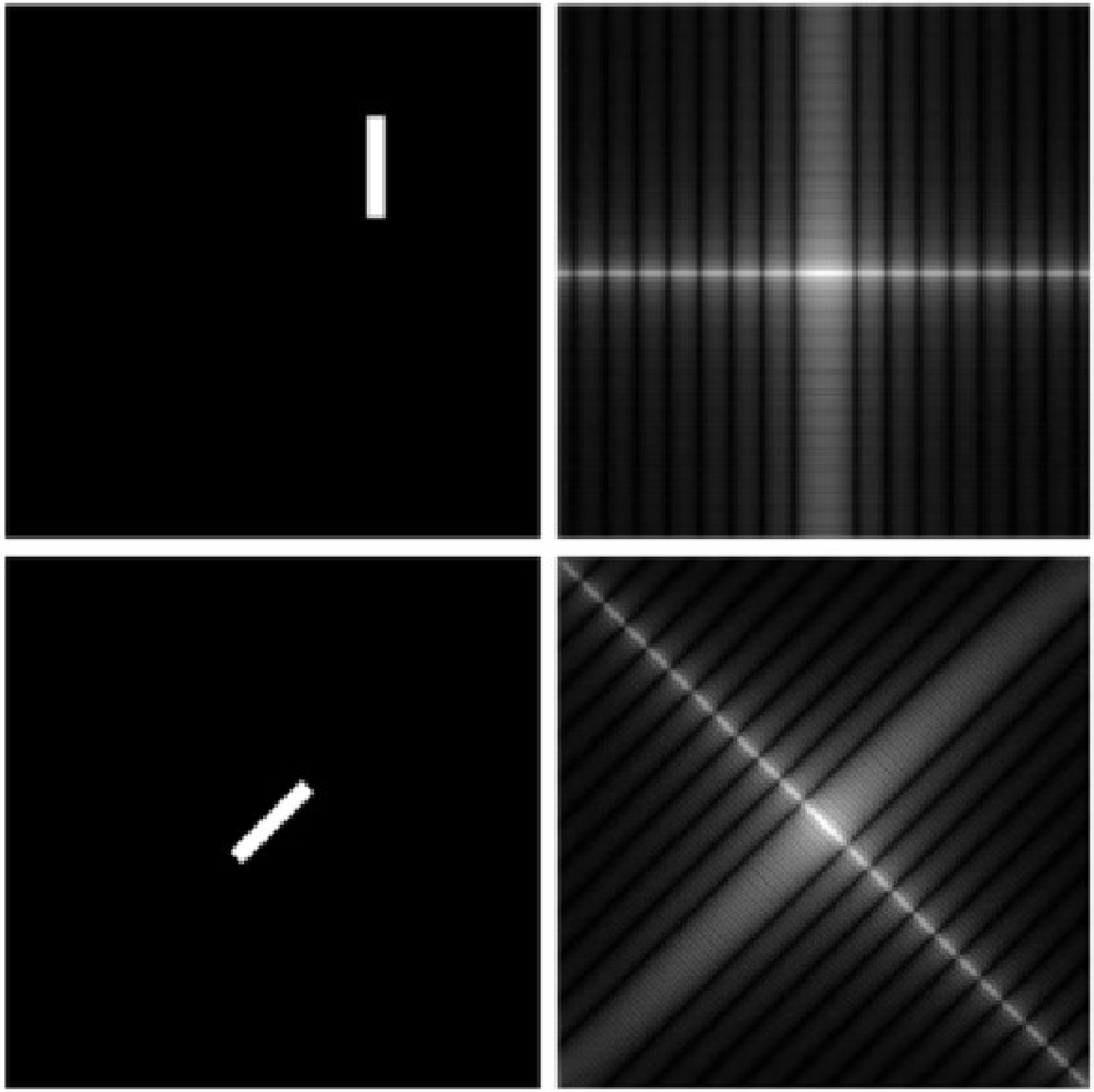
Cuatro funciones senosoidales con distinta frecuencia

Suma de estas funciones.

Ejemplos

Rectángulo
desplazado

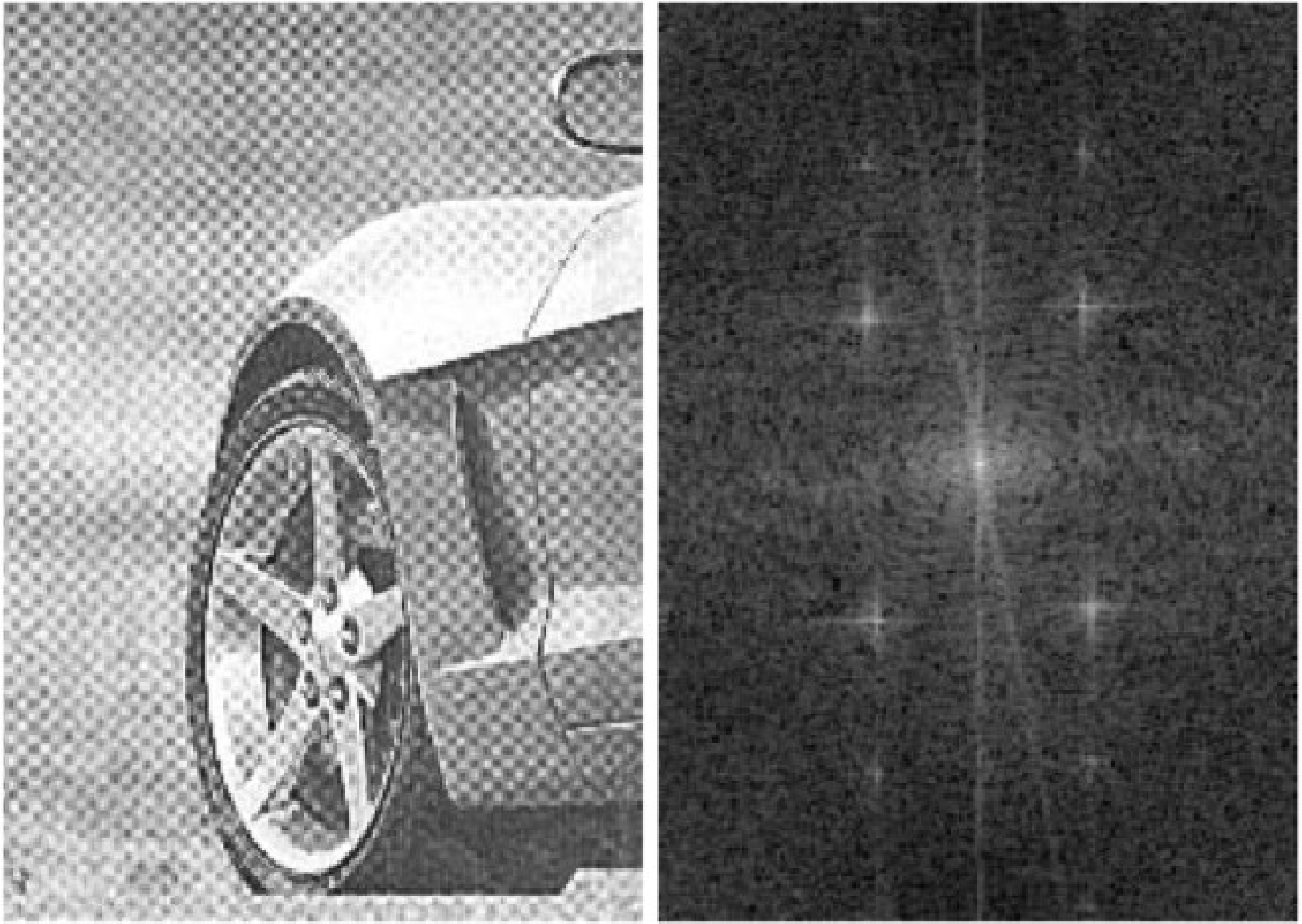
Espectro



Rectángulo
rotado

Espectro

Ejemplos



Convoluciones

- Las convoluciones cumplen con la siguiente propiedad:

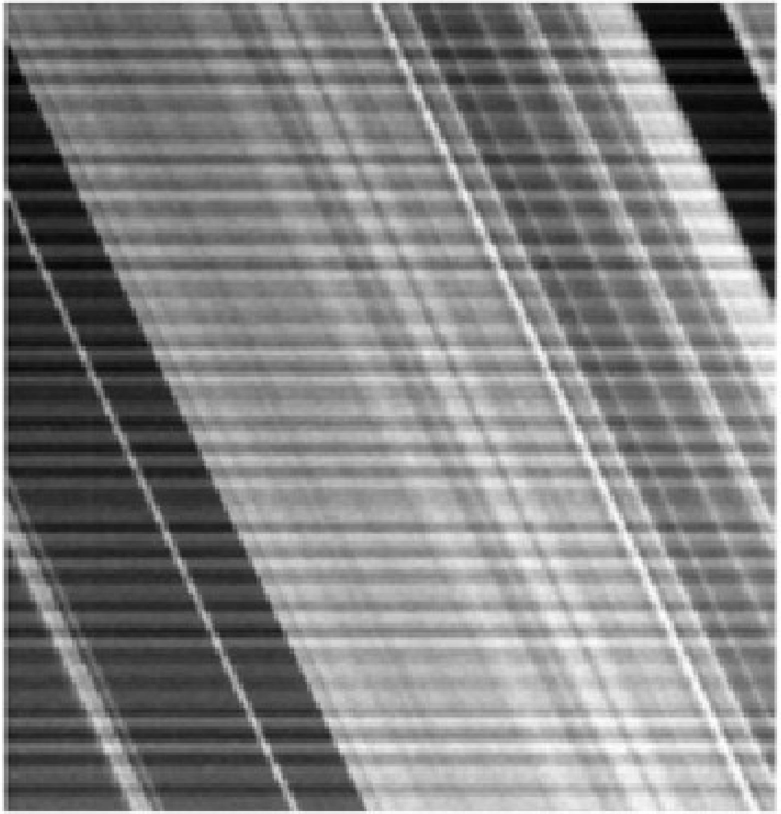
$$f(x,y) * h(x,y) \Longleftrightarrow F(\omega,\nu)H(\omega,\nu)$$

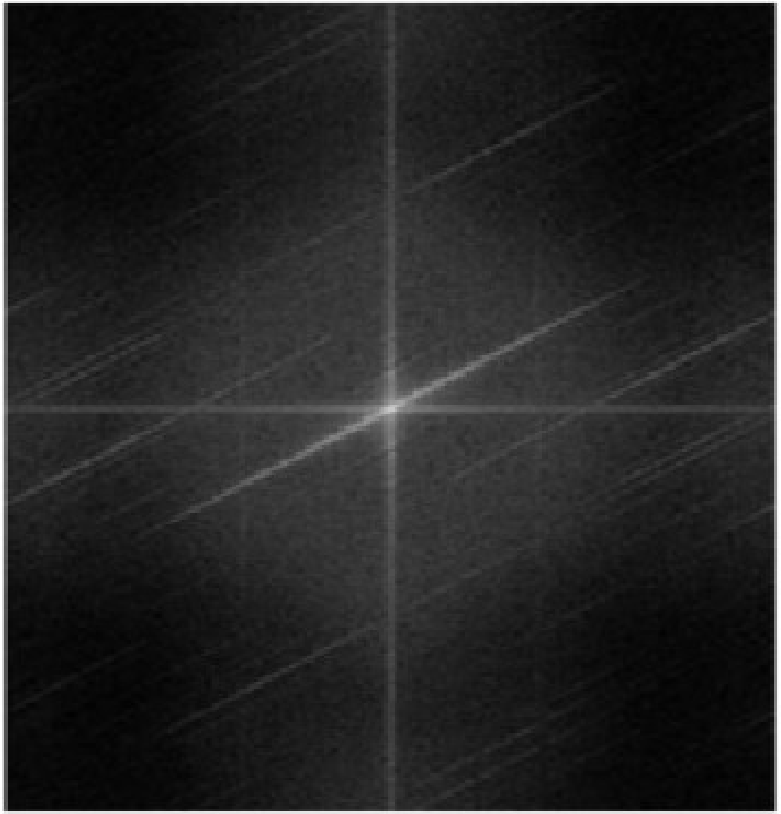
- Basta con multiplicar los coeficientes de Fourier para efectuar las convoluciones espaciales.
- Podemos operar directamente en el espacio de frecuencia.

Filtro Notch

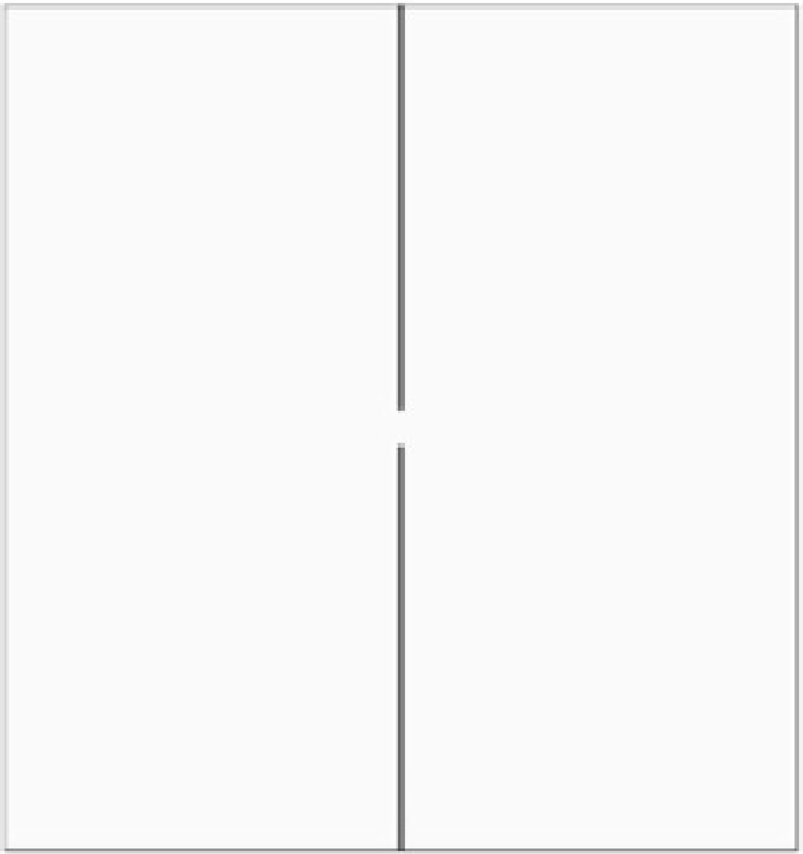
Imagen de los anillos de Saturno que muestra una interferencia casi periódica

