

Taller 3: Filtering Methods on MRI.

Jesús Ernesto Suárez Triana

jesusesuarez@javeriana.edu.co

Oscar Manuel Beltran Camacho

beltranom@javeriana.edu.co

1) **Entendimiento de la lógica de los algoritmos** analizando el artículo Filtering Methods on MRI, antes de proceder a construir los algoritmos de las respectivas metodologías procedimos a realizar un entendimiento de los filtros a trabajar y concluimos los siguiente:

- El filtro de mediana del artículo, tiene una particularidad y es que no incluye el centro dentro de la frontera seleccionada.
- El filtro de mediana adaptativa, considera un crecimiento de frontera en los accesos a nivel 1, por ello, la lógica que construyamos para la selección de la región debe estar parametrizada para un bando de ancho r , donde r es el radio de la frontera que queramos seleccionar, por defecto 1.
- El algoritmo de filtro de mediana adaptativa es un algoritmo recursivo.
- El algoritmo de Weiner lo trabajaremos vía estandarización, acorde a la explicación y se debe tener una parametrización de los rectángulos a trabajar (donde queda parametrizado el número de filas N y columnas M que se obtendrán para cada filtro).

A continuación anexamos las imágenes correspondientes al entendimiento visual de los algoritmos.

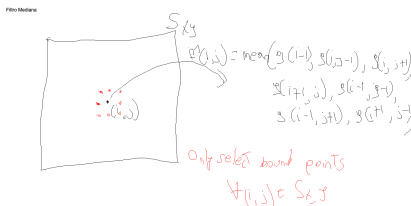


Figure 1: Filtro de mediana

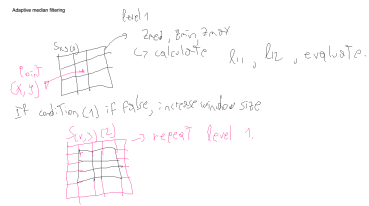


Figure 2: Filtro mediana adaptativo

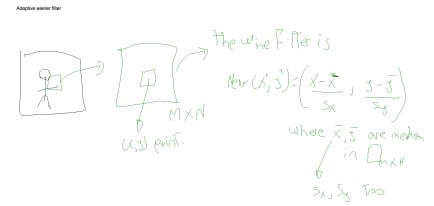


Figure 3: Filtro Weiner adaptativo

2) **Visualización imágenes originales.**

Para el desarrollo del ejercicio, realizamos la extracción de la información de la fuente brainweb la cual extrae genera simulaciones de RMI bajo características específicas. En el artículo original realizan una implementación

de ruido manual vía proceso Guasiano, pero en nuestro caso realizamos la variación de ruido a través de esta herramienta, considerando niveles de ruido de 0%, 1%, 3%, 5%, 7% y 9%, el ruido es calculado a través de la relación con el tejido más brillante.

En el caso del informe consideraremos variantes de ruido, pero el script funcional aplica para cada versión de ruido similar. a continuación visualiza

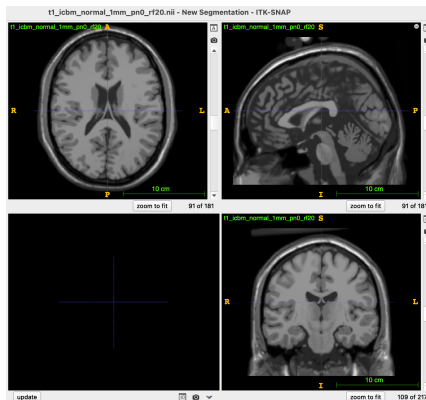


Figure 4: Imagen original ruido 0%

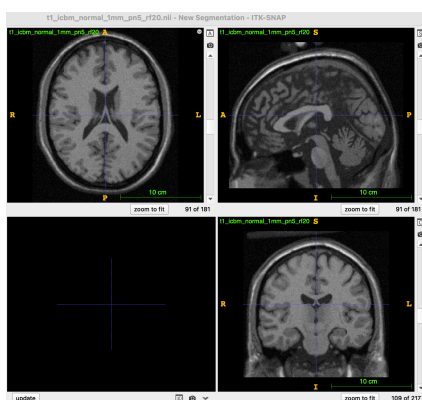


Figure 5: Imagen original ruido 5%

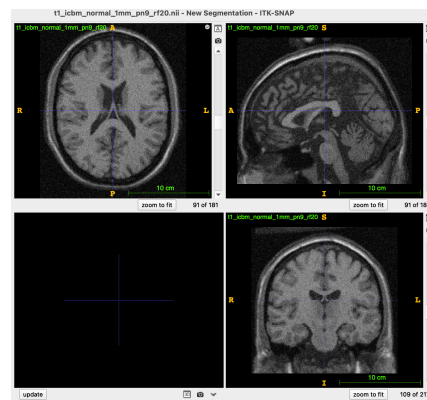


Figure 6: Imagen original ruido 9%

3) Filtro de Mediana:

