

# Procesos basicos de imagen usando ImageJ

Manuel Graña

Master Ingeniería Biomédica

UPV/EHU

# Inicio

- Download ImageJ 1.44
  - <http://rsbweb.nih.gov/ij/download.html>
- La instalación es inmediata puesto que es una aplicación Java
- Requerimientos sistema operativo: todos los sistemas

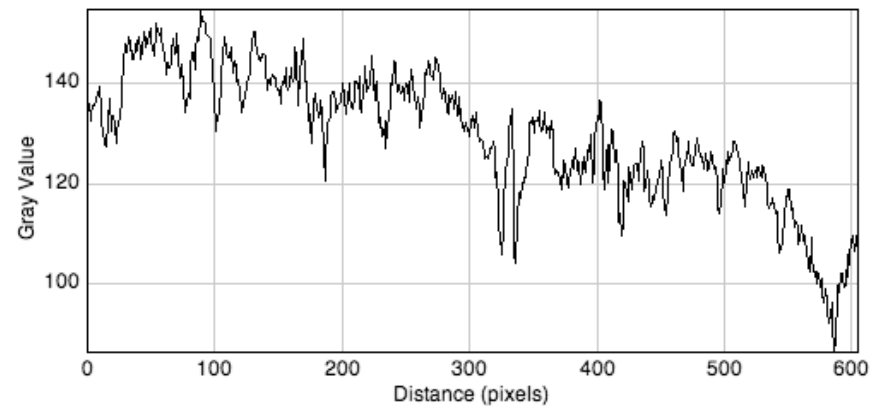
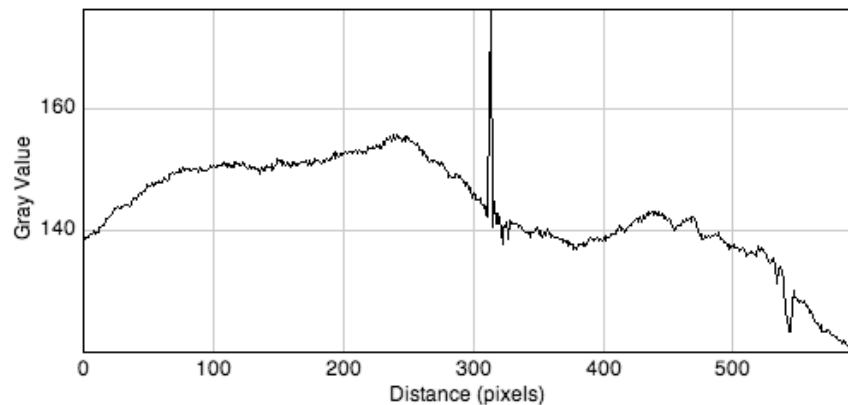
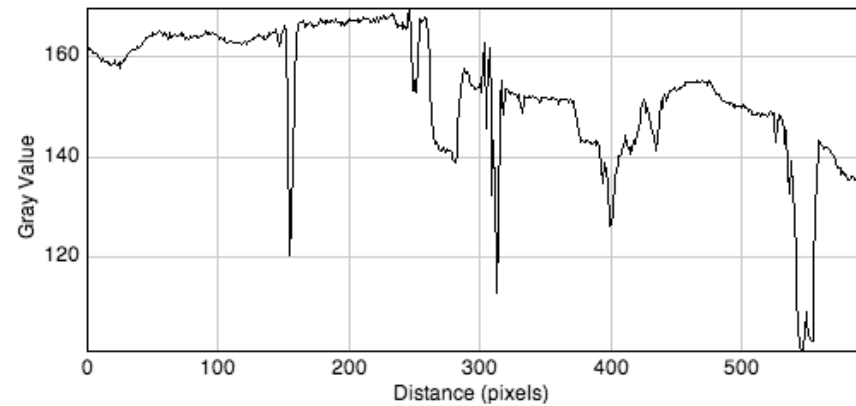
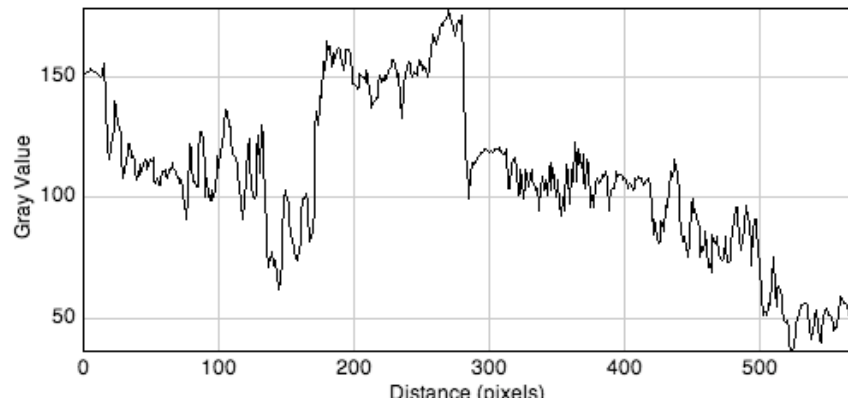
# Plan de trabajo