Espectro de la imagen





Filtro Notch



Filtro vertical de rechazo en frecuencia

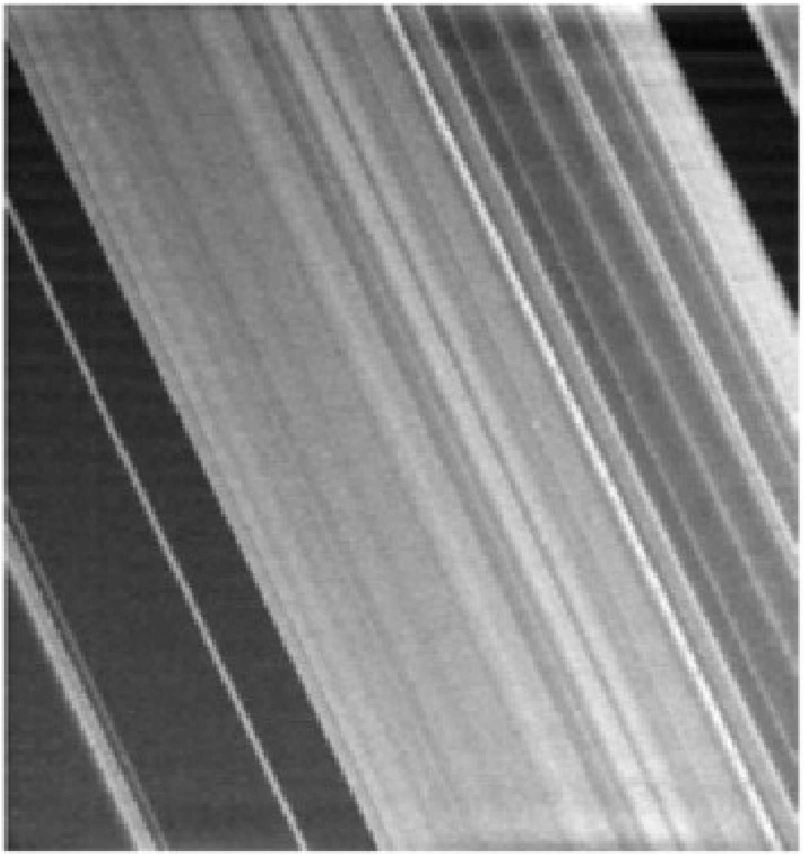
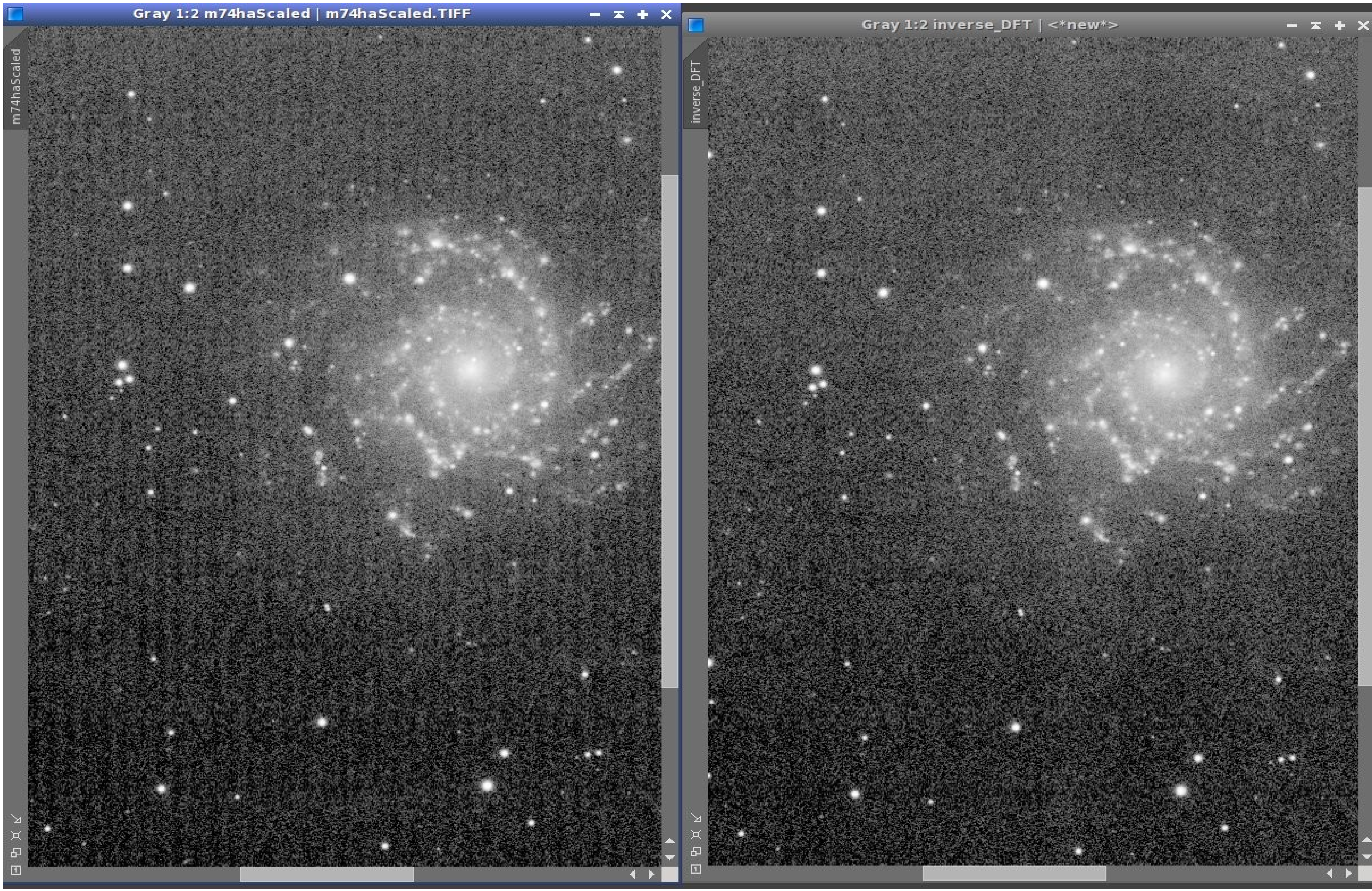


Imagen filtrada

Filtro Notch



Filtro Rechaza-banda

1

2

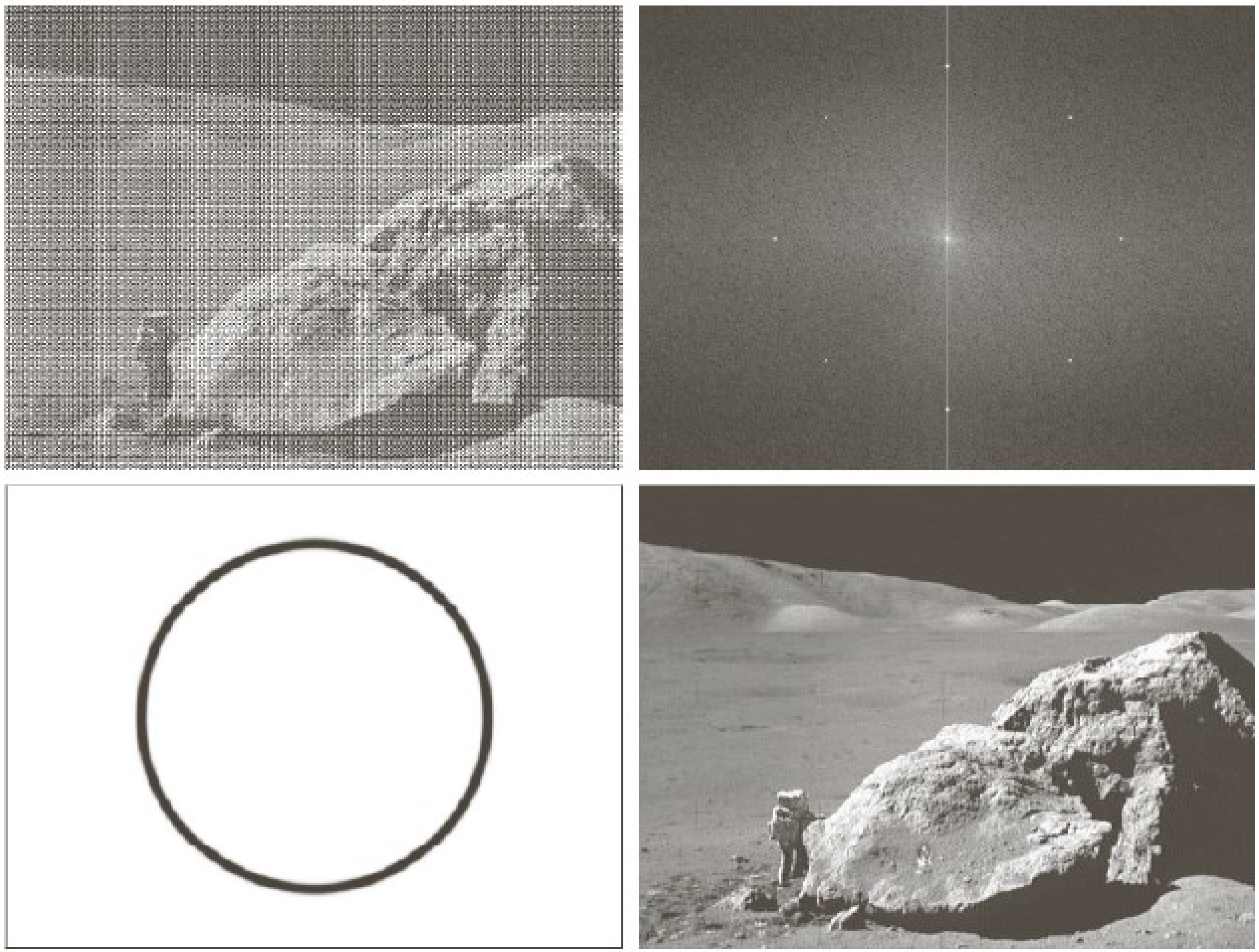


Imagen corrupta por ruido sinusoidal

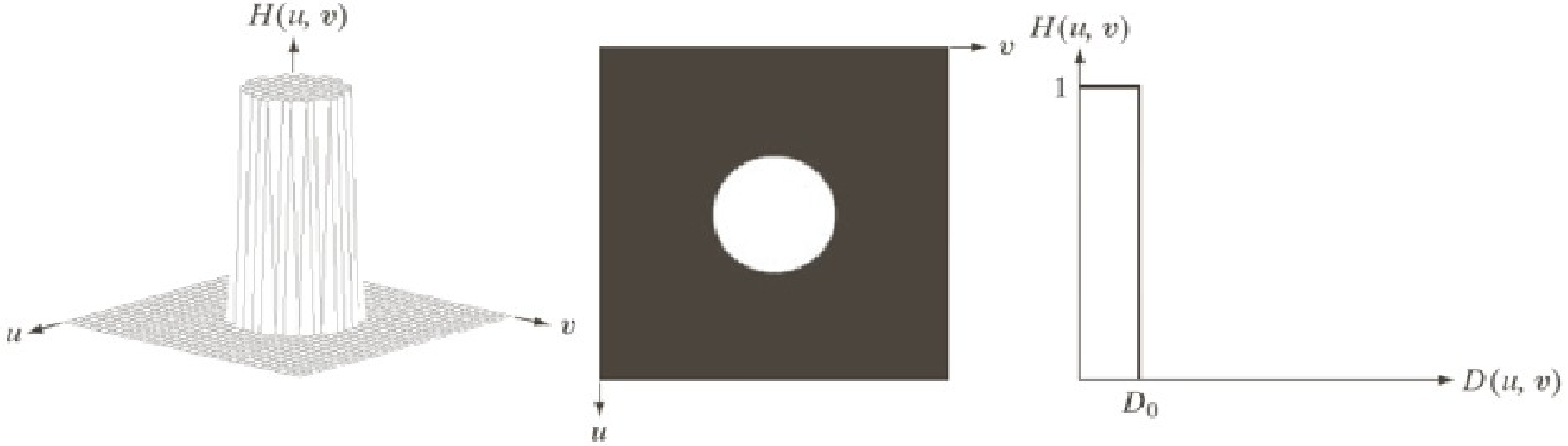
Transformada de Fourier de la imagen

Máscara utilizada para eliminar los *peaks* de energía

Transformada inversa de Fourier del producto entre 2 y 3

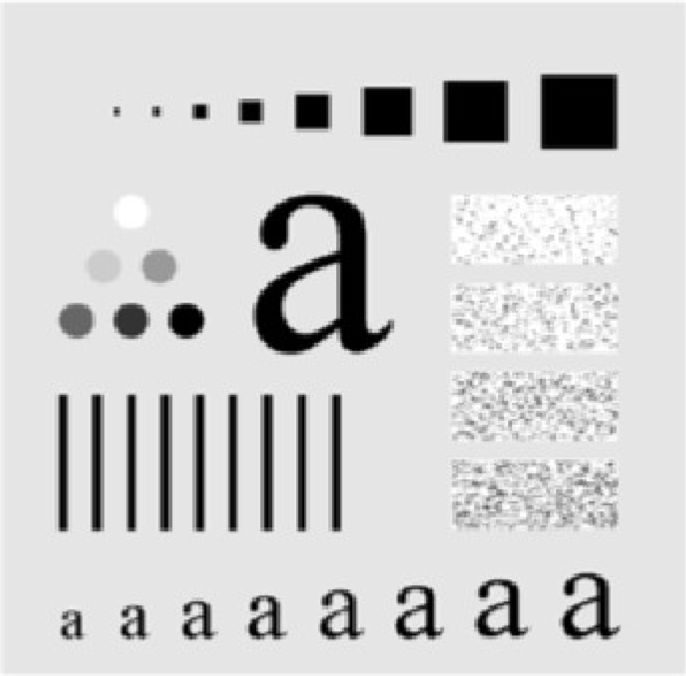
Filtros suavizado (pasa bajos)

Filtro pasabajos ideal.



Filtros suavizado (pasa bajos)

Filtro pasabajos ideal.



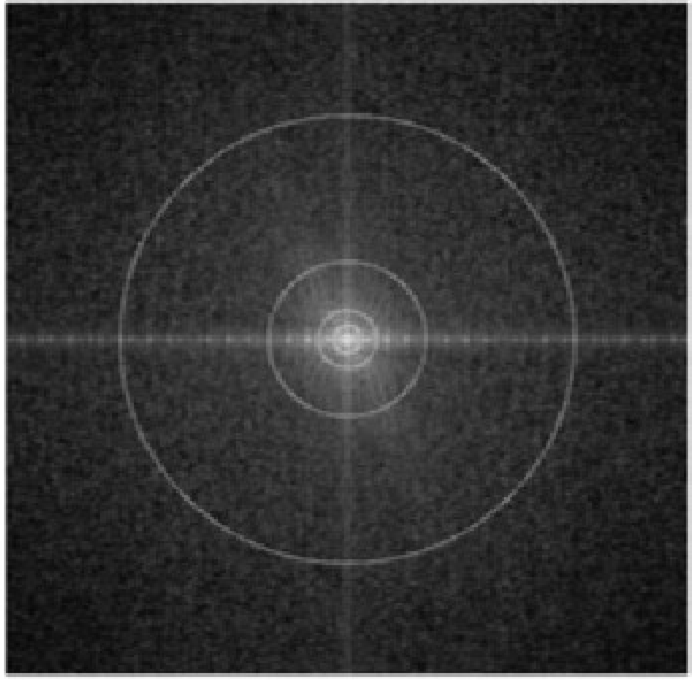


Imagen de prueba de 688 x 688

Espectro de Fourier con círculos sobreimpuestos de radio 10, 30, 60,160 y 460, encerrando 87.0, 93.1, 95.7, 97.8. y 99.2 % de la potencia de la imagen

Filtros suavizado (pasa bajos)

