

Universidad Nacional de Córdoba



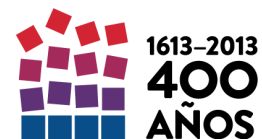
*Tesis para optar por el Título de
Magister en Análisis y Procesamiento de Imágenes*

Algoritmos de procesamiento y reconstrucción tomográfica para imágenes radiológicas

7 de Julio de 2016

Autor:
Lic. Mariano Arselan

Supervisor:
Dr. Francisco Malano



Esta obra está licenciada bajo la Licencia Creative Commons Atribución-NoComercial 2.5 Argentina.
Para ver una copia de esta licencia, visita este link: <http://creativecommons.org/licenses/by-nc/2.5/ar/>.



Dedicado a Bodoque

Agradecimientos

A Francisco Malano, Mauro Valente, José Vedelago, Jesús Calderón, Oscar Bustos, Nicolás Wolovick, y al equipo del CCAD (Centro de Computación de Alto Desempeño) de la Universidad Nacional de Córdoba.

Resumen

El presente trabajo tiene el propósito de implementar dos códigos computacionales: el primero es un simulador de proyecciones radiográficas de haz cónico monocromático aplicadas a una muestra, mientras que el segundo código implementa un algoritmo de reconstrucción tomográfica, en particular el algoritmo FDK [11], a partir de las imágenes obtenidas en la simulación del primer código. En la implementación se utilizan distintas técnicas de procesamiento de imágenes, como filtros, transformadas de Fourier, y wavelets. Además, se realiza una evaluación de la calidad de las imágenes y de las reconstrucciones tomográficas obtenidas.

Palabras clave: *FDK, Feldkamp, Davis, Kress, reconstrucción tomográfica, haz cónico, simulación, Monte Carlo, Rayos X*

Abstract

The purpose of this work is to implement two packages of computer software: the first one consists in a code that simulates radiographic projections using a monochromatic conical beam of X-rays applied to a given sample, whereas the second one implements an algorithm known as FDK [11], whose goal is to make tomographic reconstructions from a set of projections, and in this particular work, from the simulated images. In our implementation we use different image processing techniques, such as filters, Fourier and wavelets, among others. In addition to that, we assesses the quality of the output images from both the simulator and from the FDK implementation.

Key words: *FDK, Feldkamp, Davis, Kress, tomographic reconstruction, cone-beam, simulation, Monte Carlo, X-rays*

Recursos computacionales

Los siguientes son los recursos de software y hardware principales utilizados en la elaboración de este trabajo.

Software

- MATLAB® 2015b, student edition, licencia a nombre de Mariano Arselan.
- GNU Octave, versión 3.6
- Qt open source versión 5.3, con QtCreator open source 3.1
- Compilador g++ en Linux Fedora, CentOs y Ubuntu.
- Compilador gfortran en Linux Fedora, CentOs y Ubuntu.
- Librería Open MPI (versión 1.8.3 en cluster Mendieta)
- Iconion, licencia comercial a nombre de Mariano Arselan
- Gimp, Inkscape, LibreOffice Impress, Texmaker

Hardware

Este trabajo utilizó recursos computacionales del CCAD de la Universidad Nacional de Córdoba (<http://ccad.unc.edu.ar/>) en particular el Cluster Mendieta, el cual forma parte del SNCAD del MinCyT de la República Argentina.



Según requerimientos de las simulaciones ejecutadas en este cluster, se utilizaron entre 16 y 144 cores en procesadores Intel® Xeon® E5-2680.

Índice general

1. Introducción	8
1.1. Requisitos técnicos	8
1.2. Marco histórico	8
1.3. Hipótesis y objetivos	10
1.3.1. Objetivo General	10
1.3.2. Objetivos específicos	10
2. Materiales y métodos	12
2.1. Marco teórico	12
2.1.1. Introducción	12
2.1.2. Transformada Radon	13
2.1.3. Teorema de proyección	15
2.1.4. Forma Polar de Fourier	16
2.1.5. Filtros	17
2.1.6. Reconstrucción tomográfica con haz paralelo	19
2.1.6.1. Retroproyección	19
2.1.6.2. Retroproyección filtrada	20
2.1.7. Reconstrucción tomográfica con haz de abanico	21
2.1.8. Reconstrucción tomográfica con haz cónico	26
2.2. Implementación	30
2.2.1. simrx - PENELOPE	30
2.2.2. DICOM	32
2.2.3. Algoritmos de reconstrucción tomográfica	34
3. Resultados	36
3.1. simrx	36
3.2. Haz paralelo	37
3.2.1. Introducción	37
3.2.2. Reconstrucción del cubo	38
3.2.3. Shepp-Logan	38
3.2.4. Barra de silicio simulada	39

3.3. Haz de abanico	42
3.3.1. Introducción	42
3.3.2. Resultados	42
3.4. Haz cónico	43
3.4.1. Introducción	43
3.4.2. Reconstrucción del cubo	45
3.4.3. Reconstrucción del fantoma Shepp-Logan	46
3.4.4. Reconstrucción de la simulación cubo-cilindro	51
4. Conclusiones	58
Bibliografía	60

Capítulo 1

Introducción

1.1. Requisitos técnicos

Para la correcta comprensión de este texto, se requiere que el lector tenga noción de teoría de procesamiento de señales, conocimiento básico de la física relacionada con el transporte de radiación, en particular con la interacción de las ondas electromagnéticas con la materia, simulaciones de Monte Carlo, cálculo infinitesimal, álgebra lineal, y una base de programación de Matlab, como así también una idea de computación distribuída en clusters.