- Operaciones sobre las imágenes muestra que proporciona la instalación básica
- Para cada caso indicaré
  - La imagen utilizada
  - El tipo de proceso sin especificar los detalles, el alumno debe navegar en los menus para encontrar la operación apropiada
  - La imagen resultado que el alumno debe reproducir
- El alumno debe construir un powerpoint similar a este con
  - El resultado obtenido en cada caso
  - El detalle de las operaciones que ha empleado, indicando parámetros de las operaciones/filtros

# Profiles

- Las filas de las imagenes son funciones que pueden visualizarse como plots, valores altos corresponden con colores claros en la imagen
- Podemos usar la herramienta de seleccion rectangular para seleccionar una fila de la imagen y
- Plotear el profile

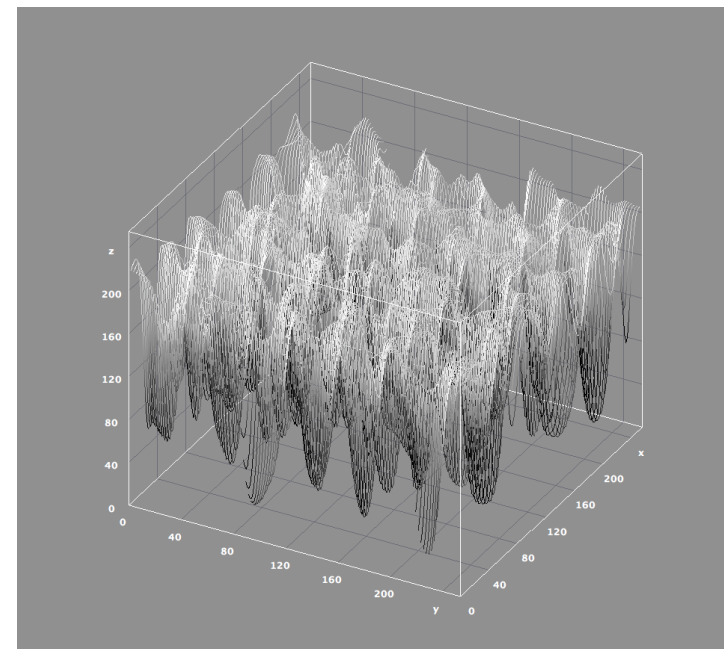
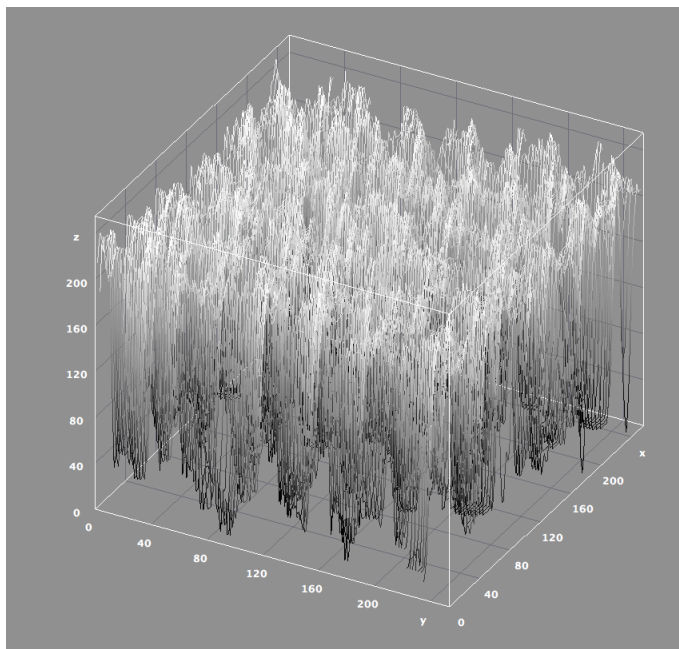
- Imagen de prueba: boats.gif
- Resultados a reproducir