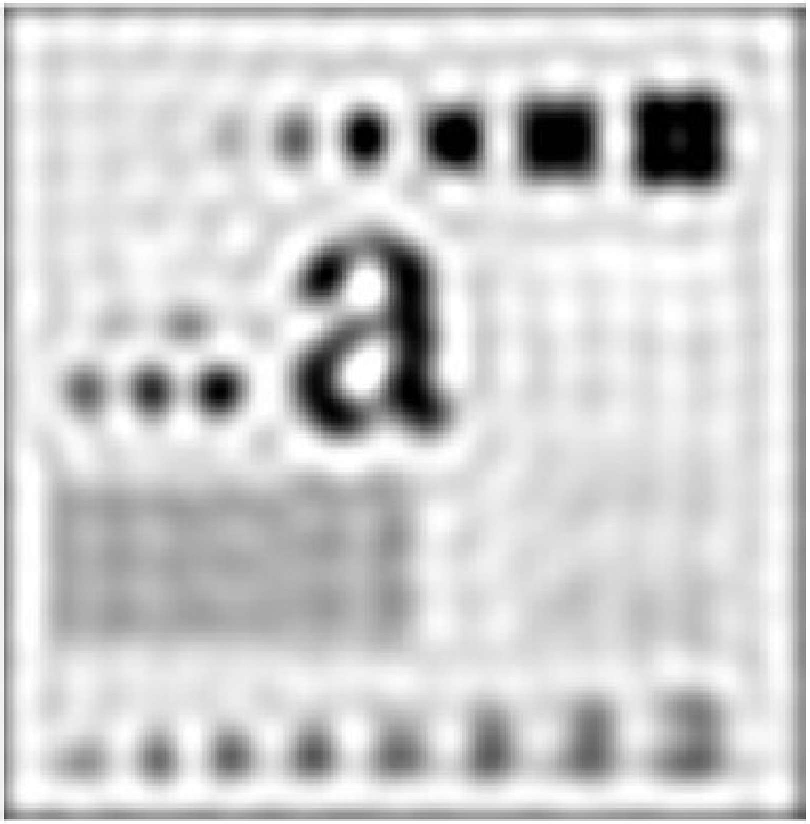


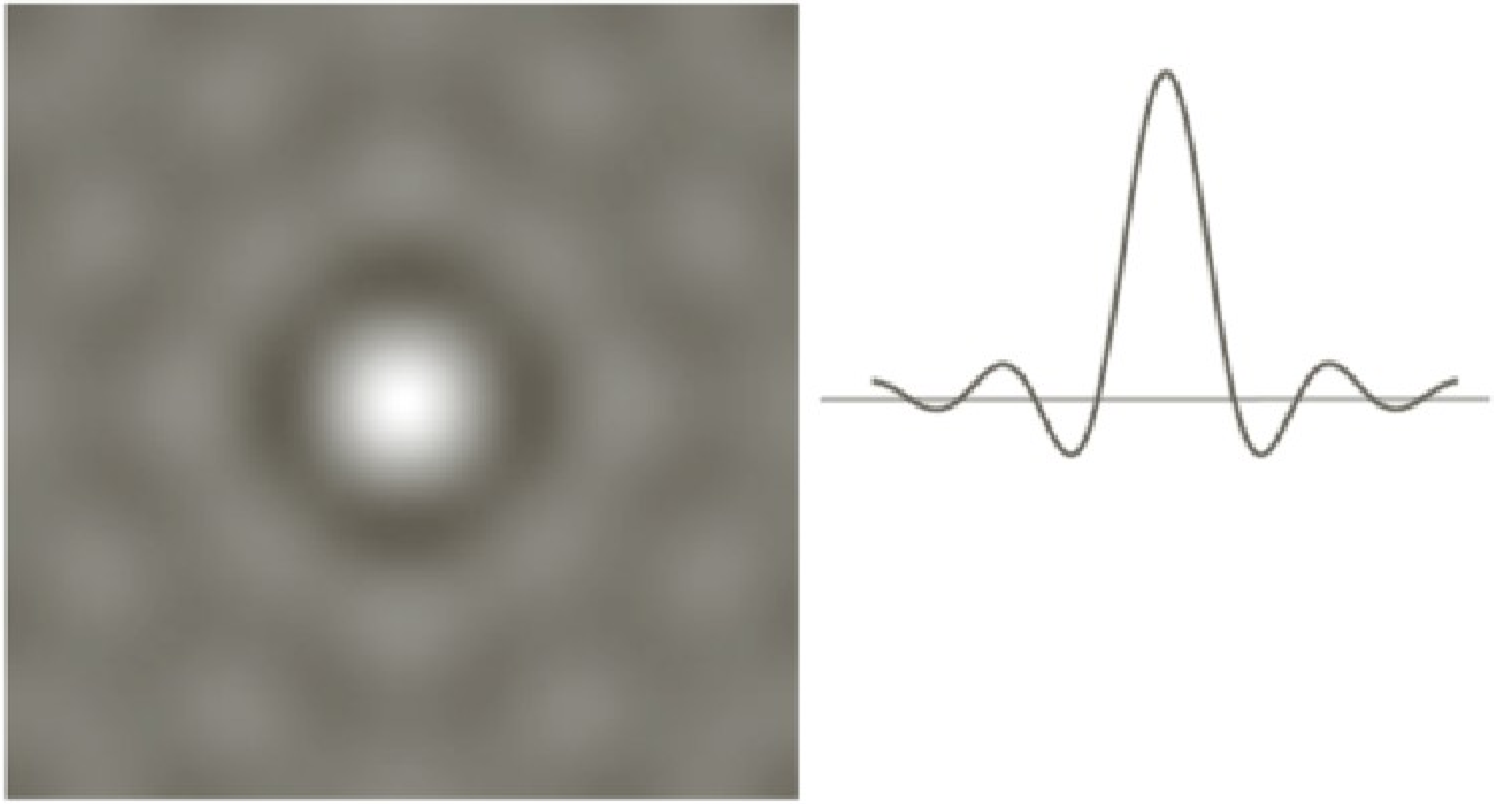
Imagen de prueba filtrada con filtro de radio 30

Imagen de prueba filtrada con filtro de radio 60

Filtros suavizado (pasa bajos)

Filtro pasabajos ideal

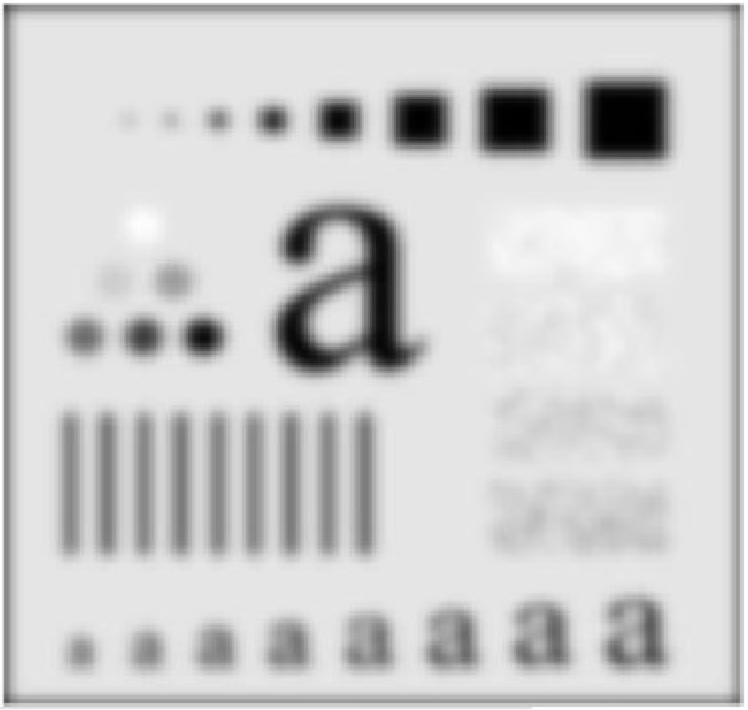
Representación en el dominio del espacio de un filtro pasabajos ideal de radio 5 y tamaño 1000 x 1000



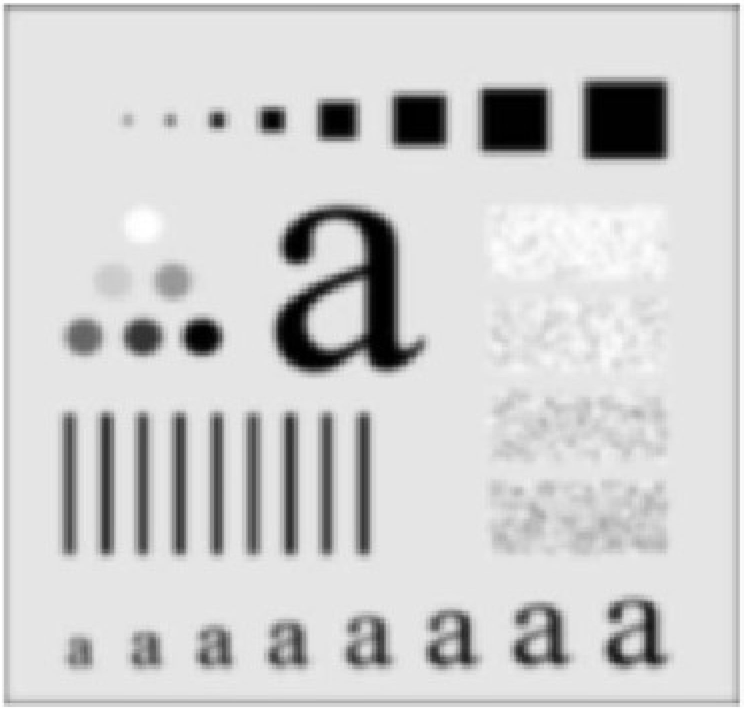
Filtros suavizado (pasa bajos)

Filtro pasabajos Gaussiano

Filtro PB Gaussiano con frecuencia de corte = 30



Filtro PB Gaussiano con frecuencia de corte = 60

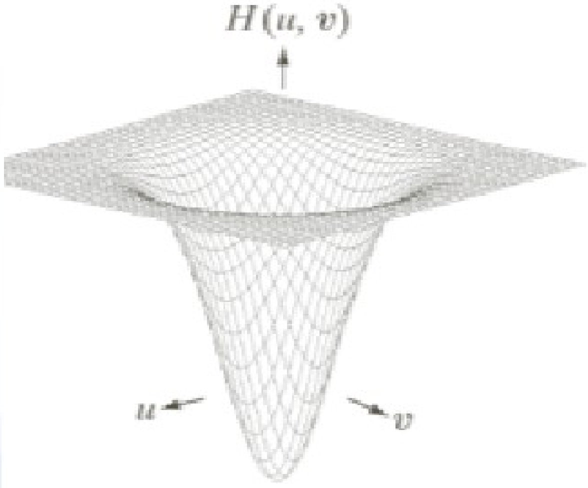


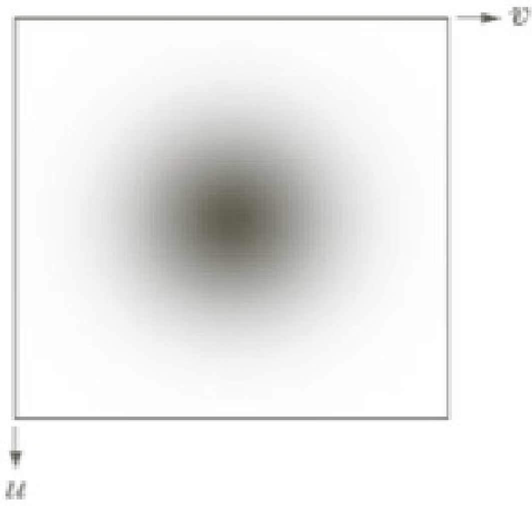
Filtros agudizado (pasa altos)

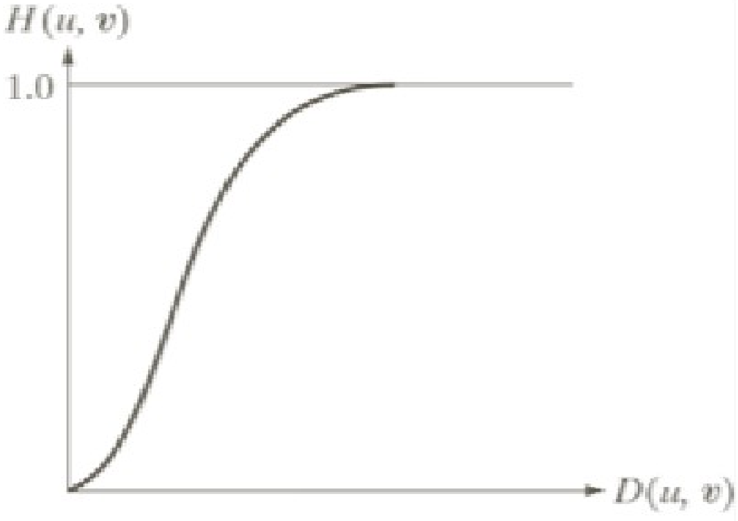
- Básicamente la idea es la misma, salvo que atenuamos las bajas frecuencias.
- Un pasa altos es el “inverso” de un pasa bajos.

Filtros agudizado (pasa altos)

Filtro pasa-altos Gaussiano



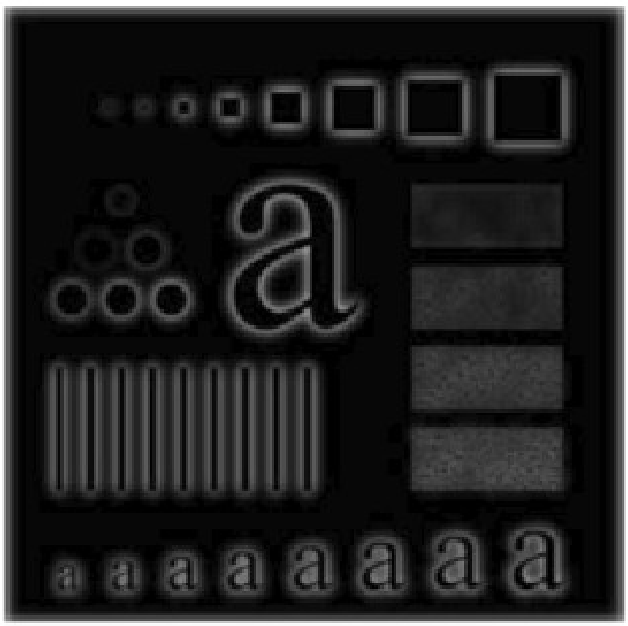




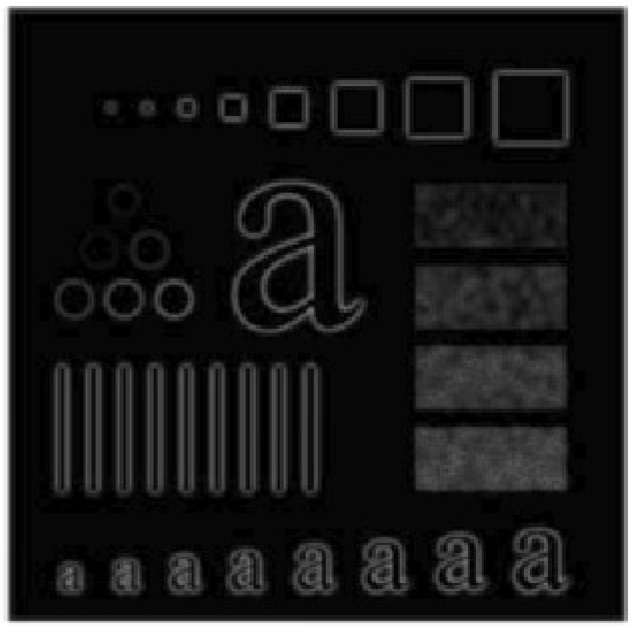
Filtros agudizado (pasa altos)

Filtros pasa-altos Gaussiano (aplicación a la imagen de prueba)

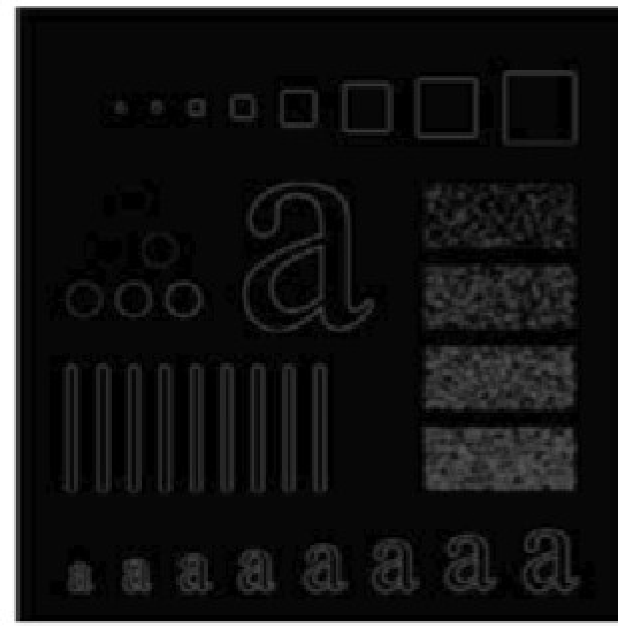
Do = 30



Do = 60



Do = 160



Limitaciones de Fourier

- Representa frecuencias de funciones senosoidales solamente.
- Estructuras no periódicas son difíciles de representar o aislar.
- Es no localizada. Cambiar un coeficiente de Fourier afecta a toda la imagen.

