

Variación de número de puntos, ángulos y algoritmos

A fin de realizar la experimentación con los datos simulados, es importante abordar con anterioridad las siguientes cuestiones:

1. ¿Qué librerías se usarán para la implementación?
2. ¿Cómo definir una precisión de reconstrucción?
3. ¿Qué algoritmos tienen mejor desempeño (según precisión y tiempo de ejecución)?
4. ¿Cuál es el mínimo rango angular para obtener una precisión definida?
5. ¿Cuál es el mínimo número de puntos posibles para obtener una precisión definida?

1. Librerías

Se utilizó la librería *TomoPy* para realizar las reconstrucciones tomográficas, para las cuales requiere los datos (muogramas) y los ángulos de los detectores (puntos de observación) respecto al centro de rotación (punto de referencia).

2. Precisión de reconstrucción

Por el momento, se utiliza el rmse y el mae como métrica para medir la precisión de las reconstrucciones. De momento no se ha establecido una precisión definida (ej. una precisión del 90%), pero sí podemos analizar las diferentes reconstrucciones comparando su error, siendo la de menor valor la más similar a la original. El mae es más fácil de leer (En promedio se tiene un error de x unidades respecto a la original), de esta manera, podríamos definir por ejemplo que el error no sea mayor a 100 metros. Por otra parte, el rmse parece detectar mejor el incremento de error.

3. Elección de algoritmo

Comenzamos con un ejemplo sencillo, reconstruyendo el objeto Shepp-Logan.

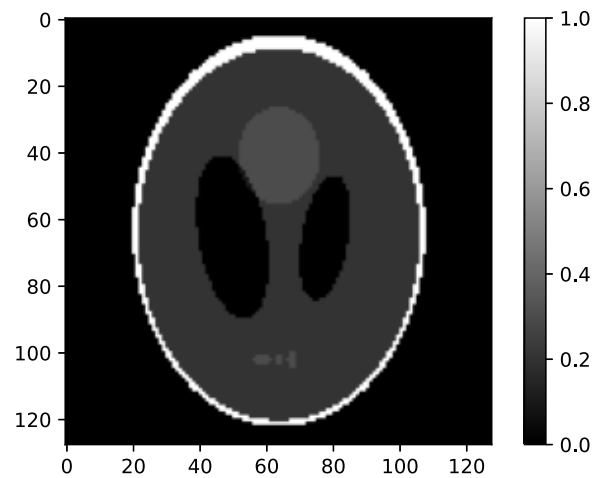


Fig 1. Corte transversal del objeto Shepp-Logan.

Este es un volumen disponible en la librería TomoPy, de dimensiones 128x128x128 píxeles, cuyos valores internos son unidades de densidad. La idea es sacarle proyecciones en diferentes ángulos (también posible desde TomoPy) y con esas proyecciones reconstruirlo, obteniendo nuevamente un volumen, idealmente igual al original. Vamos a usar todos los algoritmos de reconstrucción con los que cuenta la librería. Posteriormente, con el método elegido, vamos a reconstruir con variaciones de número de puntos y ángulos, y con esto medir la precisión de esas reconstrucciones.