Aplicando el algoritmo de mediana sobre las imagenes presentadas, logramos observar lo siguiente:

- Sobre la imagen de ruido 0% no observamos cambios significativos sobre la imagen.
- La imagen retornada por el algoritmo desde la parte frontal hay una inversión, lo cual puede dar dudas en su visualización.
- a medida que el ruido ingrementa, si vemos un cambio significativo el cual es un leve suavisamiento en la imagen proyectada. En particular los pixeles sobre las imagenes son más suaves y uniformes despues de aplicar el filtro.
- Entre el volumen de ruido sea más amplio, vemos que el filtro de mediana se comporta mejor.



Figure 7: Imagen filtro mediana ruido 0%

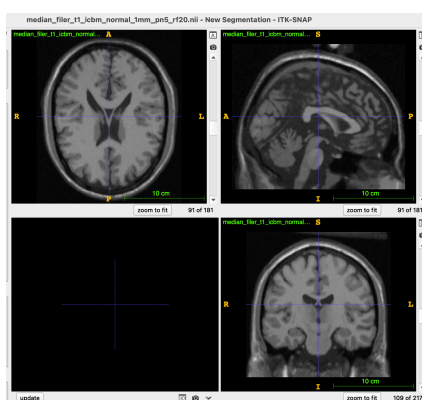


Figure 8: Imagen filtro mediana ruido 5%

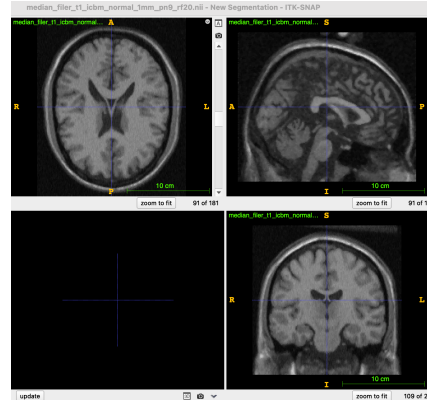


Figure 9: Imagen filtro mediana ruido 9%

4) Filtro Mediana adaptativo

Para el caso del algoritmo de filtro d mediana adaptativo, vemos que el proceso de suavizamiento de la imagen es más estricto que la mediana, observando una mayor similitud con la imagen con 0% de ruido. Esto desde la vista frontal. En el caso del filtro aplicado a la imagen de 9% observamos otra vista donde se observa mejor el corte sagital de la MRI.



Figure 10: Imagen filtro mediana adaptativo ruido 0%

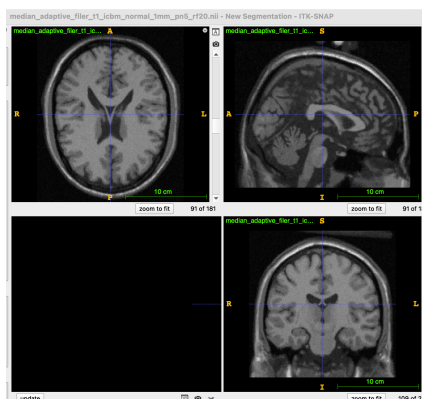


Figure 11: Imagen filtro mediana adaptativo ruido 5%

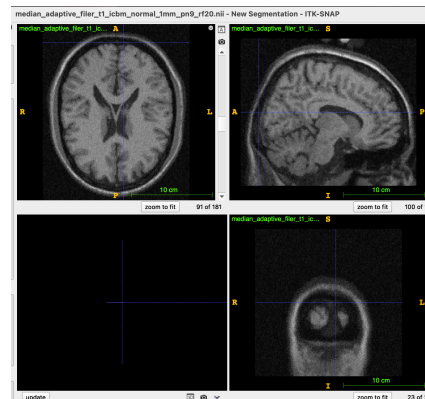


Figure 12: Imagen filtro mediana adaptativo ruido 9%

5) Filtro de Weiner

Este filtro busca principalmente ajustar la salida en función de la varianza local y la desviación standard de los pixeles que se encuentran dentro del filtro definido en una ventana de X,Y, para la implementación del ejercicio se utilizó una matriz cuadrada, el efecto observado en las imágenes en primera instancia pareciera preservar el borde y mejorar las regiones de la imagen en las que el ruido adiciona alta frecuencia , de acuerdo a las características del filtro es útil en el uso de superficies rugosas, como las imágenes tratadas en el ejercicio.