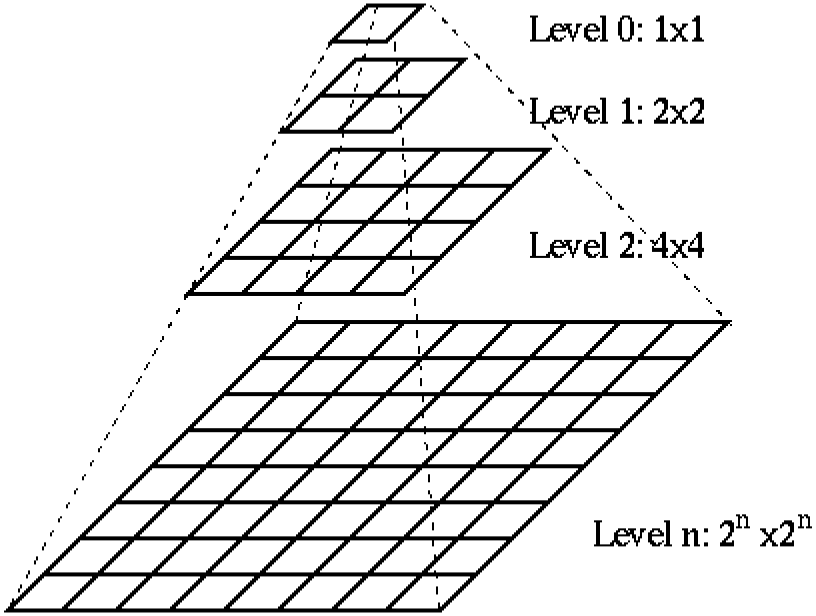
- Queremos buscar una nueva forma de representar a una imagen.

Wavelets

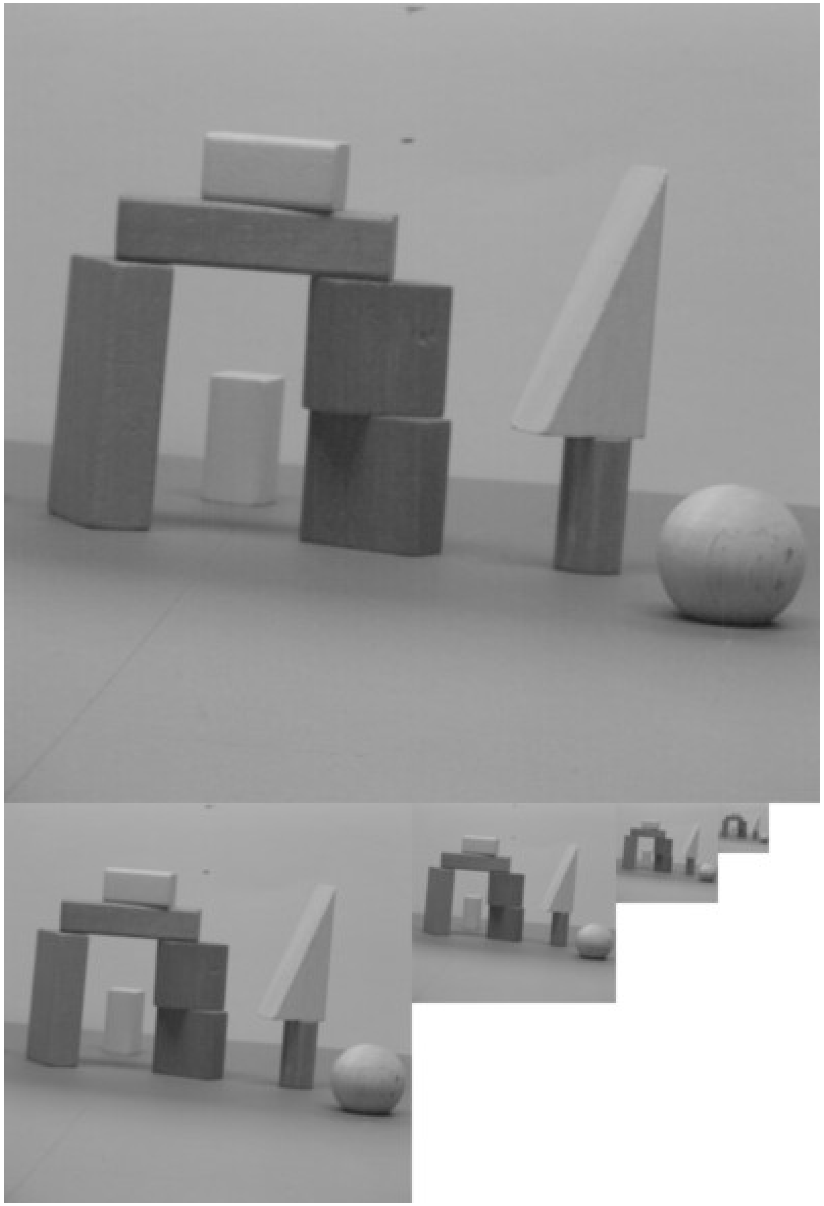
- De forma similar a Fourier, representaremos las imágenes con funciones.
- Ellas estarán localizadas tanto en el espacio, como en la frecuencia.
- Hagamos una deducción gráfica de los coeficientes de una transformación de wavelets.

Pirámide Gaussiana

- Tomemos una imagen, y la desenfocamos con un filtro gaussiano pequeño (por ejemplo, con sigma = 1). Luego, la achicamos a la mitad. Repetimos el proceso hasta tener 1px.

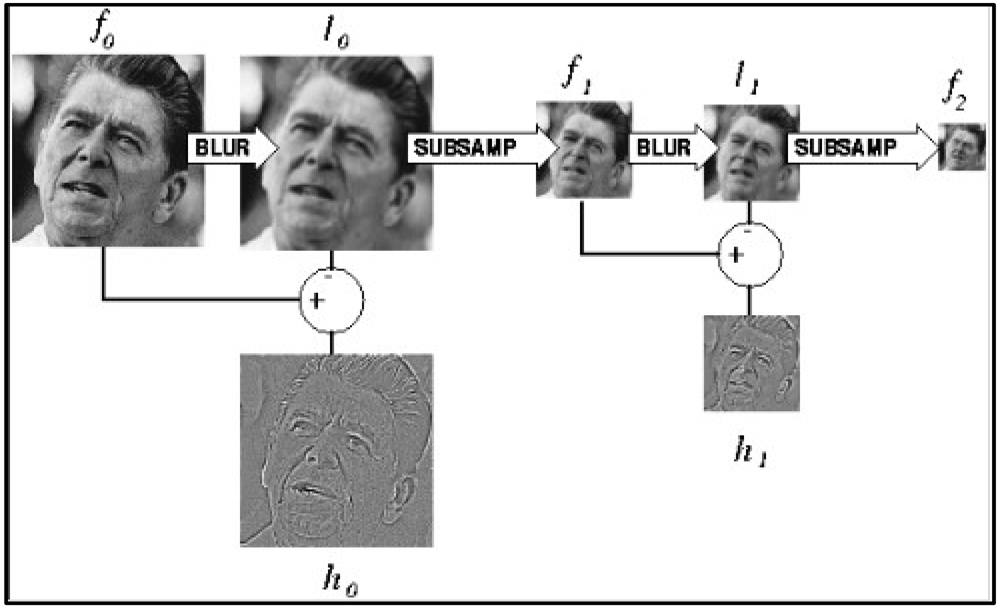


Pirámide Gaussiana - Ejemplo

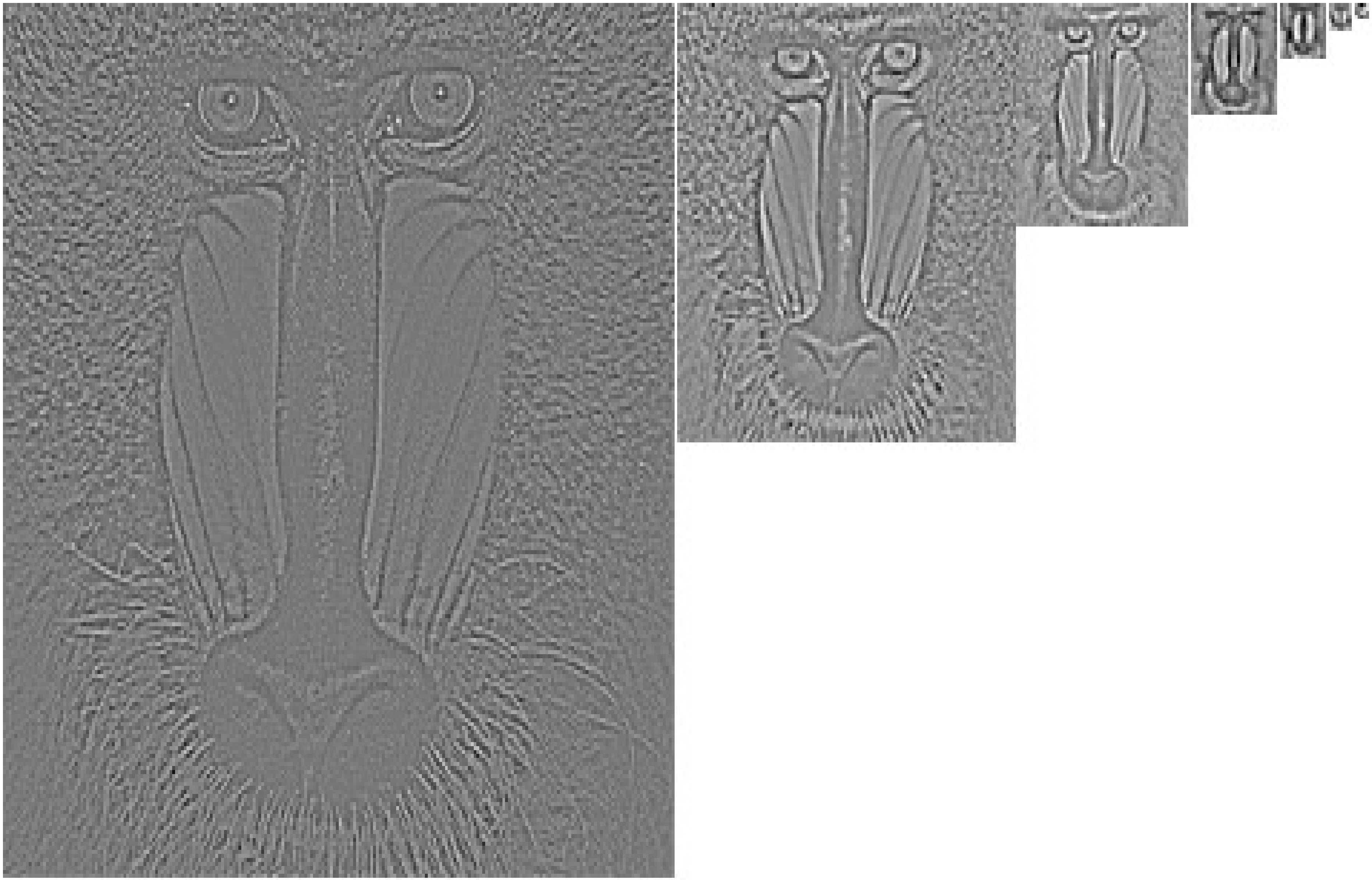


Pirámide Laplaciana

- En vez de ir guardando la imagen desenfocada, guardamos la diferencia con la imagen sin desenfocar, para cada capa.



Pirámide Laplaciana - Ejemplo



Análisis Multirresolución

- Estas pirámides nos permiten analizar estructuras a distintas escalas.
- La pirámide laplaciana aísla detalles de una escala particular.
- Computacionalmente ocupa poco espacio, y es rápida de generar, pero dificulta el análisis gráfico.
- Para reconstruir, hay que agrandar e interpolar las capas.

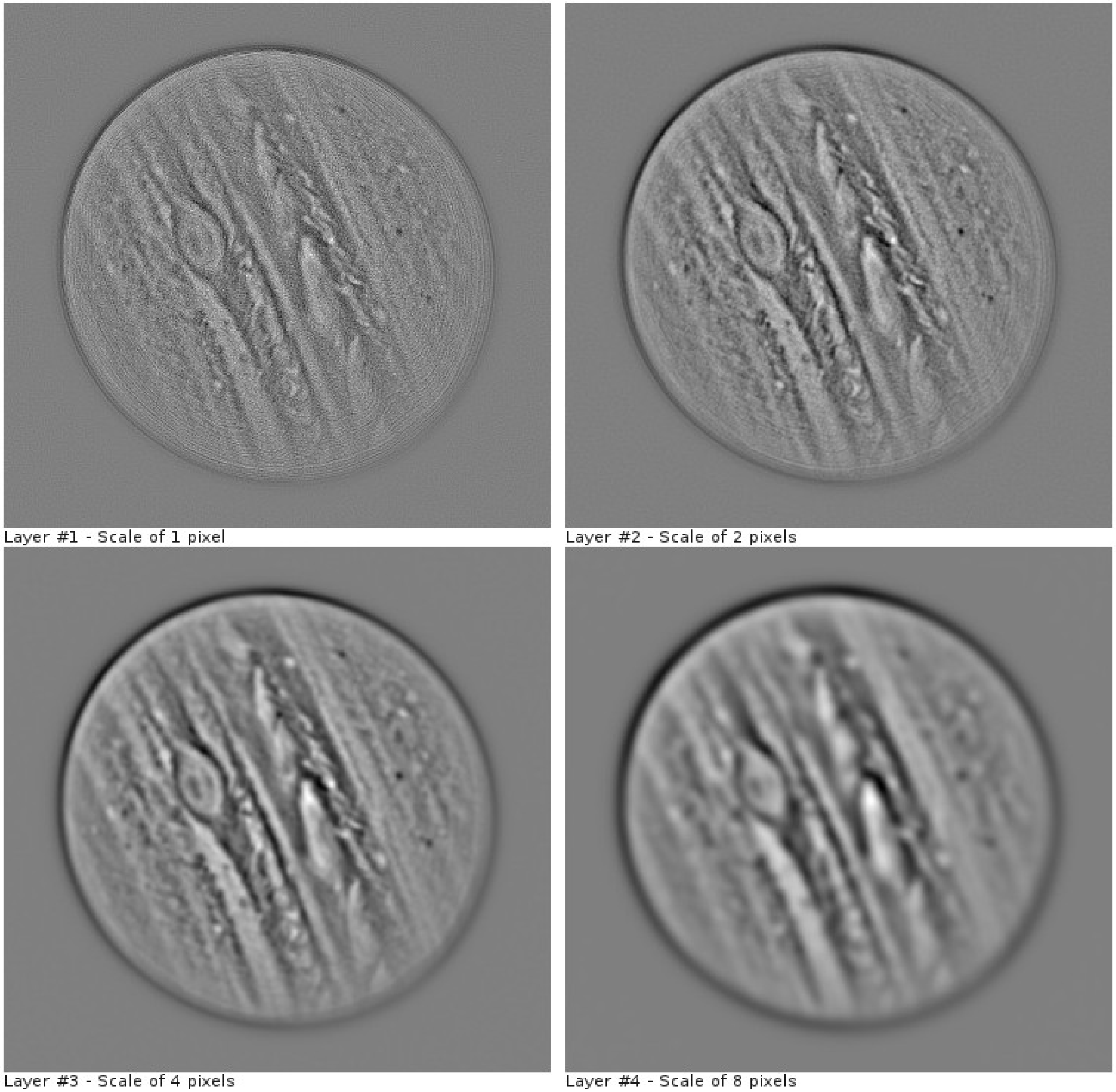
À Trous Wavelets

- ¿Qué pasa si agrandamos el filtro, en vez de reducir la imagen?
  - Mejora la visibilidad.
  - Correlación espacial más directa.
  - Pero, ocupa más espacio de almacenamiento, y el filtrado es más lento.
- Ya que estamos trabajando sobre imágenes ya suavizadas, secuencialmente, es posible hacer el filtrado con filtros agujereados, tomando el mismo tiempo.