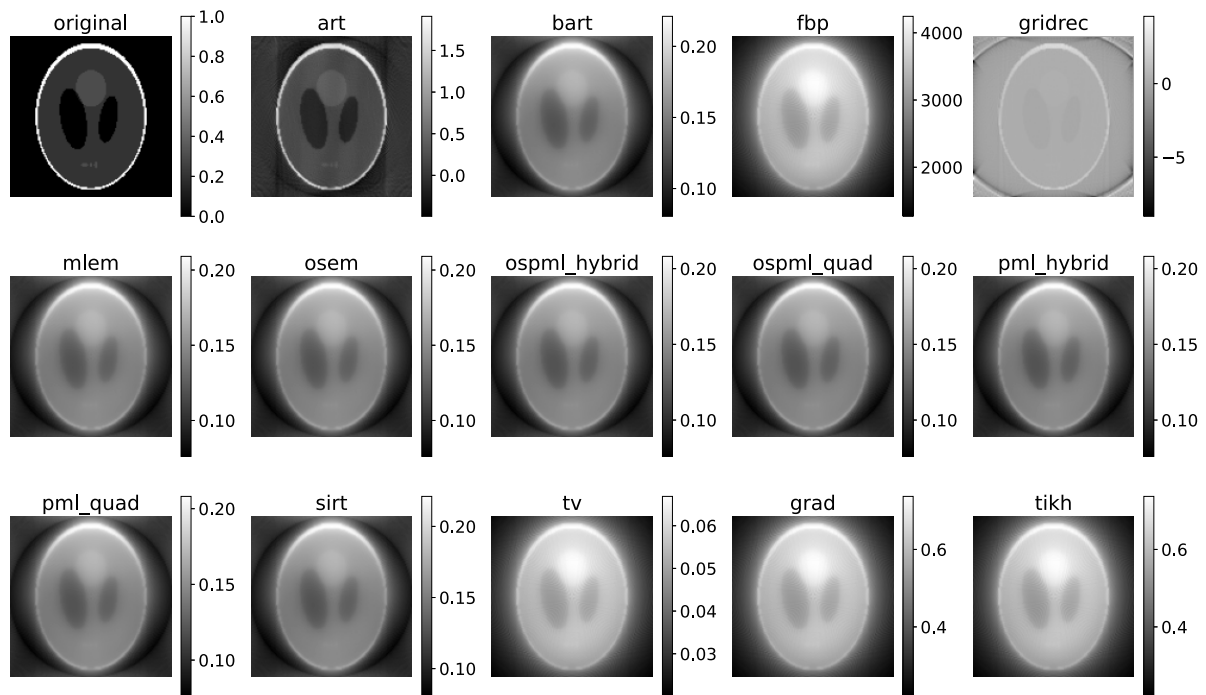


Fig 2. Reconstrucciones con diferentes algoritmos con 180 puntos de observación distribuidos en un rango de 0 a 180 grados.

```

art: 1.39181 seconds
bart: 4.68652 seconds
fbp: 3.89114 seconds
gridrec: 1.30684 seconds
mlem: 5.20035 seconds
osem: 4.63202 seconds
ospml_hybrid: 5.1935 seconds
ospml_quad: 4.83098 seconds
pml_hybrid: 4.81047 seconds
pml_quad: 5.71941 seconds
sirt: 4.64881 seconds
tv: 4.90999 seconds
grad: 4.7982 seconds
tikh: 4.87174 seconds

```

Fig 3. Tiempos de ejecución de algoritmos.

De la figura 2 y 3 se observa que el algoritmo que mejores resultados obtiene es el art, por lo tanto, se empleará este en el resto de experimentos. Puede que los otros métodos obtuvieran mejores resultados si TomoPy tuviera implementado los filtros, pero debido al desempeño del art, como también la facilidad de uso de la librería, en el contexto de este módulo siendo parte de una herramienta para uso libre, decidimos seguir adelante.

Se consideró la librería AstraToolbox, pero los resultados no fueron satisfactorios en ningún caso, además de su mayor complejidad de uso.

3.1 Algebraic Reconstruction technique (ART)

Este es un método iterativo, expresado por la siguiente ecuación:

$$x^{k+1} = x^k + \lambda_k \frac{b_i - \langle a_i, x^k \rangle}{||a_i||^2} a_i^T$$

El algoritmo itera por un número de iteraciones k dado, en cada iteración itera por cada detector b , y por cada detector itera por cada pixel de este. Se inicializa el volumen a reconstruir x , y en cada pixel se traza una línea transversal al eje de proyección, donde se calculan las distancias a de los voxeles que intersectan la línea. Puede decirse que se llega a una solución cuando la suma de los voxeles que intersectan la línea de proyección, multiplicados por su respectiva distancia, igualan al valor original encontrado en el pixel del detector. En otras palabras, se llega a una solución cuando el valor encontrado en el pixel del detector se ha distribuido correctamente a lo largo del volumen.