1.2. Marco histórico

Los antecedentes históricos a este trabajo, que incluyen los descubrimientos y avances en física médica y radiología clínica producidos en un período de más de cien años, son tan numerosos y complejos que es necesario limitar el marco histórico al contexto del desarrollo de este material, cuyo punto más importante es el de la reconstrucción tomográfica con haz de rayos X de geometría cónica.

Resulta una tarea difícil, como se indica en [35], establecer quién fué la primera persona que descubrió un determinado fenómeno físico o que construyó un dispositivo para aprovechar y aplicar ese fenómeno al campo de la medicina. Mas aún, en [35] se trata de distinguir entre los trabajos realizados en laboratorio y los dispositivos que tuvieron sus inicios en aplicaciones clínicas.

Debemos comenzar, sin embargo, con *Wilhelm Conrad Röntgen*, físico alemán que en Noviembre de 1895, siendo profesor en Wurzburg, descubrió en su laboratorio una radiación diferente a la que llamó rayos X (recordar que en álgebra, se denota con una X a cualquier magnitud desconocida). En Diciembre de ese

mismo año, Röntgen logra capturar una imagen en una placa fotográfica. Se trataba de una radiografía de la mano de su esposa. Luego, ese mismo mes de Diciembre, apenas 40 días después de su observación inicial, Röntgen envía una publicación llamada '*preliminary communication of the new kind of ray*', que luego sería publicada en Enero de 1896. Finalmente, Röntgen recibe el premio Nobel en 1901 por este descubrimiento.

Sin embargo, es interesante notar en [12] que *Goodspeed* y *Jennings* habían obtenido accidentalmente una radiografía con la imagen de unas monedas unos años antes, en 1890, pero tal incidente había pasado desapercibido por ellos hasta que el anuncio de Röntgen hizo que revisaran su experimento.

En 1917, el matemático Austríaco *Johann Radon* desarrolla una teoría en el marco de la geometría integral, en la cual nos interesa la denominada Transformada Radón, estructura matemática utilizada para obtener una reconstrucción en tres dimensiones de una función a partir de proyecciones, conocida como **reconstrucción tomográfica**. Actualmente, por supuesto, las tomografías se construyen de manera computacional. Algunos de estos son implementados en el presente trabajo.

Quién inventó la tomografía computada, abreviada como *TC*, es objeto de debate.

El concepto original se lo adjudica usualmente a Radon (1917). Oldendorf (1961) es referido como el primero que publicó los resultados de un experimento de TC con un fantoma. Lo que hizo Oldendorf en realidad fué rotar el fantoma, consistente en una caja con clavos, sobre la plataforma giratoria de un gramófono, mostrando las señales producidas por las distintas proyecciones a distintos ángulos. Aunque conciente de las implicancias médicas del experimento, en realidad Oldendorf no realizó la reconstrucción tomográfica.

Durante los años 60's, se publicaron varios trabajos sobre el potencial de la TC incluyendo sugerencias sobre distintas técnicas y métodos tanto en experimentos como en simulación, entre ellos, el de Cormarck. El caso menos conocido es quizás el de un tomógrafo construido en Rusia en 1958. Korenblyum *et al* (1958), publicó el desarrollo matemático de la reconstrucción junto con los detalles del experimento. En esa publicación se puede ver que utilizaron una reconstrucción analógica basado en un haz de rayos X de geometría de abanico. Incluso hay reportes anteriores, también provenientes de Rusia, como el caso de Tetel' Baum en 1957.

En 1972, durante la conferencia anual del Instituto Británico de Radiología, *Hounsfield* dió la noticia de una máquina usada para realizar tomografías directamente en uso clínico, lo cual significaba un gran salto para el campo de la

radiología desde el descubrimiento de Röntgen. Las sucesivas publicaciones de Hounsfield y Ambrose fueron consideradas el comienzo de la tomografía computada con rayos X para aplicaciones clínicas. Hounsfield recibió el Nobel de Fisiología y Medicina en 1979 junto a Cormack. Lo que se dejó en claro es que Hounsfield nunca se proclamó el real inventor de la TC. El anuncio de 1972 fue la primera experiencia real de la técnica, que derivó en un gran interés para aplicaciones clínicas en los años siguientes.

En cuanto a reconstrucción tomográfica volumétrica, métodos de reconstrucción 3D directos han sido propuestos como se detalla en Altschuler [1], aunque no se conocen tomógrafos comerciales que empleen estas técnicas. También métodos iterativos han sido propuestos como por ejemplo en Colsher [6], Altschuler [2] y Schlindwein [28]. El uso de funciones base fue explorada también por Altschuler [3] y por Kowalski [17]. Minerbo [21] usa la inversa de la transformada Radon en 3D para derivar una solución aproximada. Procedimientos descritos por Herman [34], y por Lewitt y McKay [20], en el que se utiliza el algoritmo de haz de abanico sin modificaciones en casos donde el ángulo del haz cónico es pequeño. Finalmente, Feldkamp, Davis y Kress [11] desarrollan un algoritmo de reconstrucción basado en el de haz de abanico, simplemente cambiando el sistema de coordenadas, tal como se muestra en el desarrollo de este trabajo.