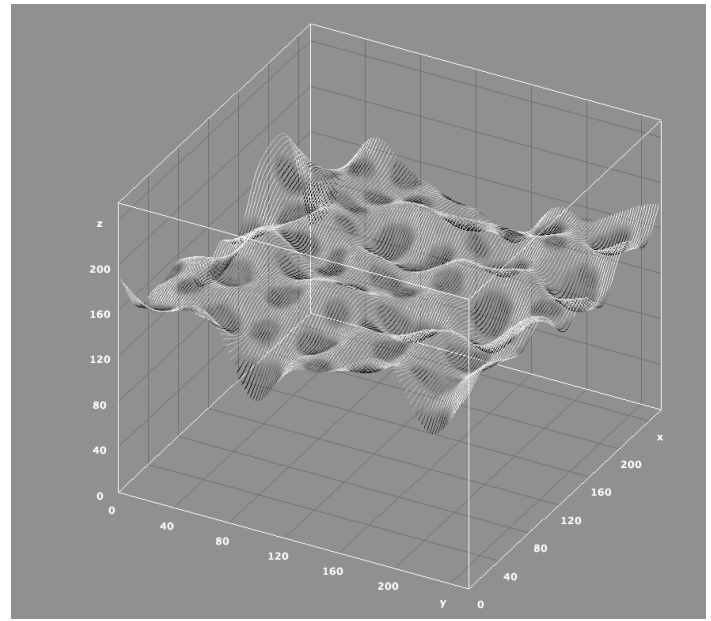
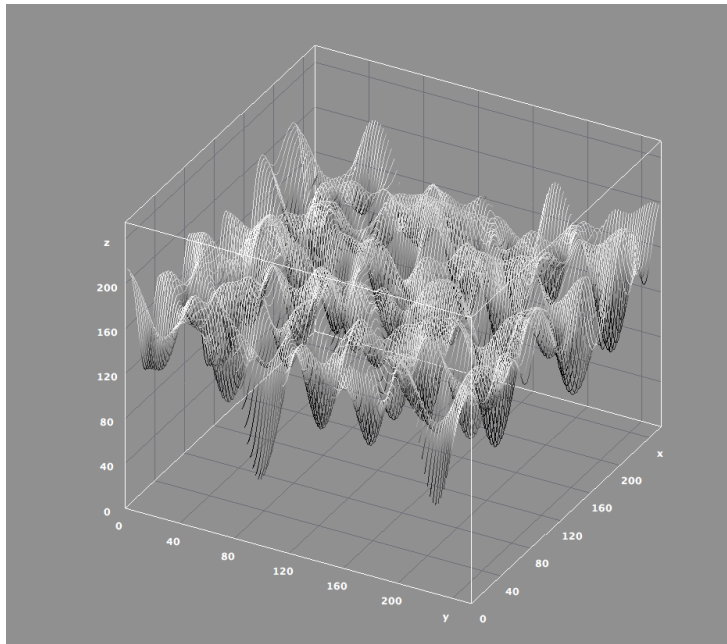


# Imágenes como superficies 3D

- Las imágenes son superficies en un espacio 3D, equivalentes a una función escalar con rango 2D
- Podemos visualizar esta estructura mediante el plugin correspondiente
- la suavización tiene el efecto de convertir la superficie en una banda con un rango de valores reducido y cambios suaves

- Imagen de prueba: blobs.gif
- Resultados esperados: detalles de color y fondo pueden cambiar entre sistemas operativos



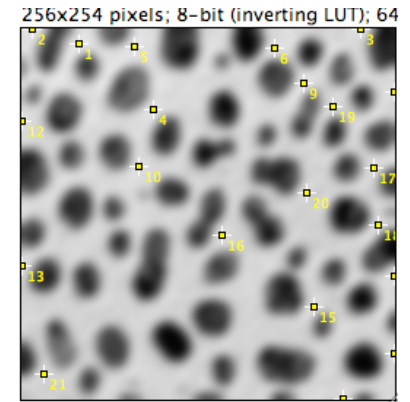
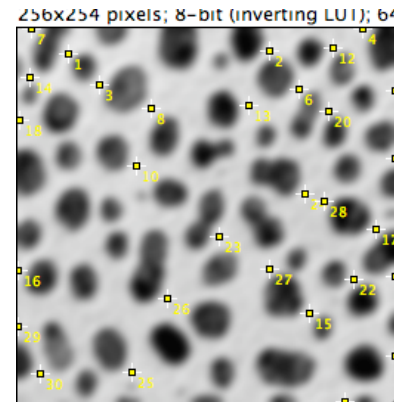
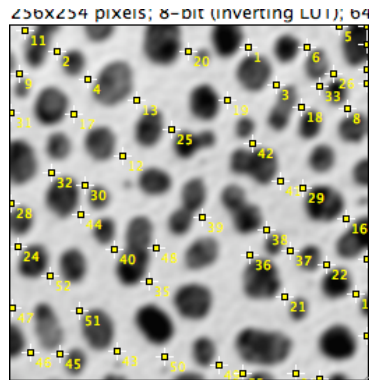
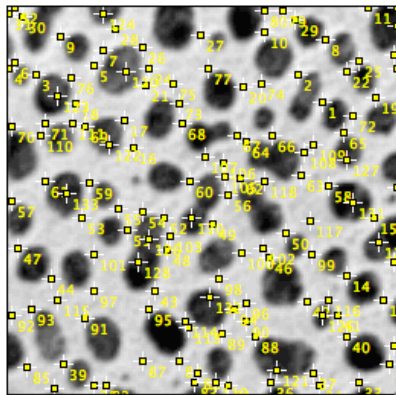