À Trous Wavelets

$$h_{j+1} = \frac{1}{256}\left\{\begin{matrix} 1 & 0 & 4 & 0 & 6 & 0 & 4 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 16 & 0 & 24 & 0 & 16 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 0 & 24 & 0 & 36 & 0 & 24 & 0 & 6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 16 & 0 & 24 & 0 & 16 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 4 & 0 & 6 & 0 & 4 & 0 & 1 \end{matrix}\right\}$$

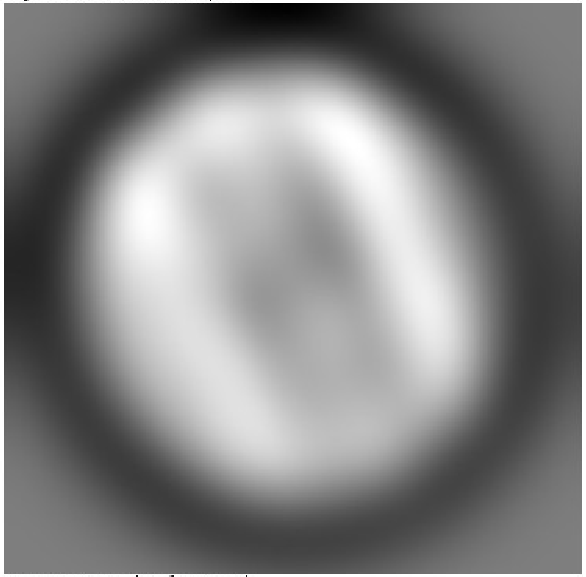
À Trous Wavelets - Ejemplo



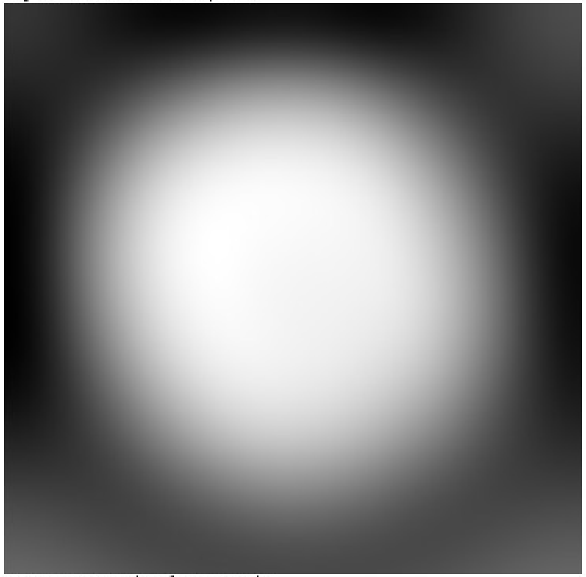
À Trous Wavelets - Ejemplo

Layer #5 - Scale of 16 pixels

Layer #6 - Scale of 32 pixels

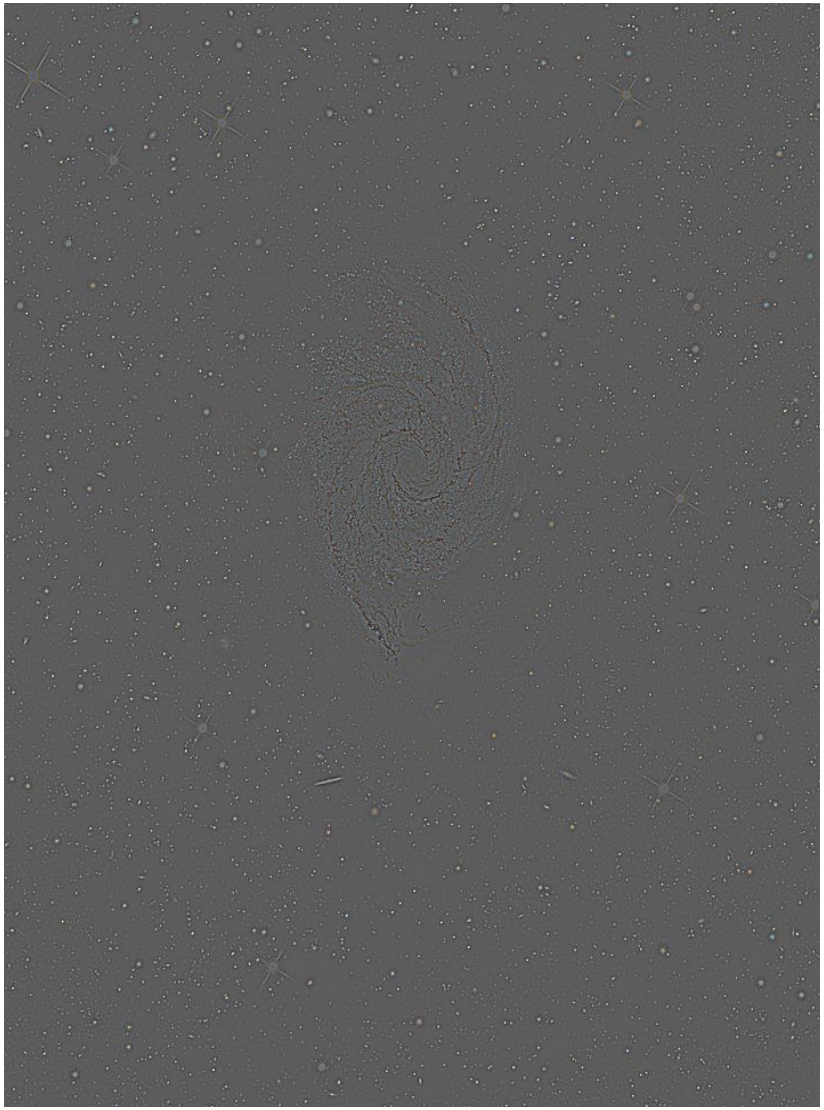


Layer #7 - Scale of 64 pixels



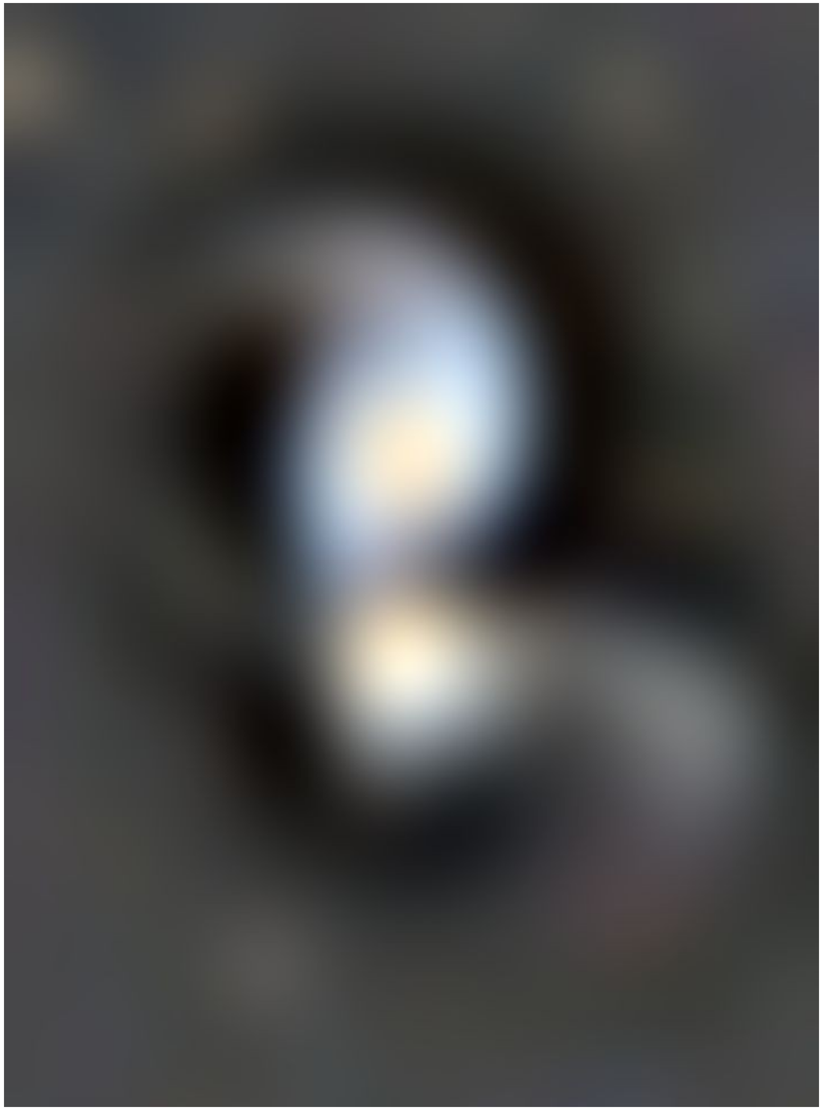
Layer #8 - Scale of 128 pixels

M51 – Escala 1 px



M51 – Escala 16 px

M51 – Escala 64 px



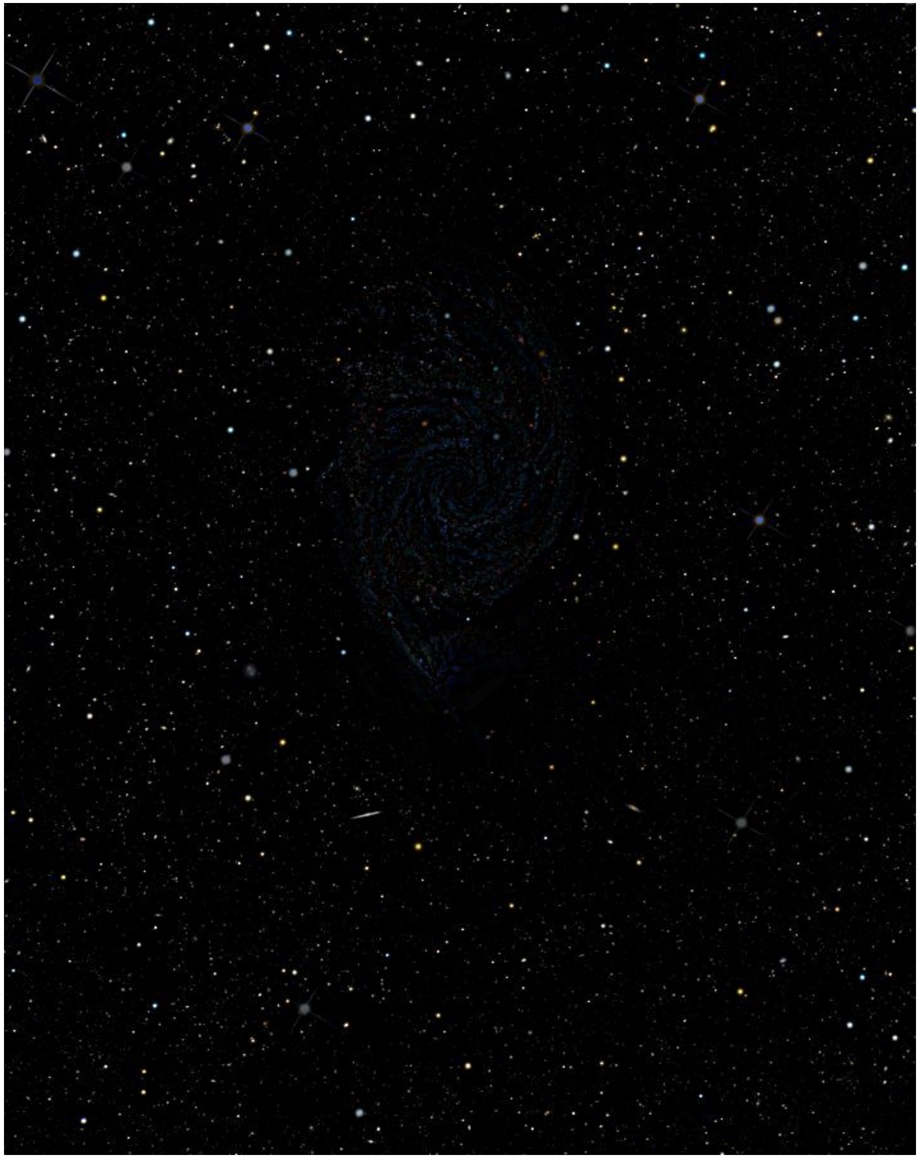
Reconstrucción de imagen original a partir de capas y remanente.