1.3. Hipótesis y objetivos

1.3.1. Objetivo General

Estudiar, desarrollar e implementar en códigos computacionales algoritmos de procesamiento y reconstrucción tomográfica para imágenes obtenidas mediante simulación Monte Carlo.

1.3.2. Objetivos específicos

Dentro del marco del objetivo general, están contemplados los siguientes objetivos específicos:

1. Revisión y estudio de los procesos físicos básicos de interacción entre radiación y materia involucrados en las técnicas de adquisición de imágenes radiológicas.
2. Estudio básico del protocolo de codificación y comunicación de imágenes médicas DICOM, que es un estándar universal utilizado en todos los centros

de salud para compartir imágenes médicas. Implementación de un codificador de formato de imagen RAW (crudo) a archivos de formato DICOM.

3. Estudio, adaptación y paralelización del código de simulación Monte Carlo PENELOPE para generar proyecciones radiográficas a fin de poder servir de entrada de datos al algoritmo de reconstrucción tomográfica especificado en el siguiente punto.
4. Estudio e implementación del algoritmo de reconstrucción tomográfica FDK [11].

En el marco de los objetivos específicos se establecen entonces las siguientes hipótesis:

- Las simulaciones de Monte Carlo implementadas en este trabajo generan proyecciones radiográficas cuya señal tiene un error relativo menor al 2% con respecto a los valores teóricos.
- Es posible utilizar las imágenes generadas en el punto anterior para ejercitar el algoritmo FDK. Las reconstrucciones tomográficas utilizando la implementación desarrollada en este trabajo producen imágenes cuyas señales tienen un error relativo menor al 1% con respecto a su valor teórico, dentro del área de reconstrucción en donde existen la totalidad de los datos.

Las hipótesis anteriores son verificadas en el capítulo 3.

Capítulo 2

Materiales y métodos

2.1. Marco teórico

2.1.1. Introducción

El primer requerimiento necesario para poder comenzar a trabajar en el campo de las imágenes radiológicas, es tener un buen entendimiento de la **interacción entre la radiación electromagnética y la materia**. Los rayos X, con su capacidad de atravesar y ser absorbidos por objetos de distinto material y densidad, tienen como efecto producir contraste entre las distintas estructuras internas de esos objetos, siendo estas diferencias de contraste capturadas con un detector apropiado para formar una imagen.

Los efectos producidos al someter a un cuerpo bajo la radiación de rayos X, los cuales incluyen el efecto Compton, fotoeléctrico, scattering coherente, ionización, excitación y desexcitación de átomos, etc, se estudian en [12], [30], [5] y [32].

En segundo lugar, es deseable contar con un mecanismo que pueda producir imágenes radiológicas sin tener que recurrir a un dispositivo de rayos X real. Entre todos los métodos existentes para lograr este propósito, el utilizado en este trabajo se basa en los **métodos de simulación de Monte Carlo**. Estos métodos se estudian en detalle en [33]. Consisten en generar valores de distribuciones aleatorias con el fin de simular procesos estocásticos. En particular nos interesa simular los efectos entre radiación y materia descritos anteriormente, cuyo resultado final de todo el proceso será una proyección radiográfica. La realización de las imágenes necesarias para ser utilizadas en este trabajo se generan utilizando un código de simulación de transporte de radiación, llamado **PENELOPE**, que sirve para nuestro propósito. Este código, y la extensión que desarrollamos, llamada **simrx**, se describen en la sección 2.2.

En las secciones siguientes, se desarrolla en detalle la matemática asociada para la implementación de los algoritmos de reconstrucción tomográfica, en cumplimiento con los objetivos planteados, y cuya ejecución permitirá comprobar o refutar las hipótesis planteadas.

2.1.2. Transformada Radon

Consideremos la geometría de la figura 2.1, la cual muestra un cuerpo (en color gris) constituido por un material con coeficiente de atenuación μ , y que es atravesada por rayos paralelos con intensidad inicial I_0 . Los rayos llegan al detector plano (en rojo), produciendo una señal I_θ . Esta señal se ve en la figura 2.2 (a). La inclinación de los rayos con respecto al eje central Y forman un ángulo θ . Estos son perpendiculares al plano detector. Se forma un nuevo sistema de coordenadas, cuyo eje s es paralelo a los rayos, mientras que el eje r es perpendicular a estos.