# Suavizacion

- El proceso visual de suavización (smoothing) produce imagenes cada vez mas difuminadas (blurred).
- Iterando el proceso de smoothing se puede reproducir visualmente los resultados anteriores
- Para observar que efectivamente se está produciendo un “alisamiento” podemos visualizar los máximos locales detectados en la imagen

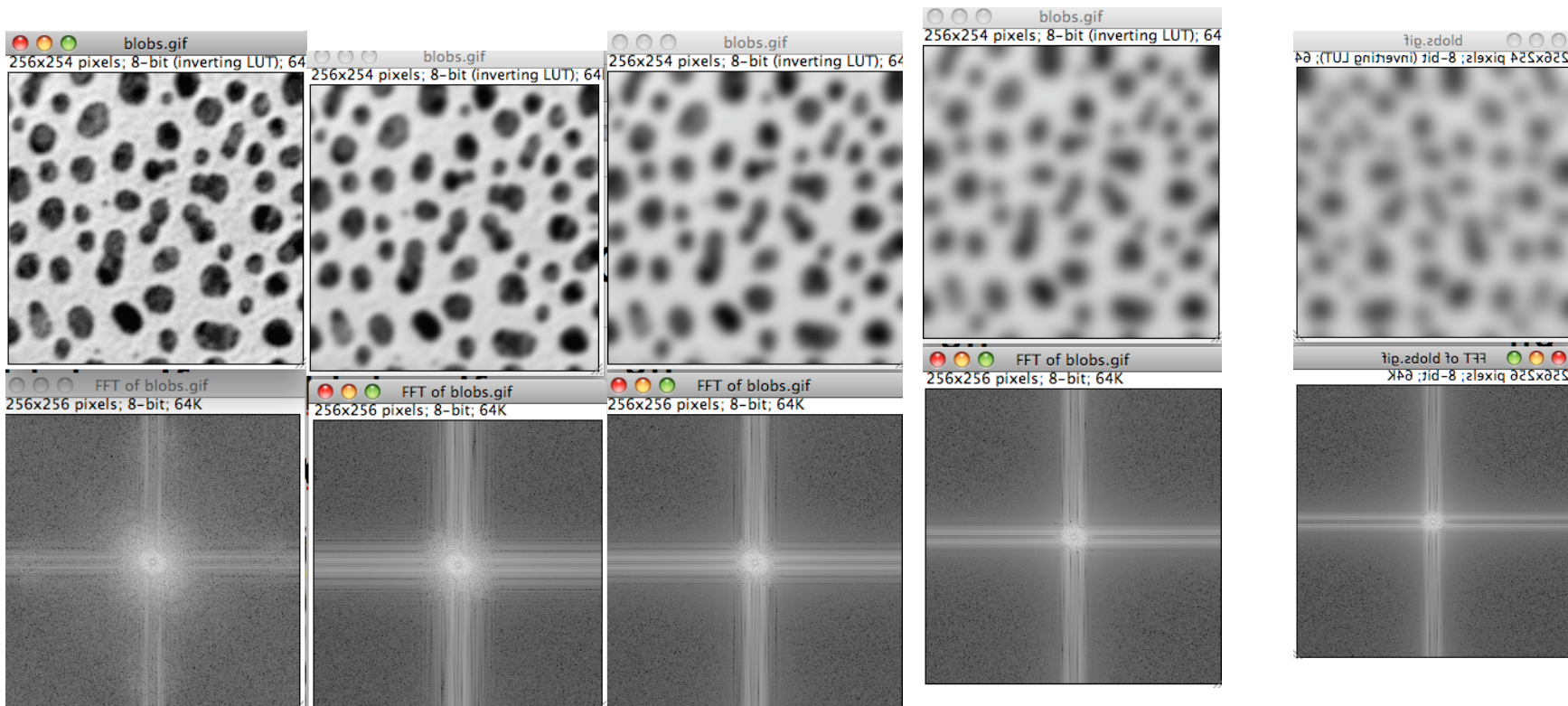
- Imagen: blobs.gif
- Resultados esperados de la iteracion



# La transformada de Fourier

- La transformada de Fourier nos da información de la composición de la imagen como una mezcla de ondas sinusoidales de diversas frecuencias espaciales.
- La Fast Fourier Transform (FFT) es una implementación altamente eficiente.
- Podemos visualizar el efecto de la suavización en el espacio transformado
- Podemos aplicar un filtro de difuminación gaussiano con radios crecientes para obtener suavizaciones progresivamente más fuertes