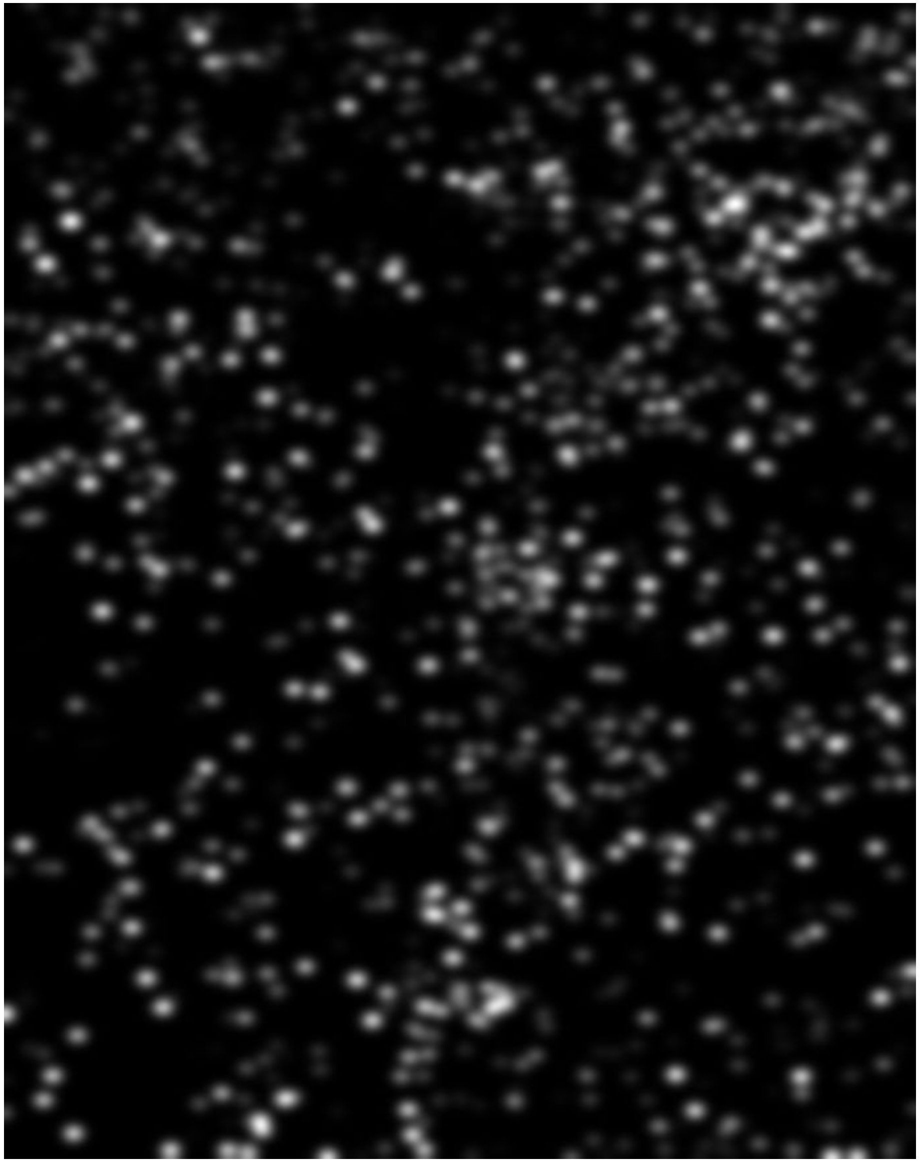
- Con el algoritmo À Trous, simplemente se suman las capas y el remanente.
- Se puede hacer la descomposición hasta el nivel que se quiera. El remanente guarda todas las escalas mayores a la última capa.
- Si la escala de la capa es mayor que la imagen, la información es nula.

Aplicaciones

- Segmentación o detección de estructuras.
    - Útil para fabricar máscaras
- Realce o suavizado de estructuras.
    - Reducción de ruido selectiva.
    - Incremento contraste local.
    - Efecto de enfoque muy controlado.
- Compresión de Alto Rango Dinámico (HDR).

Máscaras de estrellas

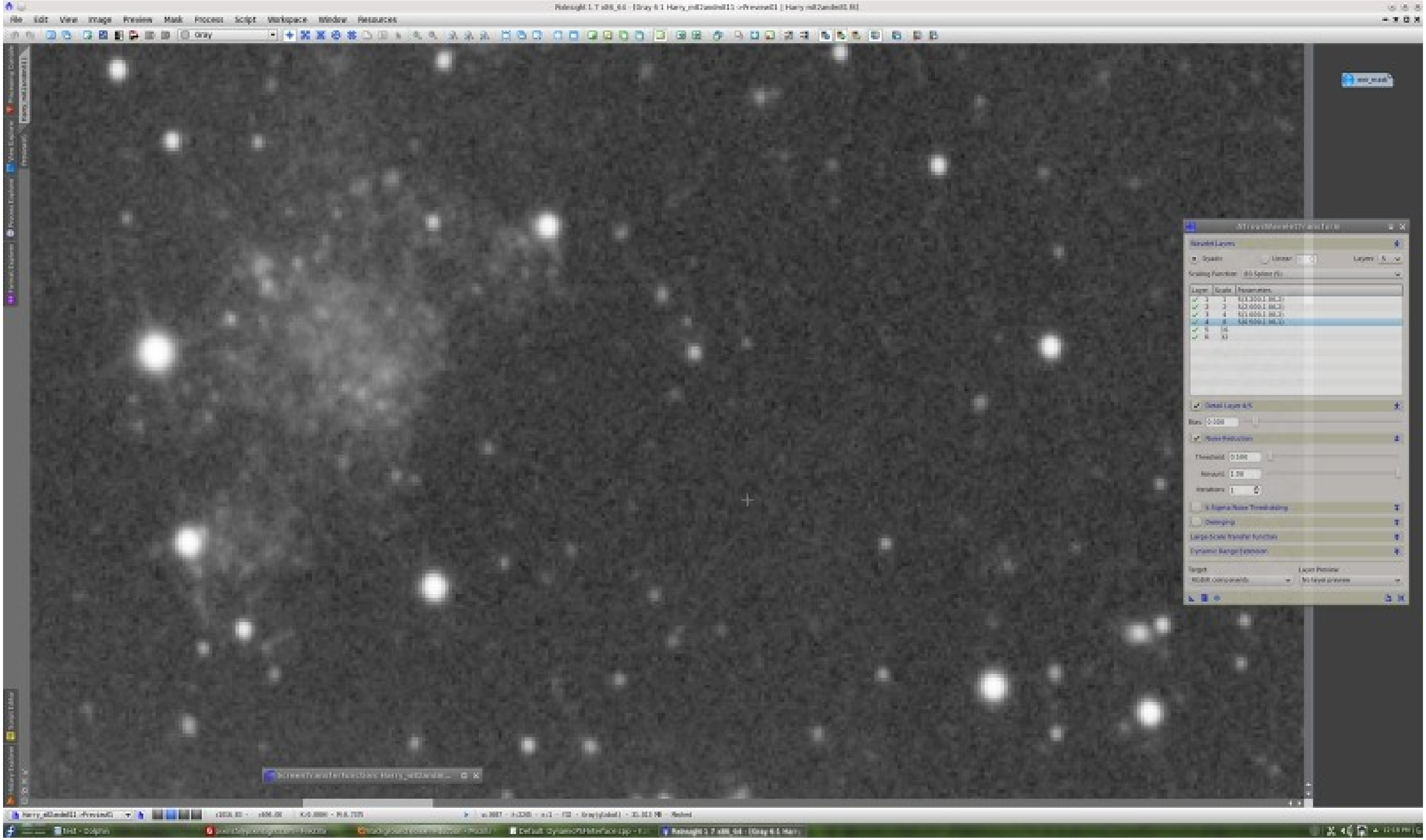




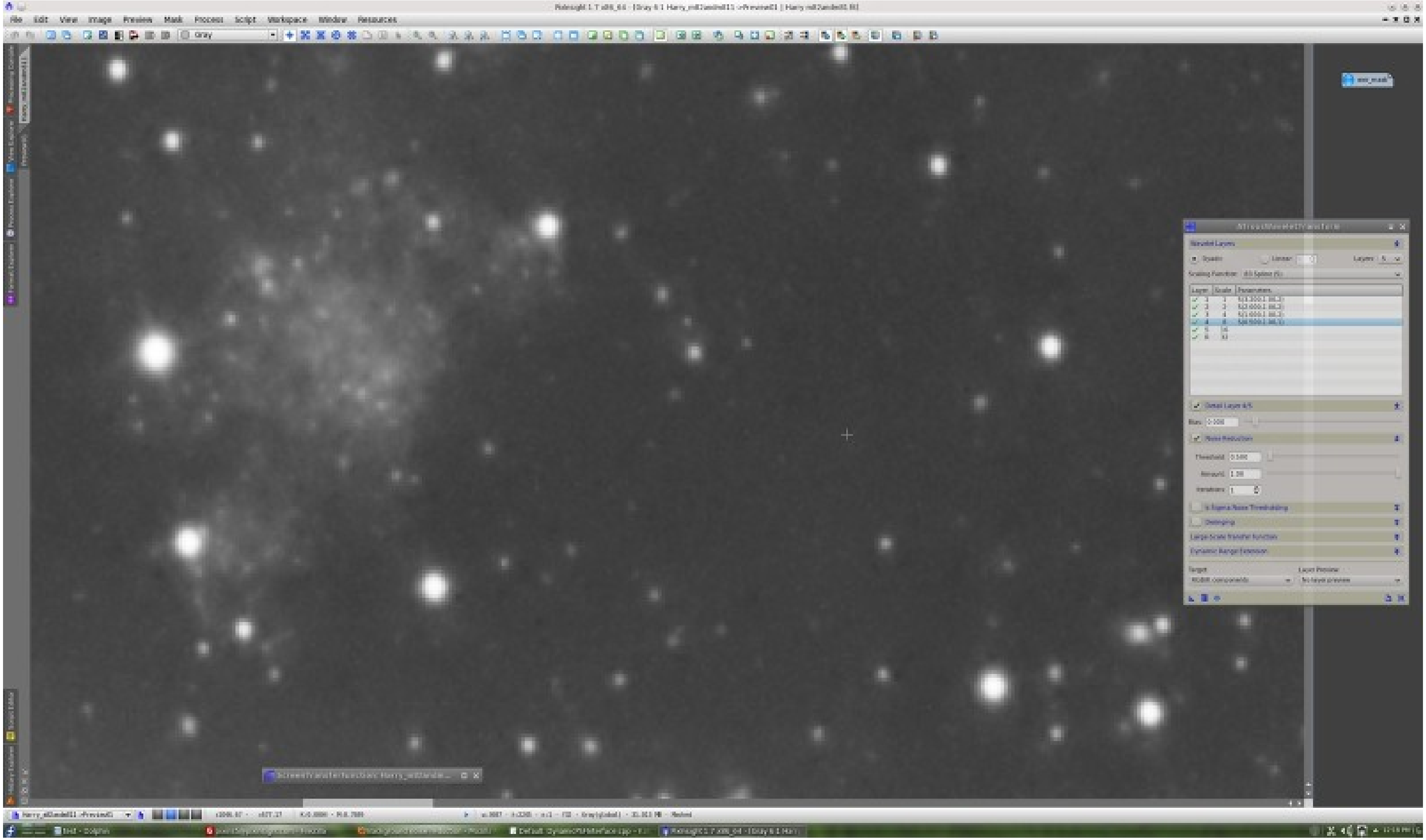
Reducción de ruido

Reducción de ruido

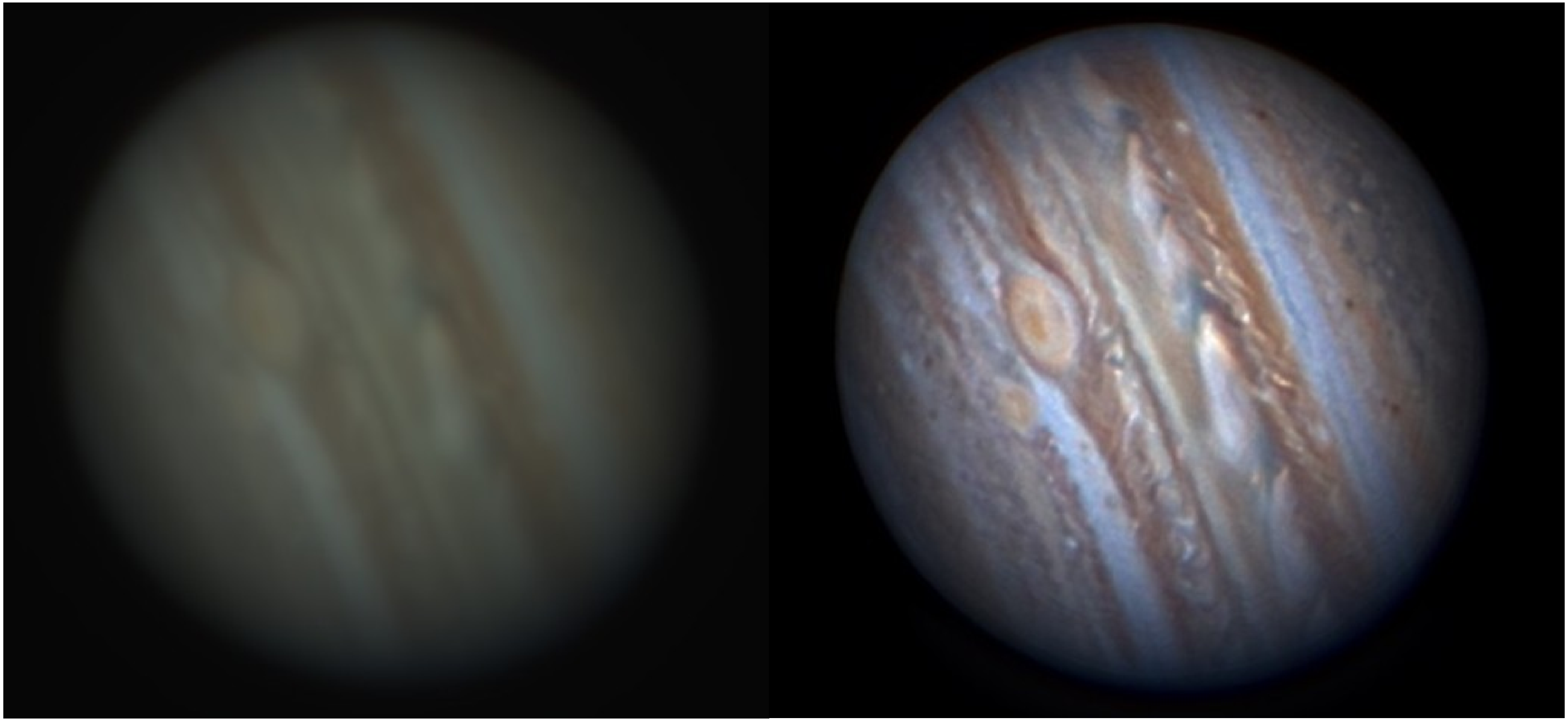
Reducción de ruido



Reducción de ruido



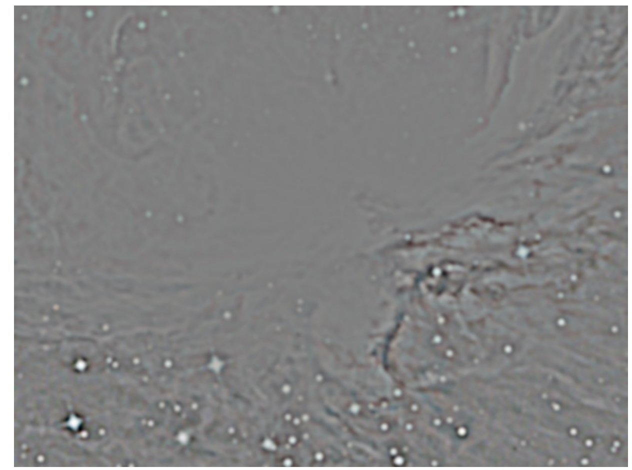
Realce Estructuras



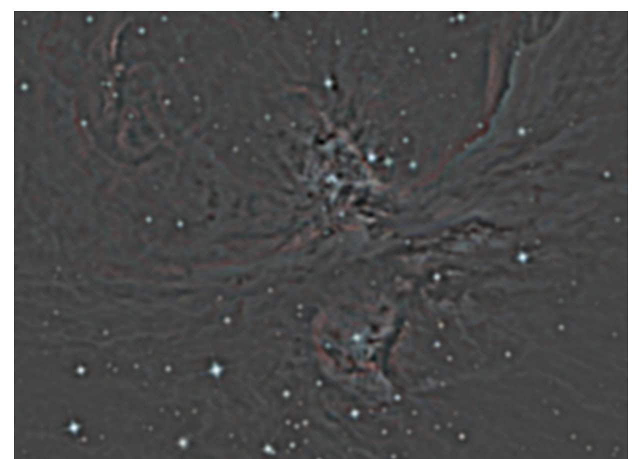
Con una función de escalamiento de alta resolución, se puede tener gran control sobre las escalas pequeñas.