En este caso aplicamos un filtro de 5X5, y observamos un cambio radical en las imágenes para todos los pixeles, principalmente una forma de oscurecer las imágenes, esto puede deberse a que el tamaño de la caja al cual calcula es grande y la varianza y media de los pixeles puede generar el contraste de colores entre las imágenes.

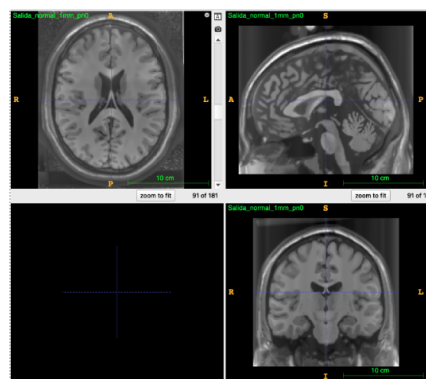


Figure 13: Imagen filtro Wiener adaptativo ruido 0% rectangulo 5x5

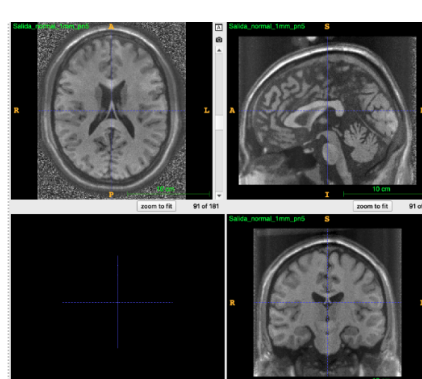


Figure 14: Imagen filtro Wiener adaptativo ruido 5% rectangulo 5x5

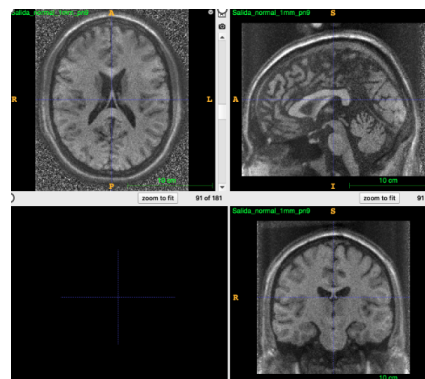


Figure 15: Imagen filtro Wiener adaptativo ruido 9% rectangulo 5x5

Para validar el estado del filtro, realizamos un proceso analogo con rectangulos 1x1, donde podemos ver un suavizamiento de la imagen analogo al de la mediana, sin contar los bordes los cuales quedaron también transformados en sus pixeles.