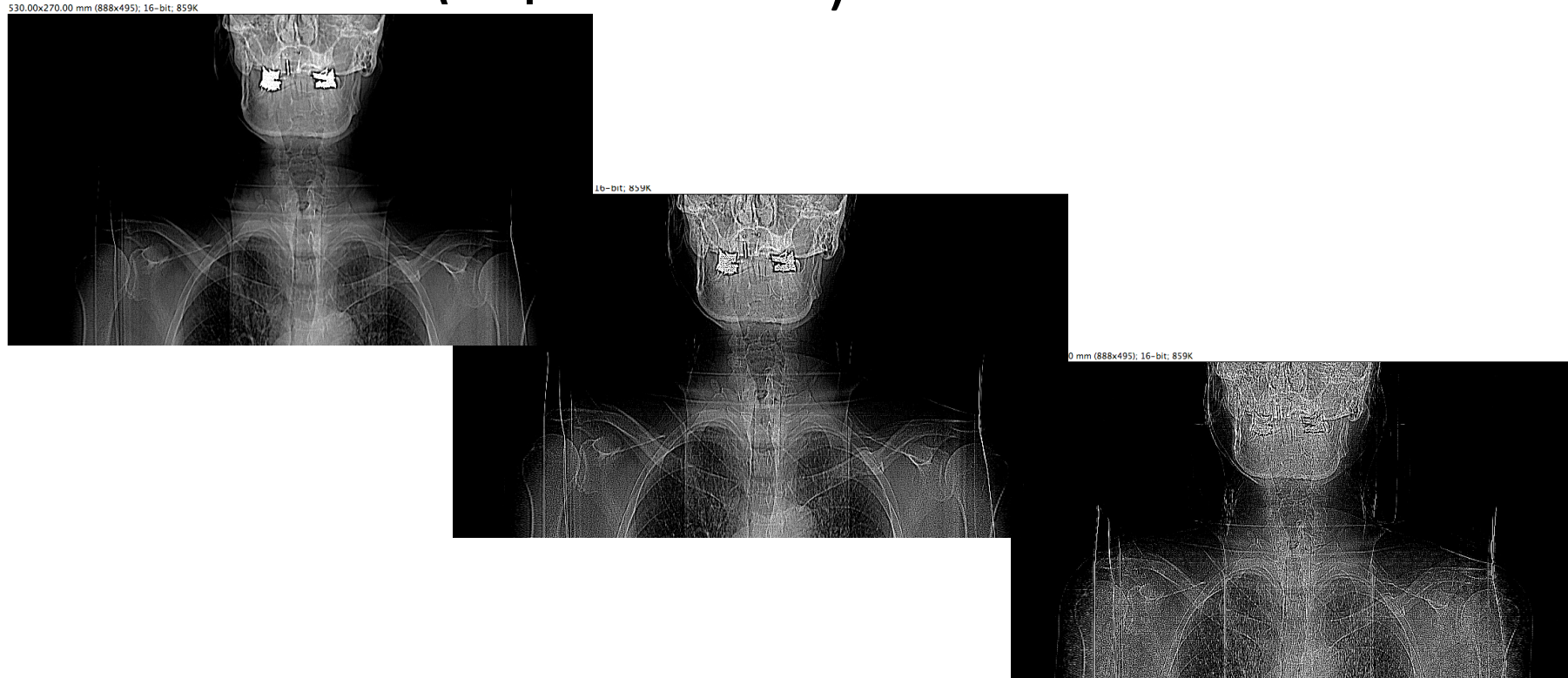
- Imagen: blobs.gif
- Resultados esperados



# Afilado

- La operación dual a la suavización es el afilado, (sharpening) o filtrado pasa alta.
- Esta operación resalta las altas frecuencias con el peligro de resaltar el ruido sobre la señal

- Imagen : CT.dcm
- Resultados (izqda a dcha)