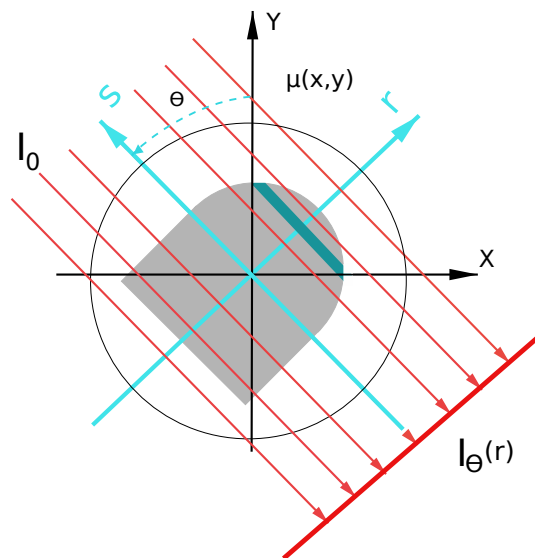


Figura 2.1: Geometría con haz paralelo.

Como se describe en [32], los eventos que ocurren a nivel atómico durante la interacción del campo electromagnético con la materia, son de tipo estocástico, pero a nivel macro, el resultado de la combinación de todos estos procesos produce una salida que es de carácter determinístico. Así, se puede calcular la intensidad de salida I_θ con respecto a la intensidad de entrada I_0 con la siguiente relación:

$$I_\theta(r) = I_0 \cdot e^{-\int_{L_{r,\theta}} \mu(x,y) ds}$$

donde $L_{r,\theta}$ son todos los puntos que forman la línea que pasa por r con un ángulo θ .

Teniendo en cuenta la siguiente relación:

$$x = r \cos \theta - s \sin \theta$$

$$y = r \sin \theta + s \cos \theta$$

tenemos que

$$I_{\theta}(r) = I_0 \cdot e^{-\int_{L_{r,\theta}} \mu(r \cos \theta - s \sin \theta, r \sin \theta + s \cos \theta) ds}$$

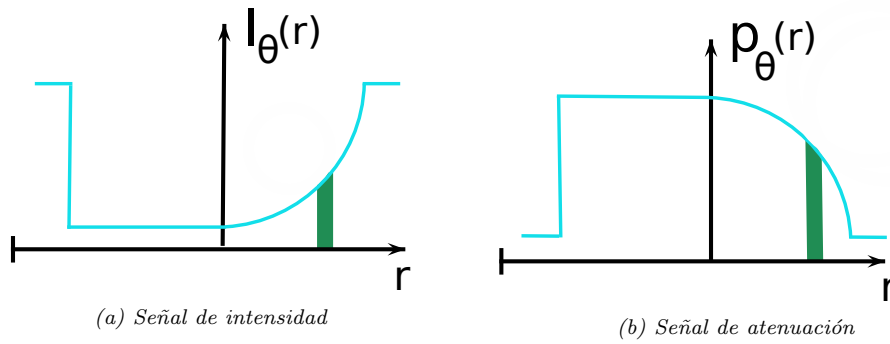


Figura 2.2: Perfiles obtenidos a partir de la figura 2.1

El perfil de atenuación, como el mostrado en la figura 2.2(b), se obtiene a partir del perfil de intensidad de la siguiente manera:

$$\begin{aligned}
 p_{\theta}(r) &= -\log \frac{I_{\theta}(r)}{I_0} \\
 &= \int_{L_{r,\theta}} \mu(r \cos \theta - s \sin \theta, r \sin \theta + s \cos \theta) ds
 \end{aligned}
 \tag{2.1}$$

donde $p_{\theta}(r)$ es la proyección de la función $\mu(x, y)$ a lo largo del ángulo θ . Si tomamos varias proyecciones a distintos ángulos θ y las apilamos, obtenemos un conjunto de datos llamado *sinograma*. En la figura 2.3 (a) tenemos un ejemplo de un fantoma consistente en un cubo de agua rodeando a un cubo de silicio. Si tomamos la proyección de atenuación a un ángulo $\theta = 0$ **en todos los planos**,

obtenemos la imagen mostrada en 2.3 (c). Si ahora nos concentramos en el plano medio de color verde, como se ve en la figura 2.3 (b), el cual corta a ambos cubos, y tomamos proyecciones a distintos ángulos sobre ese plano, obtenemos el sinograma mostrado en la figura 2.4 (a).

2.1.3. Teorema de proyección

Sea $f(x, y)$ una función de atenuación cualquiera, y sea $F(u, v)$ su transformada 2D de Fourier.

$$F(u, v) = \iint_{-\infty}^{+\infty} f(x, y) e^{-2\pi i(xu + yv)} dx dy$$

Dado un ángulo θ **fijo**, sea $p_\theta(r)$ el perfil de atenuación de la función $f(x, y)$ en ese ángulo, y sea $P(k, \theta)$ la transformada de Fourier de $p_\theta(r)$. Por definición tenemos entonces:

$$P(k, \theta) = \int_{-\infty}^{+\infty} p_\theta(r) e^{-2\pi i(kr)} dr$$

El teorema de proyección establece que:

$$P(k, \theta) = F(u, v) \tag{2.2}$$

con

$$\begin{aligned} u &= k \cos \theta \\ v &= k \sin \theta \\ k &= \sqrt{u^2 + v^2} \end{aligned} \tag{2.3}$$

La demostración de este teorema se desarrolla en [30].