Ejemplo HDR

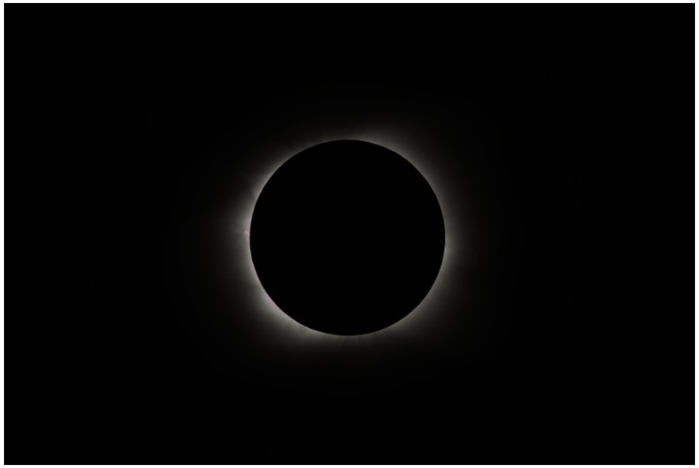
HDRWT

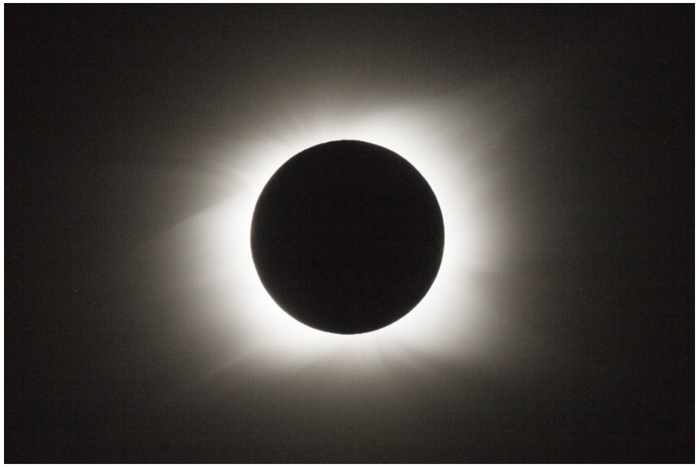


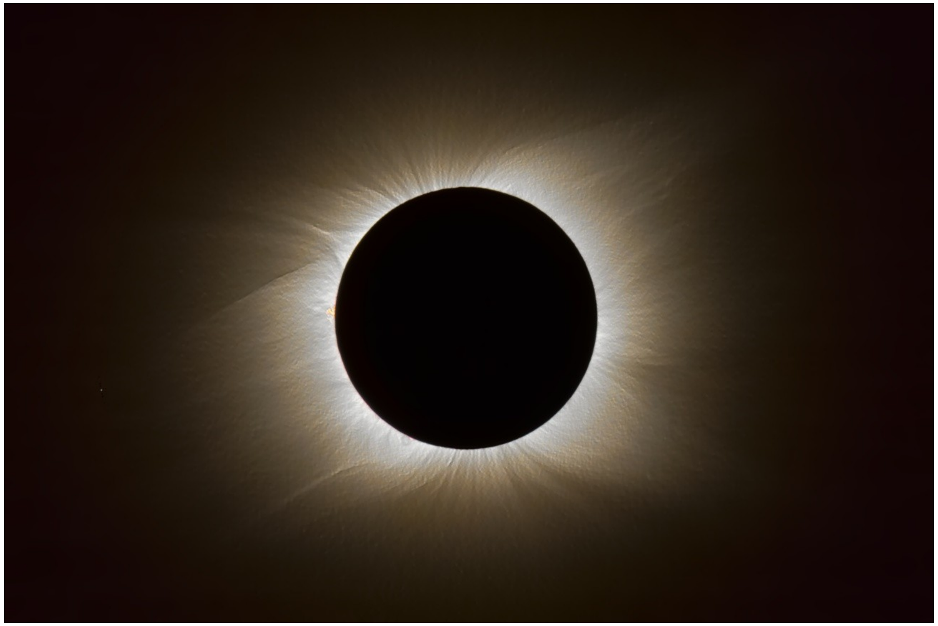
- Cada nueva capa reescala los datos usando información de escalas superiores.
- Contraste local se amplifica, en toda escala.



Otro ejemplo HDR

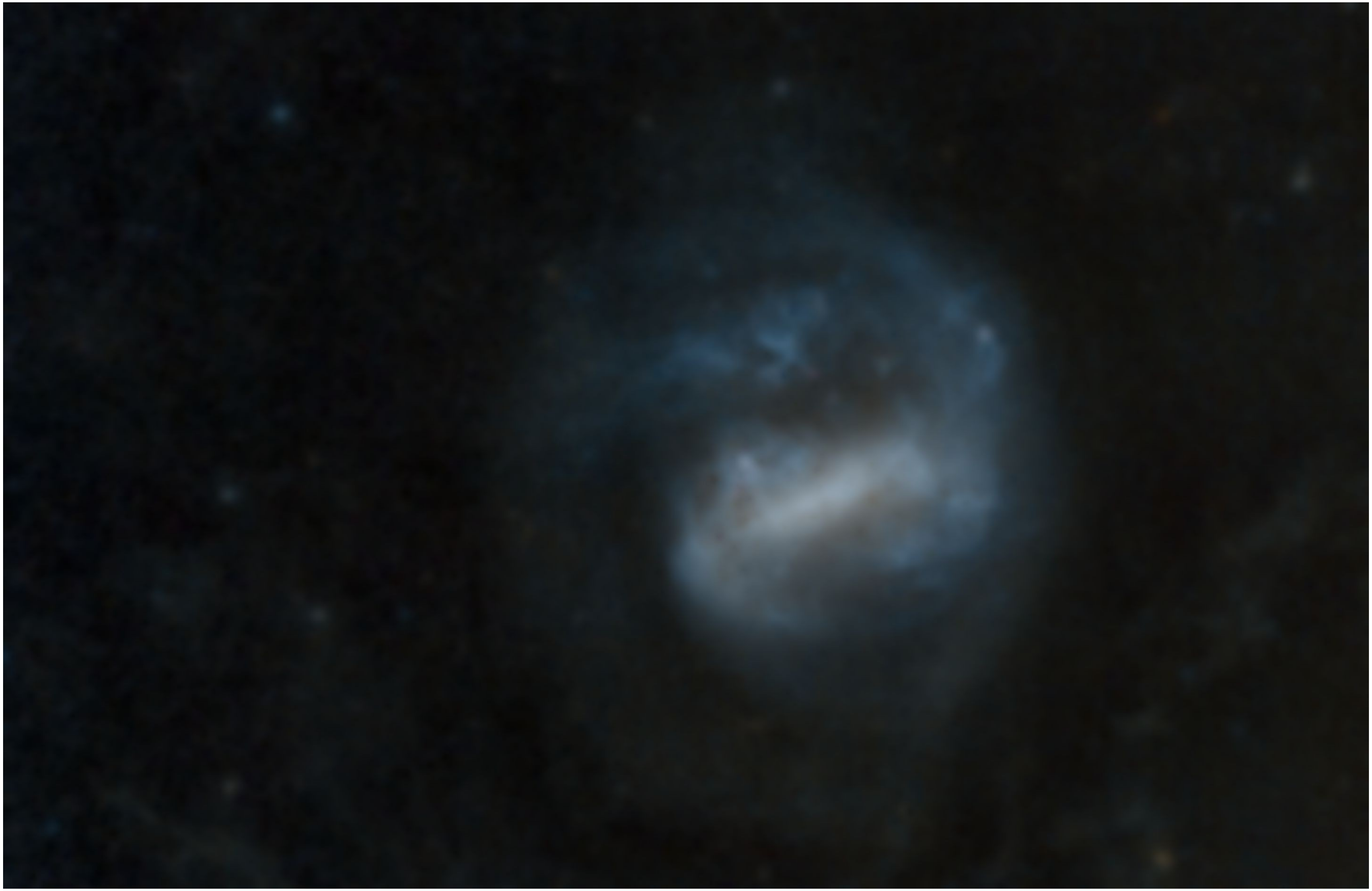






Separación de escalas con filtros morfológicos y wavelets

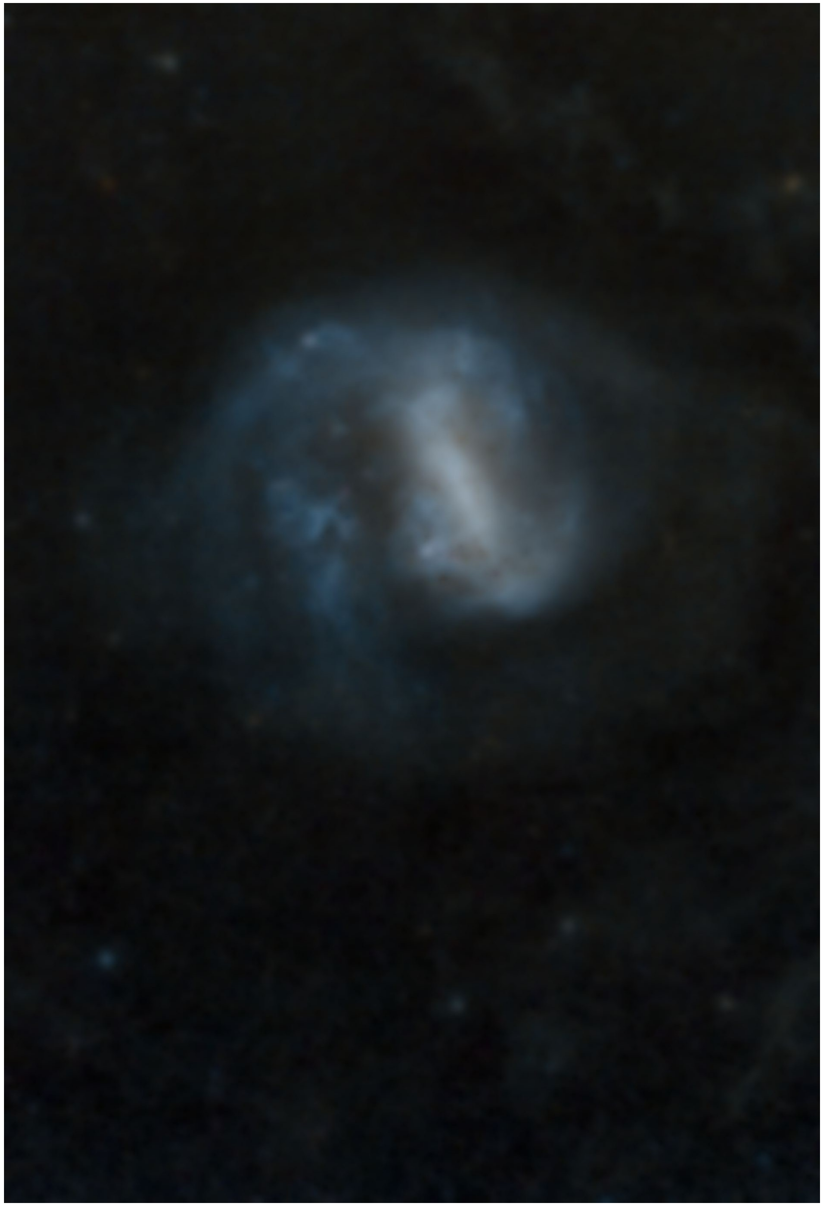
Separación de escalas con filtros morfológicos y wavelets