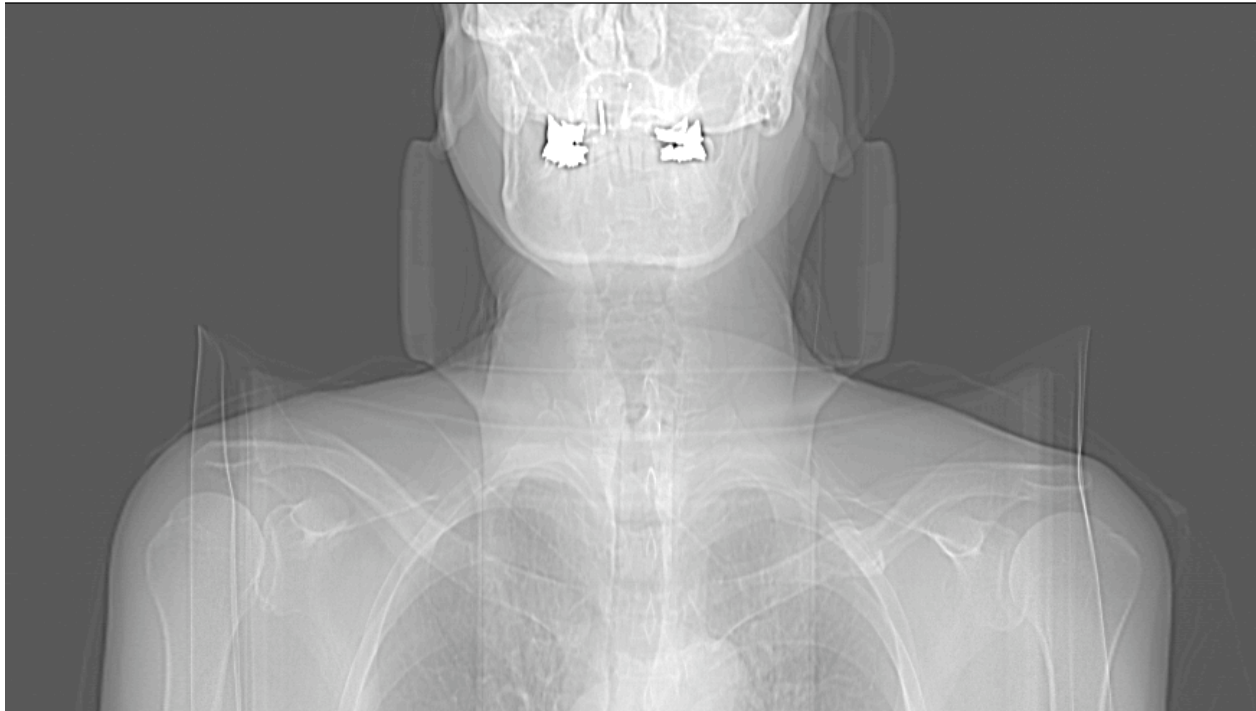


# Transformaciones de intensidad

- El rango de valores de la imagen puede abarcar muchos mas niveles de gris o colores de los que podemos visualizar, por lo que transformaciones de visualizacion pueden ofrecernos estructuras que no son visibles inmediatamente...( y pueden ser irrelevantes desde el punto de vista médico)