4. Variación de amplitud angular con número de puntos de observación fijo

Una vez elegido el algoritmo, reconstruimos con 180 puntos de observación, variando la amplitud angular en 180° , 90° , 60° y 45° , como se muestra en la siguiente figura:

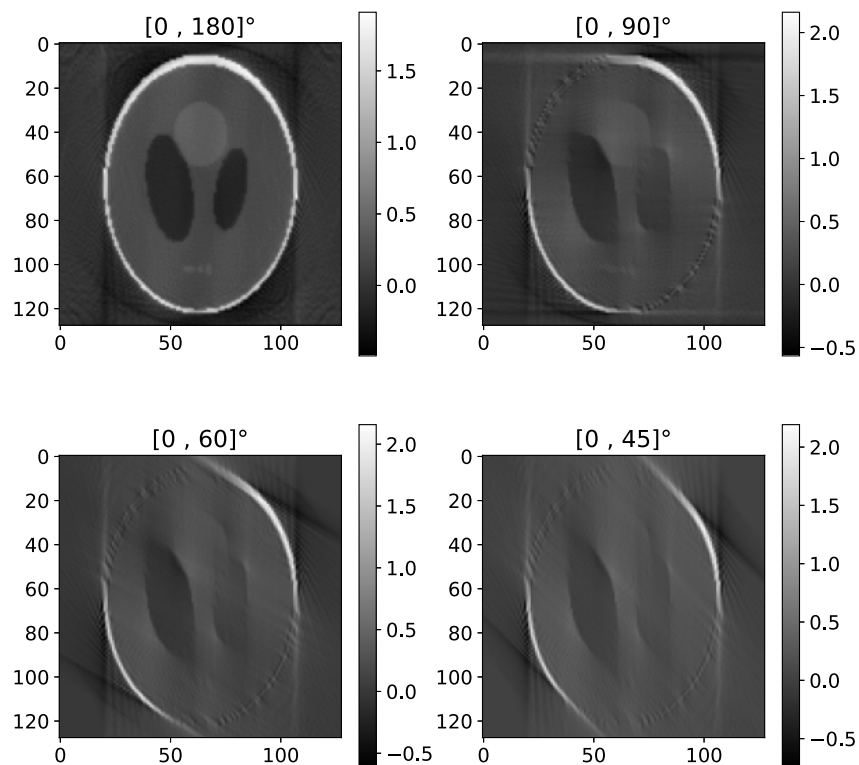


Fig 4. Reconstrucciones con 180 puntos y variación de amplitud angular.

Podemos observar que la forma del objeto se va perdiendo a menor amplitud angular. Esto indica que no solo basta con tener suficientes puntos de observación, sino también que estén

distribuidos en la mayor amplitud posible. También a partir de la figura se demuestra que con una amplitud de 180° se reconstruye la totalidad de la reconstrucción, como indica los límites de la integral en la solución analítica.

5. Variación de número de puntos de observación con amplitud angular fija

Esta vez fijamos una amplitud angular de 180° , y creamos puntos de observación distribuidos uniformemente en ese rango, la siguiente figura muestra 4 reconstrucciones con 4, 16, 64 y 256 puntos de observación.