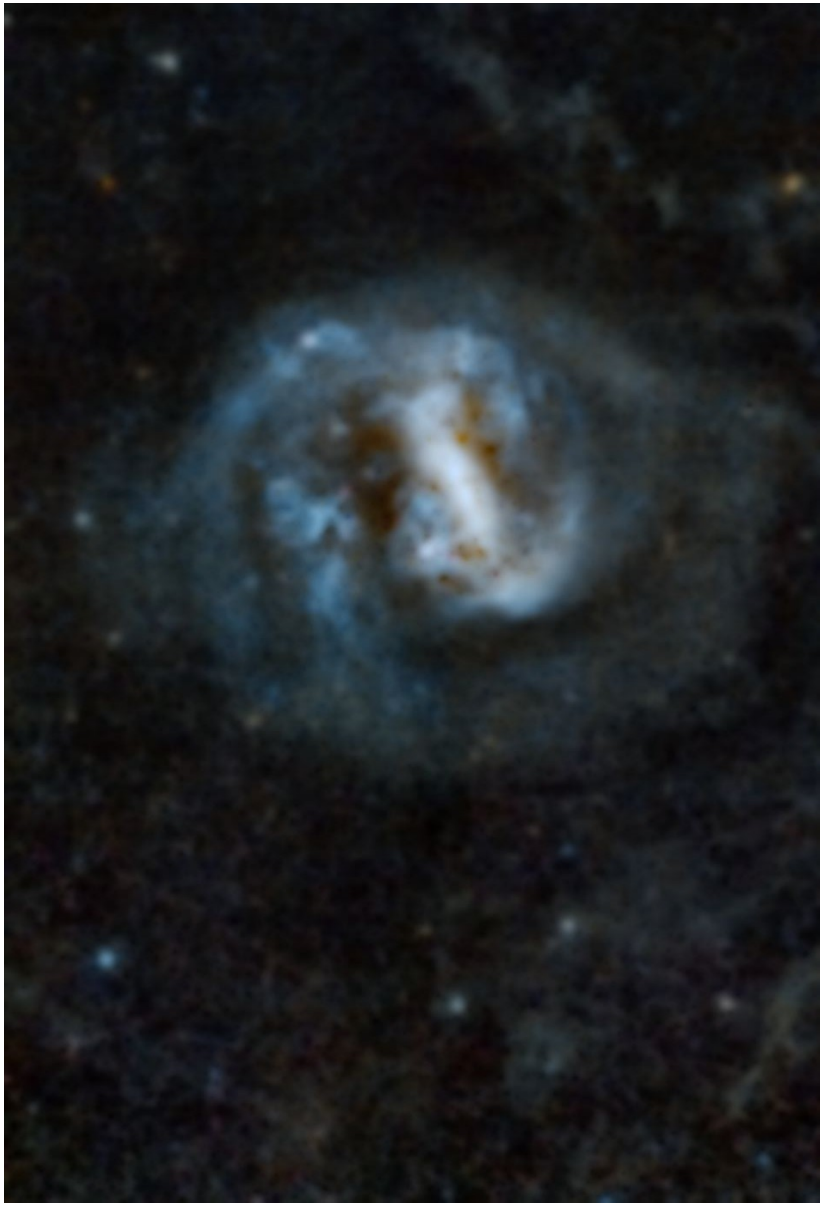


Separación de escalas

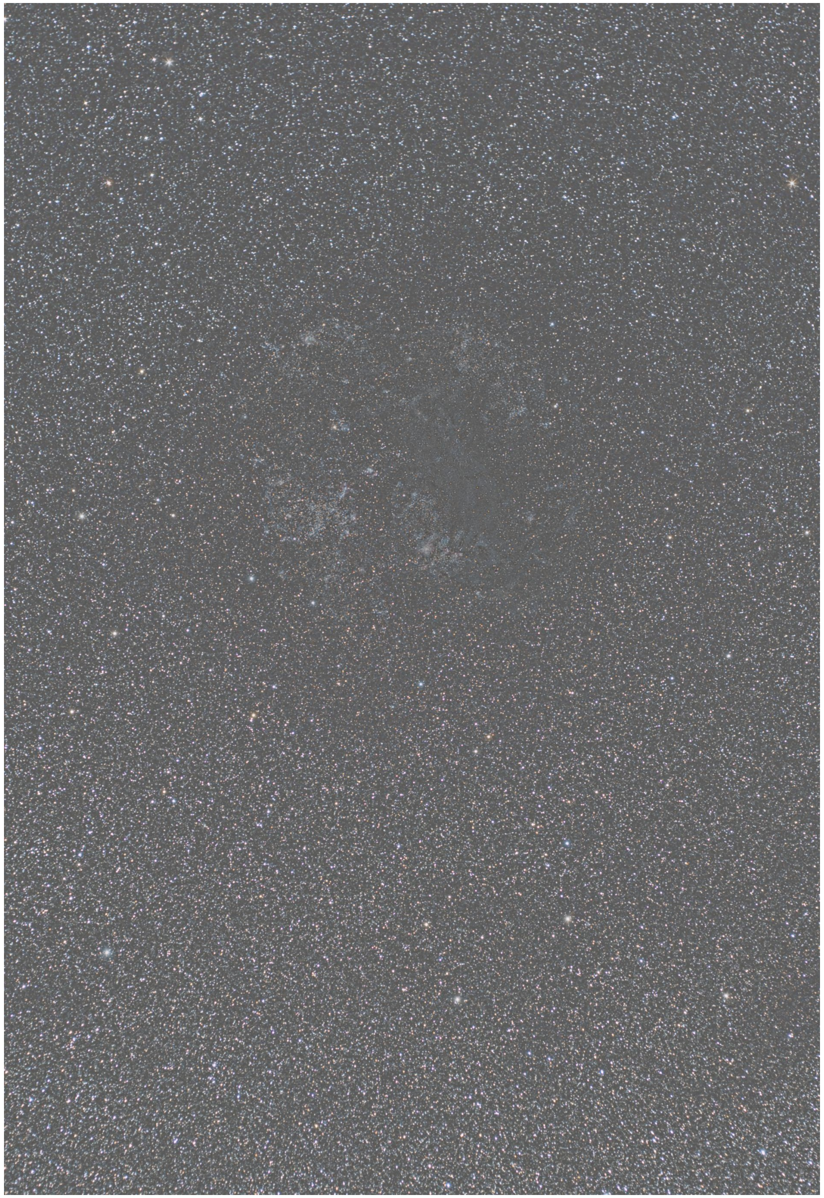
- Secuencias de filtros de erosión o selección (0.10 a 0.20) con formas de diamante y cuadrados. Borrar estrellas.
- Eliminar capas de wavelets hasta 4 u 8 px para suavizar la imagen.
- Resultado: Escalas grandes, sin efecto “fantasma” de objetos pequeños muy intensos (estrellas brillantes).
- Pequeña escala se obtiene tomando la diferencia con el original.

Procesamiento escalas separadas

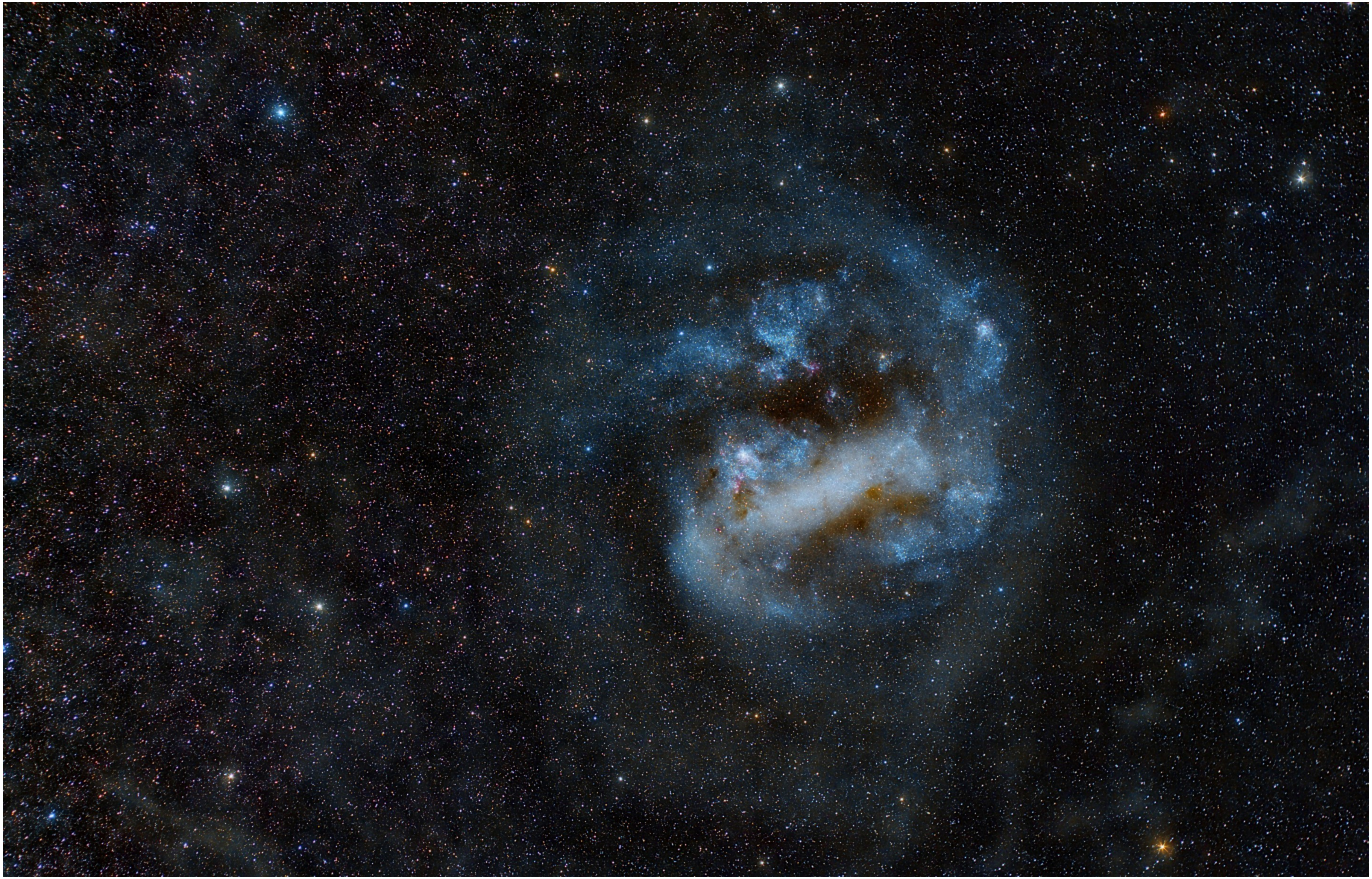




Procesamiento escalas separadas



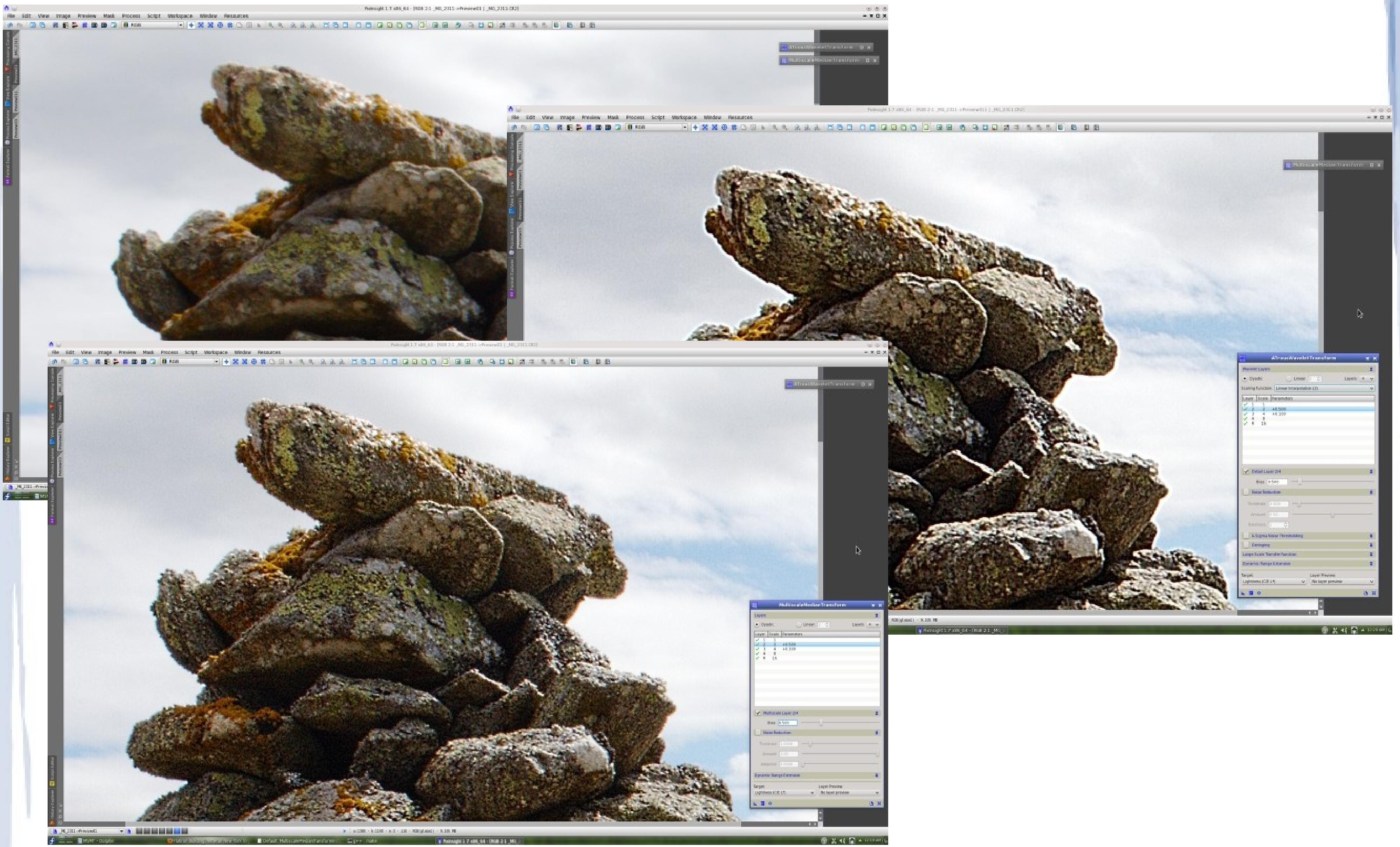
Sumamos escalas



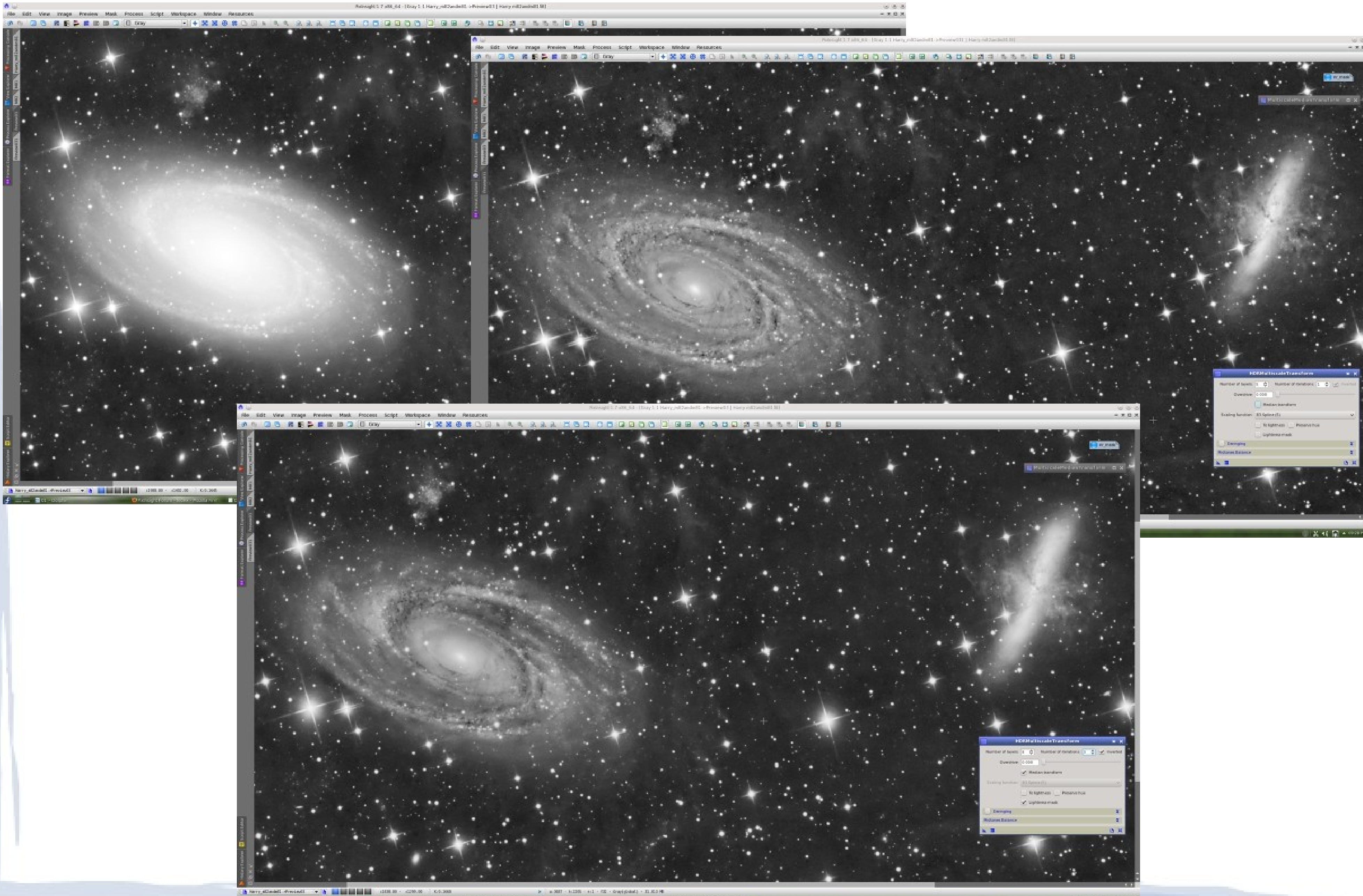
Trans. Multiescala con Mediana

- Siguiendo el mismo paradigma, usamos filtros estadísticos (mediana) como función de escalamiento.
- Ventaja: Significativa reducción de sombras o fantasmas en capas superiores. Menos artefactos de halos.

Ejemplo



HDR con Mediana



Fin

Créditos imágenes: Carlos Milovic, Vicent Peris, Harry Page, Mauricio Lopez,
Christopher Go, Juan Conejero.

Introducción basada en apuntes del curso “Fundamentos de Procesamiento de Imágenes”, Escuela de Ingeniería, Pontificia Universidad Católica de Chile.