- Imagen : CT.dcm
- Resultados del ajuste de brillo y contraste

530.00x270.00 mm (888x495); 16-bit; 859K

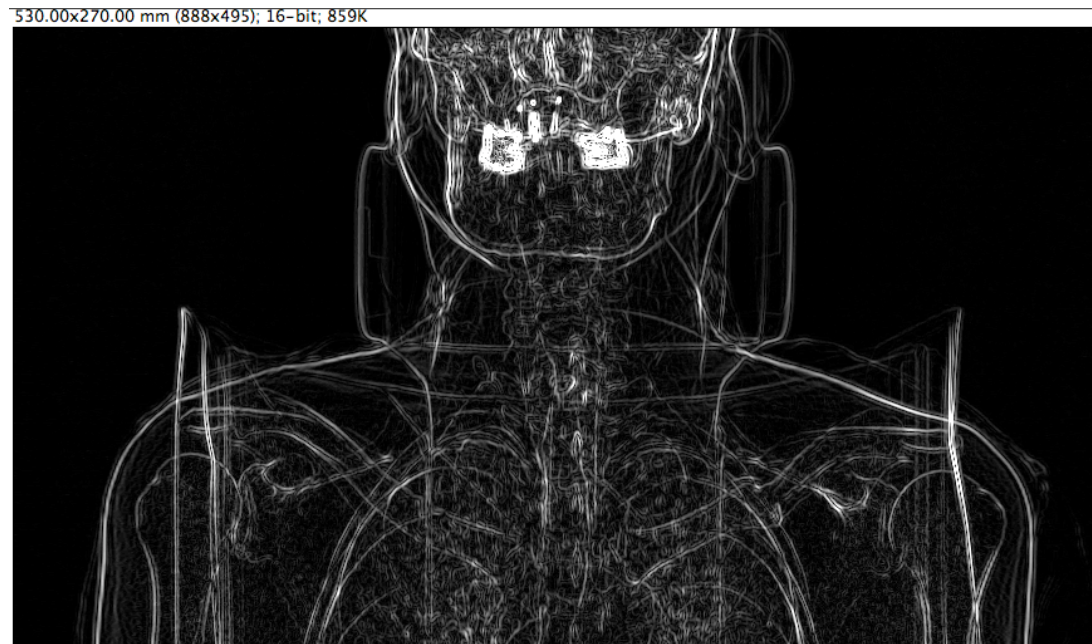




# Bordes en la imagen

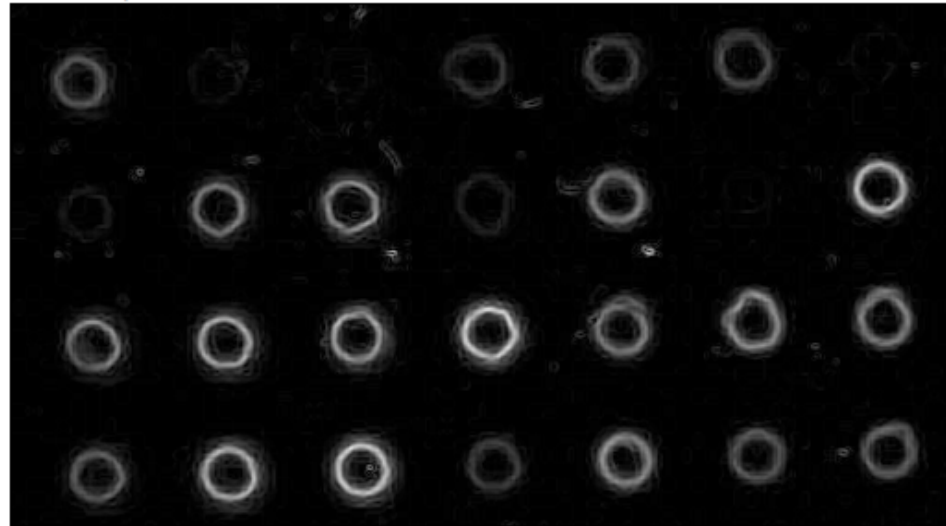
- Los bordes (edges) son las fronteras entre regiones de la imagen u objetos que podemos distinguir.
- Cuando procesamos la imagen para obtener los bordes, dependiendo de la implementación, los resultados pueden ser “negros” pero un pequeño ajuste de brillo y contraste puede sacar a la luz lo que está escondido.....

- Imagen CT.dcm
- Resultado de buscar los bordes y ajustar....



- Otras veces no hace falta hacer cosas raras...  
imagen Dot\_blot.jpg

513x282 pixels; 8-bit; 141K

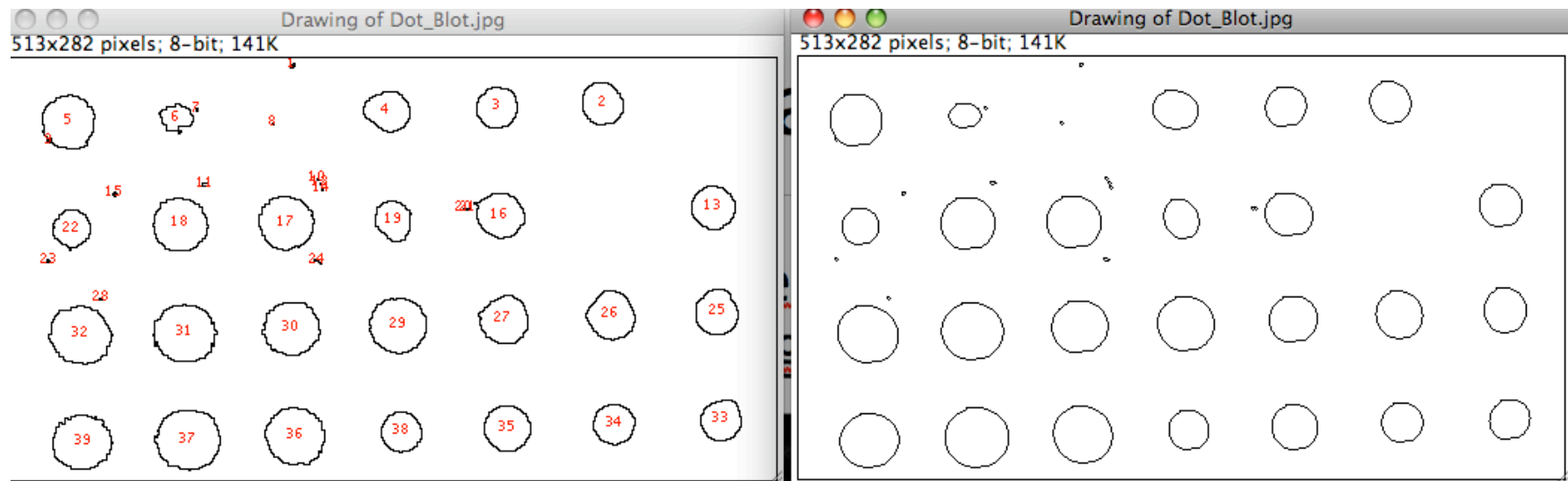


- Atención a los efectos indeseados, ¡¡¡son resultado de la compresión con pérdida de jpg!!!