Es muy importante notar que $P(k, \theta)$ **no es la transformada 2D de Fourier** de $p_\theta(r)$, sino que es la transformada 2D de Fourier de la función $f(x, y)$. Por otro lado, para cada θ fijo, por ejemplo para $\theta = \theta'$, $P(k, \theta')$ es la transformada 1D de Fourier de $p_{\theta'}(r)$.

Como mencionamos anteriormente, la figura 2.3 muestra un escenario consistente en un cubo de silicio contenido en un cubo de agua. Esto define la función de atenuación $f(x, y)$. Si consideramos todas las proyecciones de este escenario, obtenemos el sinograma de la figura 2.4(a). Si realizamos un corte en este sinograma para un determinado ángulo, digamos θ' (línea celeste), obtenemos la señal $p_{\theta'}(r)$. La transformada 1D de Fourier de esta señal es la correspondiente a la línea roja en 2.4(b), cuya señal es $P(\theta', k)$. La transformada 2D de Fourier de la imagen $f(x, y)$ corresponde a imagen $P(\theta', k)$ entera.

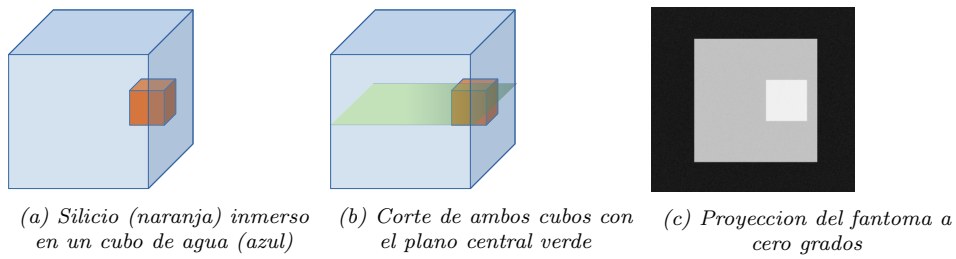


Figura 2.3

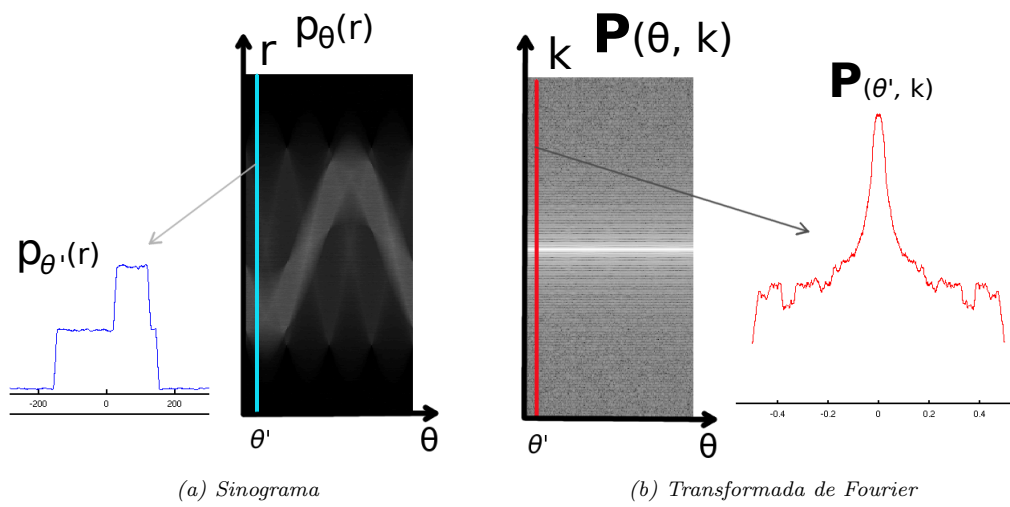


Figura 2.4: Perfiles de señal en el ángulo θ'

2.1.4. Forma Polar de Fourier

La ecuación de la transformada inversa de Fourier:

$$f(x, y) = \iint_{-\infty}^{+\infty} F(u, v) e^{2\pi i(xu+yv)} du dv$$

puede ser reescrita en forma polar usando las siguientes relaciones:

$$u = k \cos \theta$$

$$v = k \sin \theta$$

calculando el Jacobiano

$$J = \begin{vmatrix} \frac{\partial u}{\partial k} & \frac{\partial u}{\partial \theta} \\ \frac{\partial v}{\partial k} & \frac{\partial v}{\partial \theta} \end{vmatrix} = \begin{vmatrix} \cos \theta & -k \sin \theta \\ \sin \theta & k \cos \theta \end{vmatrix} = k (\cos^2 \theta + \sin^2 \theta) = k$$

nos queda que la forma polar es:

$$\begin{aligned}
 f(x, y) &= \int_0^{2\pi} \int_0^{+\infty} F(k \cos \theta, k \sin \theta) e^{2\pi i(x k \cos \theta + y k \sin \theta)} k dk d\theta \\
 &= \int_0^{\pi} \int_{-\infty}^{+\infty} F(k \cos \theta, k \sin \theta) |k| e^{2\pi i(x k \cos \theta + y k \sin \theta)} dk d\theta
 \end{aligned} \tag{2.4}$$

2.1.5. Filtros

En la ecuación 2.4, observamos el término $|k|$, el cual corresponde a la función módulo o rampa, la cual actúa como filtro en el espacio de frecuencias, ya que multiplica al término $F(k \cos \theta, k \sin \theta)$. Si pensamos en un momento en implementar esta fórmula en una computadora, debemos discretizar el dominio. En este caso, no integraremos en toda la recta real, sino que acotaremos la suma a una banda de frecuencia máxima k_{max} , la componente mas alta de Fourier. Este nuevo filtro recortado se denomina filtro **Ram-Lak**, nombre dado por sus inventores, Ramachandran y Lakshiminarayanan.