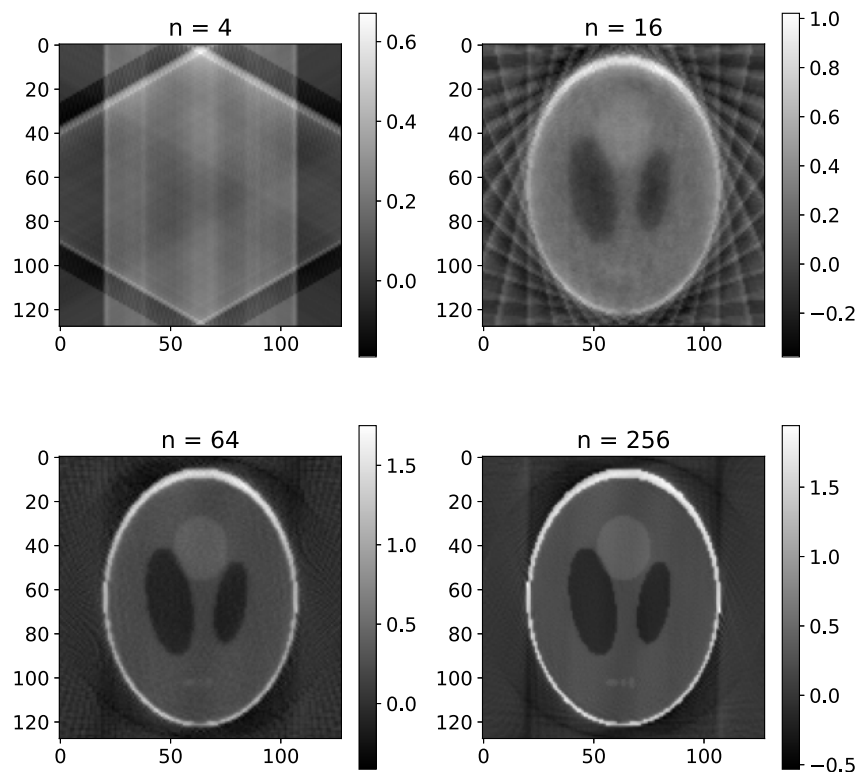


Fig 5. Reconstrucciones con amplitud de 180° y variación de número de puntos.

De la figura 4 es evidente que se obtiene una mejor resolución de imagen a mayor número de puntos de observación. Sin embargo, la reconstrucción de 64 puntos tiene un error menor a la de 256 puntos; ¿por qué? Si hacemos cuentas, 256 puntos ubicados en un rango de 180° implica que hay un punto cada 0.7° . En otras palabras, los puntos están ubicados casi en el mismo lugar, y parece que un número excesivo de puntos genera un exceso en la magnitud de la reconstrucción, como muestra la barra de colores.

En resumen, se debe tener suficientes puntos para que la reconstrucción tenga una resolución similar al volumen original, pero demasiados puntos generan un exceso en la magnitud respecto a la original. La siguiente gráfica muestra los errores obtenidos en diferentes reconstrucciones variando su número de puntos de observación, todos distribuidos en un rango de 180° :

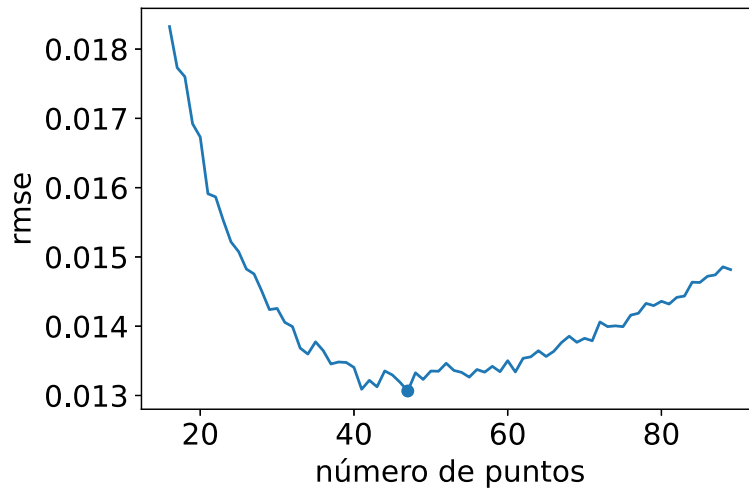


Fig 5. Error obtenido según el número de puntos de observación.

Se obtuvo un error mínimo con 47 puntos. Si bien indudablemente se obtiene una mejor resolución con 180 puntos, el exceso que se tiene produce un error mayor.