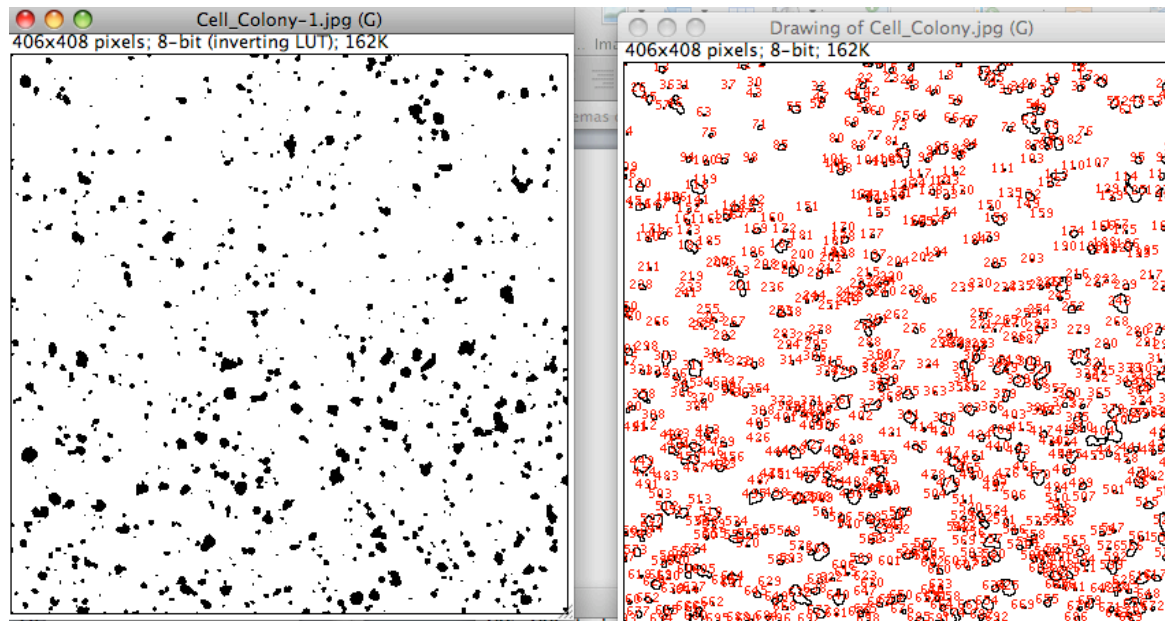
# Buscando cosas

- A veces queremos detectar las cosas que estan en la imagen, para concentrarnos en ellas.
- Analizamos particulas y encontramos donde estan colocadas.... En ImageJ requiere que la imagen sea binarizada aplicando un umbral (threshold)

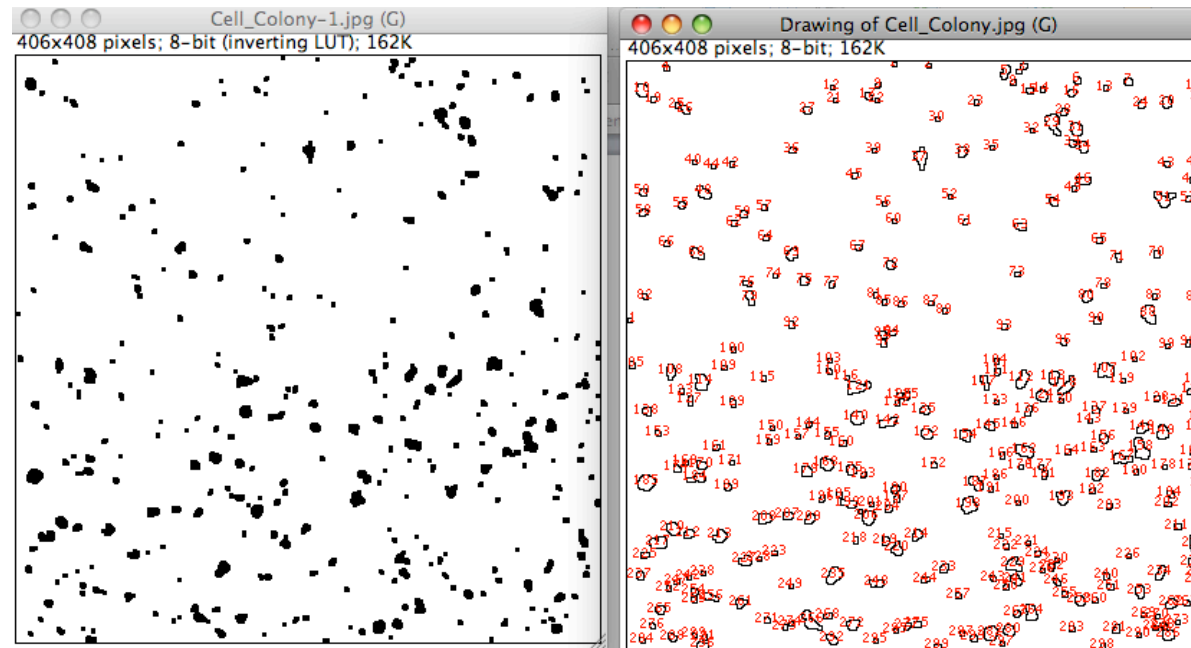
- Imagen: Dot\_blot.jpg
- Resultados ...



- Supongamos que queremos contar las células en Cell\_colony.jpg, asumiendo que los blobs pequeños son ruido



- Podemos simplificar la imagen haciendo apertura morfológica (open)



- Para obtener filtros mas fuertes podemos repetir un par de erosiones seguidas de un par de dilataciones, lo que equivale a la apertura con un filtro dos veces mas grande (limitaciones de ImageJ)