Veamos un ejemplo usando Matlab. En la figura 2.5 (a) se observa una imagen con el fantoma de Shepp-Logan. Se obtuvo su transformada Radon y a partir de esta se trata de reconstruir la imagen original usando la función de Matlab *iradon*. Cuando se realiza una típica reconstrucción tomográfica sin filtrado, se suele producir una imagen difusa (figura 2.5 (b)). El resultado de usar el filtro Ram-Lak entonces es la de eliminar las frecuencias bajas que producen este efecto, como se observa en 2.5 (c). Sin embargo, este filtro amplifica el ruido localizado en las frecuencias mas altas, por lo que en la práctica es usual acoplar el Ram-Lak con un filtro pasabajos, digamos **Q**, con lo cual la ecuación 2.4 se transforma en:

$$f(x, y) = \int_0^{\pi} \int_{-k_{max}}^{+k_{max}} F(k \cos \theta, k \sin \theta) \cdot |k| \cdot Q(k) \cdot e^{2\pi i(x k \cos \theta + y k \sin \theta)} dk d\theta \tag{2.5}$$

En 2.5 (d) se observa la reconstrucción realizada con 2.5 usando como filtro pasabajos Q un filtro Hamming. A pesar de conservar algunos artefactos, se observa que el ruido se ha reducido considerablemente. El filtro Hamming, junto con otros filtros pasabajos, se describen a continuación. En la figura 2.6 se observa la forma de los filtros Ram-Lak, Hamming, Hanning, o simplemente Hann, y el Butterworth, todos en el espacio de Fourier. La ecuación que satisfacen los filtros de Hamming y Hann se define en la ecuación 2.6. Hamming se define con $\alpha = 0,54$, mientras que para Hann tenemos $\alpha = 0,5$.

$$Q(k) = \begin{cases} \alpha + (1 - \alpha) \cos\left(\frac{\pi k}{k_{max}}\right) & \text{si } |k| < k_{max} \\ 0 & \text{si } |k| \geq k_{max} \end{cases} \tag{2.6}$$