Ahora bien, recordemos que el algoritmo es un método iterativo. Todas las reconstrucciones hechas hasta el momento han sido con una iteración. La siguiente gráfica muestra el error de una reconstrucción de 180 puntos según el número de iteraciones:

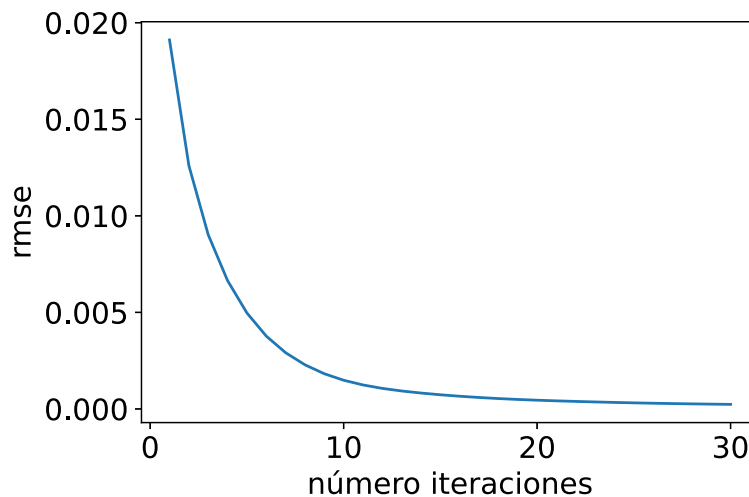


Fig 6. Error obtenido en reconstrucciones de 180 puntos variando el número de iteraciones.

Podemos observar que el error disminuye hasta converger a medida que se incrementa el número de iteraciones. En este caso el error converge a 0.

Finalmente, podemos combinar ambos experimentos para observar el error variando el número de puntos en diferentes iteraciones:

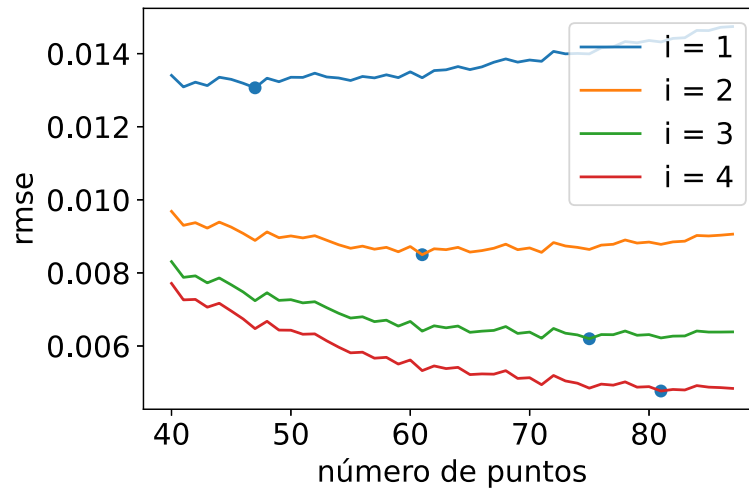


Fig 7. Error mínimo encontrado en diferente número de puntos para diferente número de iteraciones.

De la figura 7 podemos señalar que el error disminuye a mayor número de iteraciones, como ya se había visto en la figura 6, pero también que el error mínimo se encuentra en un mayor número de puntos a mayor número de iteraciones, aunque igualmente se encuentra en el mínimo de una parábola, como se muestra en la figura 5.

Conclusiones

Como conclusiones listamos un resumen de las observaciones hechas:

- A mayor número de puntos, mejor es la resolución de la reconstrucción.
- A mayor amplitud angular, mejor es la forma de la reconstrucción.
- Un exceso de número de puntos representa un exceso en la magnitud de la reconstrucción.
- Un mayor número de iteraciones mejora la reconstrucción, hasta cierto momento donde no mejorará más.
- Existe un número de puntos de observación que genera la mejor reconstrucción, y al incrementar el número de iteraciones, este número de puntos puede aumentar para seguir disminuyendo el error.