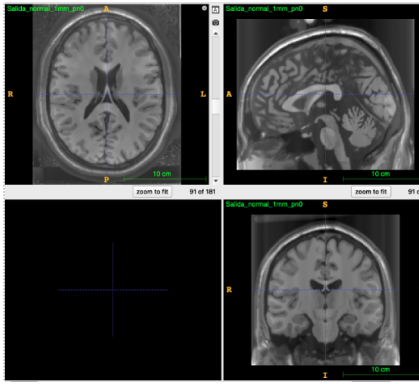


Figure 16: Imagen filtro Wiener adaptativo ruido 0% rectangulo 1x1

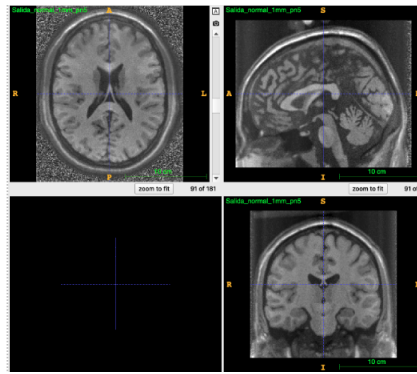


Figure 17: Imagen filtro Wiener adaptativo ruido 5% rectangulo 1x1

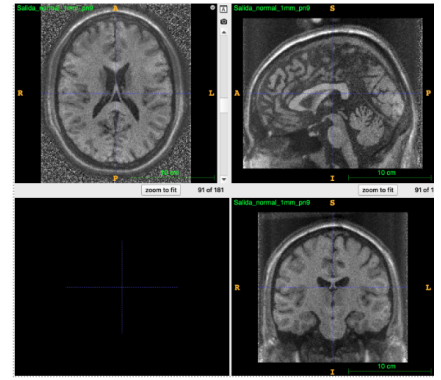


Figure 18: Imagen filtro Wiener adaptativo ruido 9% rectangulo 1x1

6) Estructura de los scripts Para ejecutar los scripts adjuntos a este documento, nosotros usamos una versión de python 3.9.13.

Adjuntamos adicionalmente los requirements.txt necesarios para tema de configuración.

Ya para la ejecución de los mismos, el script se ejecuta en consola y los parametros que debe recibir el script son el *input – pathdelaimagen output – pathdelaimagen*. Donde la imagen a leer, es una imagen en formato .mnc.gz y el formato de retorno es .nii.

Notese que dentro de los scripts usamos la libreria ITK para la conversión de tipo de dato y su posible tratamiento y retornamos en .nii, para la visualización a través del aplicativo ITK-SNAP.

Agregamos excepción en caso de enviar más o menos parametros a los solicitados. A continuación una imagen donde ilustra la presentación de dichos casos en una consola.

```
jessuarez@C0880TD7LPHW2 taller_3_filtrado_papper % python median_filter.py data/t1_icbm_normal_1mm_pn0_rf20.mnc.gz output_data/median_filer_t1_icbm_normal_1mm_pn0_rf20.nii
median_filter.py
data/t1_icbm_normal_1mm_pn0_rf20.mnc.gz
output_data/median_filer_t1_icbm_normal_1mm_pn0_rf20.nii
jessuarez@C0880TD7LPHW2 taller_3_filtrado_papper % python median_filter.py data/t1_icbm_normal_1mm_pn0_rf20.mnc.gz output_data/median_filer_t1_icbm_normal_1mm_pn0_rf20.nii 43 4
2
median_filter.py
data/t1_icbm_normal_1mm_pn0_rf20.mnc.gz
output_data/median_filer_t1_icbm_normal_1mm_pn0_rf20.nii
43
42
Usage: median_filter.py path input image .mnc.gz
path output image .nii
usar solo estos parametros
jessuarez@C0880TD7LPHW2 taller_3_filtrado_papper % python median_filter.py data/t1_icbm_normal_1mm_pn0_rf20.mnc.gz
median_filter.py
data/t1_icbm_normal_1mm_pn0_rf20.mnc.gz
Usage: median_filter.py path input image .mnc.gz
path output image .nii
usar solo estos parametros
jessuarez@C0880TD7LPHW2 taller_3_filtrado_papper %
```

Figure 19: Ejemplo consola

Para la ejecución del script de filtro de Weiner, ejecutarlo por consola sin parametros y en ejecución el solicitara el path de entrada y el path de salida para trabajar las imagenes.

7) Conclusiones

- ITK ofrece herramientas sencillas para el uso y tratamiento de imágenes, sin embargo, para el desarrollo del ejercicio, se debió recurrir a numpy, para poder trabajar los cálculos realizados para producir la imagen de salida.
- Para este grupo particularmente fue muy confuso lograr entender el manejo de las imágenes que poseen tres vistas, vs las imágenes comprendidas únicamente por dos dimensiones (png, Tif, etc), ya que el proceso de lectura por medio de los readers de las librerías nos generaron varios problemas, que nos llevó a iterar en el proceso de modelado y creación de los algoritmos, para obtener resultados.

- Observamos que los algoritmos permiten realizar cambios de visualización y eliminan el ruido de la información que se genera, lo cual permite aplicar en casos prácticos, para reducir la presencia de ruido y tener imágenes claras de MRI y con ello hacer los análisis clínicos que sean necesarios.
- En temas de implementación, para Wiener observamos que la selección de los parámetros de la caja son importantes, ya que cuadriláteros grandes pueden influir en el tipo de suavizado y cambiar la naturaleza de la imagen totalmente.