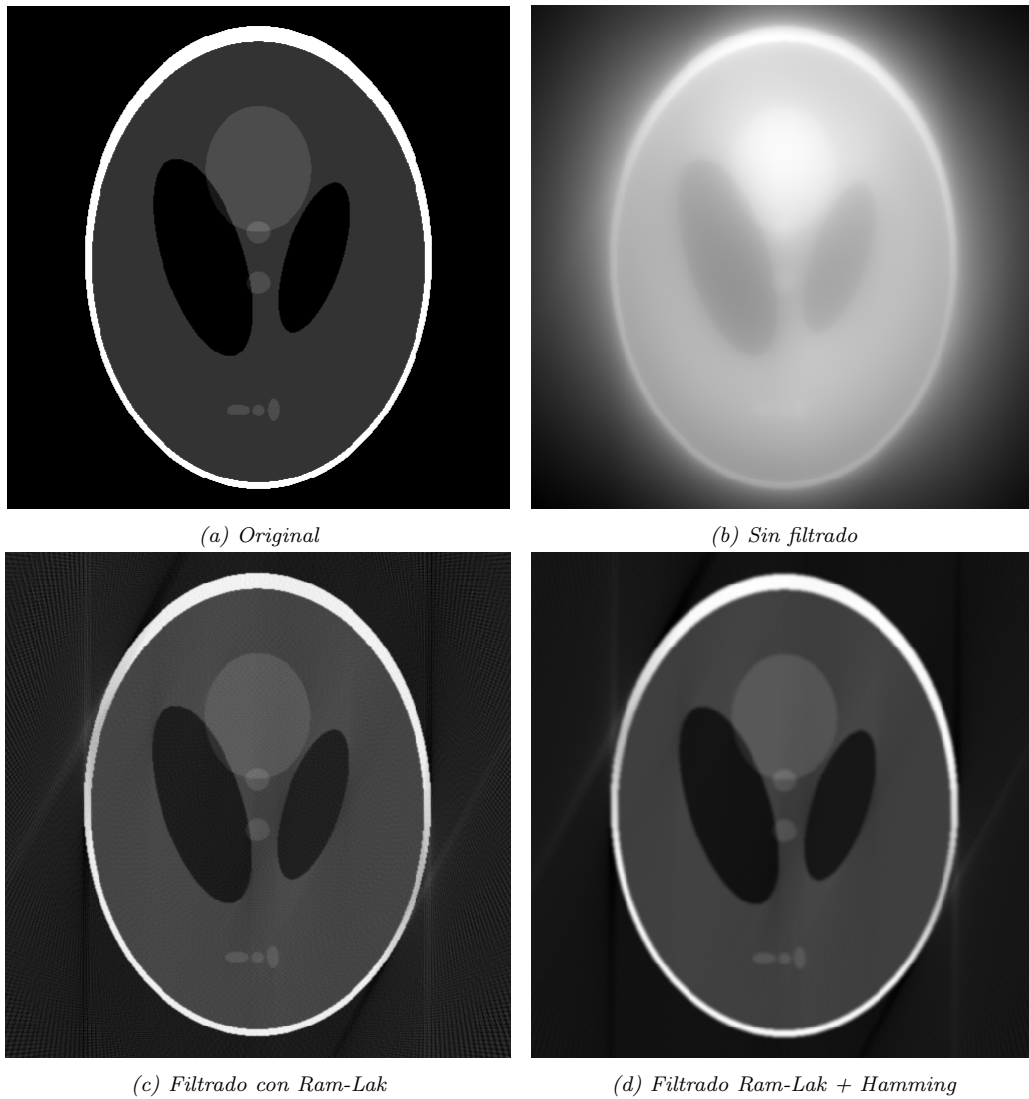


Figura 2.5: Reconstrucciones varias usando Matlab

En la ecuación 2.7 tenemos la definición de un filtro Butterworth. El parámetro n define el orden del filtro, mientras que el parámetro wc indica el valor de la frecuencia en que el valor en amplitud del filtro es igual a 0,5. En la figura 2.6 (c) se observan tres filtros Butterworth, de orden $n = 1, 2, 3$, para el mismo valor de wc .

$$Q(k) = \frac{1}{1 + \left(\frac{k}{wc}\right)^{2n}} \quad (2.7)$$

En la figura 2.6 (d) se observa el resultado de multiplicar un filtro Hamming, un Hann y un Butterworth con un filtro Ram-Lak.

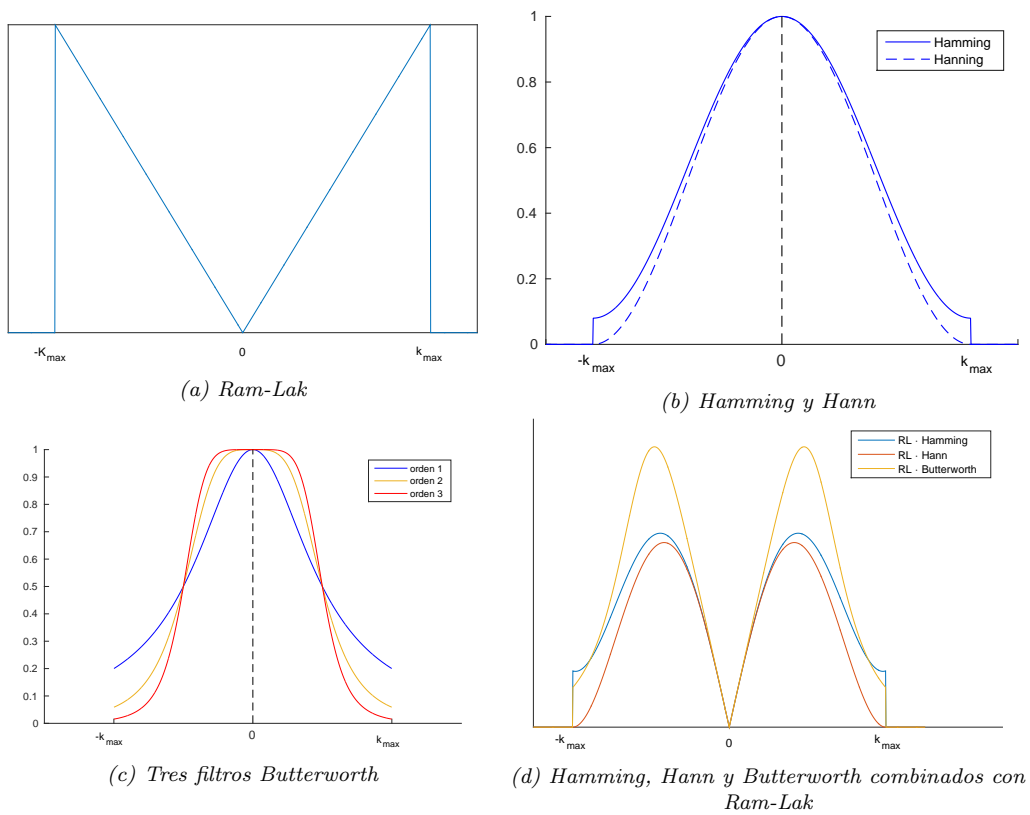


Figura 2.6: Ejemplos de filtros usados en una reconstrucción tomográfica

De ahora en adelante, por razones prácticas, sobre todo en la implementación de estos filtros en Matlab, en lugar de utilizar el rango de frecuencias desde cero hasta k_{max} directamente, usaremos el rango $[0, 1]$, con el valor 1 correspondiendo a k_{max} .

2.1.6. Reconstrucción tomográfica con haz paralelo

2.1.6.1. Retroproyección

Una forma de aproximar la función $f(x, y)$ a partir de su sinograma $p_\theta(r)$, es usar retroproyección. Esta consiste en, dada una línea $L_{r,\theta}$ en particular, asignar el valor $p_\theta(r)$ a todos los puntos (x, y) pertenecientes a esa línea. Esto se repite para todos los ángulos θ usando integración. La fórmula queda entonces:

$$f(x, y) \approx \int_0^\pi p_\theta(x \cos \theta + y \sin \theta) d\theta$$

Sin embargo, la imagen obtenida con este procedimiento usualmente resulta bastante difusa, como se discutió en la sección 2.1.5 (ver figura 2.5 (b)) por lo que no es usada en la práctica. En lugar de ello, se utiliza la retroproyección filtrada.

2.1.6.2. Retroproyección filtrada

Conocida en inglés por sus siglas FBP (filtered backprojection), consiste en hacer retroproyección después de aplicar un filtro en el dominio de la frecuencia, como veremos a continuación.

Partiendo de la forma polar de Fourier (2.4), y usando el teorema de proyección (2.2 y 2.3), tenemos que

$$\begin{aligned}
 f(x, y) &= \int_0^\pi \int_{-\infty}^{+\infty} F(k \cos \theta, k \sin \theta) |k| e^{2\pi i(x k \cos \theta + y k \sin \theta)} dk d\theta \\
 &= \int_0^\pi \int_{-\infty}^{+\infty} F(u, v) |k| e^{2\pi i(x k \cos \theta + y k \sin \theta)} dk d\theta \\
 &= \int_0^\pi \int_{-\infty}^{+\infty} P(k, \theta) |k| e^{2\pi i(x k \cos \theta + y k \sin \theta)} dk d\theta \\
 &= \int_0^\pi \int_{-\infty}^{+\infty} P(k, \theta) |k| e^{2\pi i k r} dk d\theta
 \end{aligned}$$

con

$$r = x \cos \theta + y \sin \theta$$

Sea ahora

$$P^*(k, \theta) = P(k, \theta) \cdot |k|$$

y

$$p_\theta^*(r) = \int_{-\infty}^{\infty} P^*(k, \theta) e^{2\pi i k r} dk$$

Luego, tenemos que nuestra función $f(x, y)$ puede ser reconstruída con:

$$f(x, y) = \int_0^\pi p_\theta^*(r) d\theta$$

Como se discutió en la sección 2.1.5, en la práctica se acopla el filtro Ram-Lak con otro filtro pasabajos. Sea $Q(k)$ uno de esos filtros pasabajos (Hamming, Hann, Butterworth, o cualquier otro). Entonces debemos definir una nueva función:

$$\bar{P}(k, \theta) = P(k, \theta) \cdot |k| \cdot Q(k) \quad (2.8)$$

ahora tenemos que

$$\bar{p}_\theta(r) = \int_{-\infty}^{\infty} \bar{P}(k, \theta) e^{2\pi i k r} dk$$

y finalmente la función original se reconstruye así:

$$\begin{aligned} f(x, y) &= \int_0^\pi \bar{p}_\theta(r) d\theta \\ &= \int_0^\pi \bar{p}_\theta(x \cos \theta + y \sin \theta) d\theta \end{aligned}$$

2.1.7. Reconstrucción tomográfica con haz de abanico

Consideremos el haz con forma de abanico mostrado en la figura 2.7. La fuente de haces corresponde al punto F . El detector en este caso, es recto, y corresponde al segmento R_1 . En él, se detecta la señal (en rojo). El ángulo β es el ángulo central del haz (en celeste, segmento que pasa por FOG). Tomemos arbitrariamente un punto E , cuyas coordenadas cartesianas son (x, y) y coordenadas polares (r, ϕ) . El ángulo γ es el desplazamiento angular desde el eje central del haz hasta el segmento que pasa por el punto de interés E .

Sea $R_{\beta\gamma}$ el valor de la señal en el punto E proyectado a lo largo de este haz que corresponde al segmento $FCAE$. Para simplificar el desarrollo matemático, vamos a considerar el detector virtual que corresponde al segmento celeste R_2 , en lugar de utilizar R_1 .

A partir de esto, el valor de señal que nos interesa será representado por $R_\beta(s)$, es decir, el valor proyectado que pasa por E con un haz cuyo eje central está rotado un ángulo β y distancia s en el detector R_2 .

Recordemos ahora la relación definida en el capítulo anterior:

$$t = D \sin \gamma \quad \theta = \beta + \gamma$$

la cual se puede reescribir para este caso como:

$$t = \frac{s D}{\sqrt{D^2 + s^2}} \quad \theta = \beta + \tan^{-1}\left(\frac{s}{D}\right) \quad (2.9)$$

Recordando también la fórmula para reconstituir la imagen f con haz paralelo, el valor en el punto (x, y) es:

$$f(x, y) = \frac{1}{2} \int_0^{2\pi} p_\theta^*(t) d\theta \quad (2.10)$$

donde

$$p_{\theta}^*(t) = \int_{-\infty}^{\infty} p_{\theta}(t') q(t - t') dt' \quad (2.11)$$

y

$$t = x \cos \theta + y \sin \theta \quad (2.12)$$

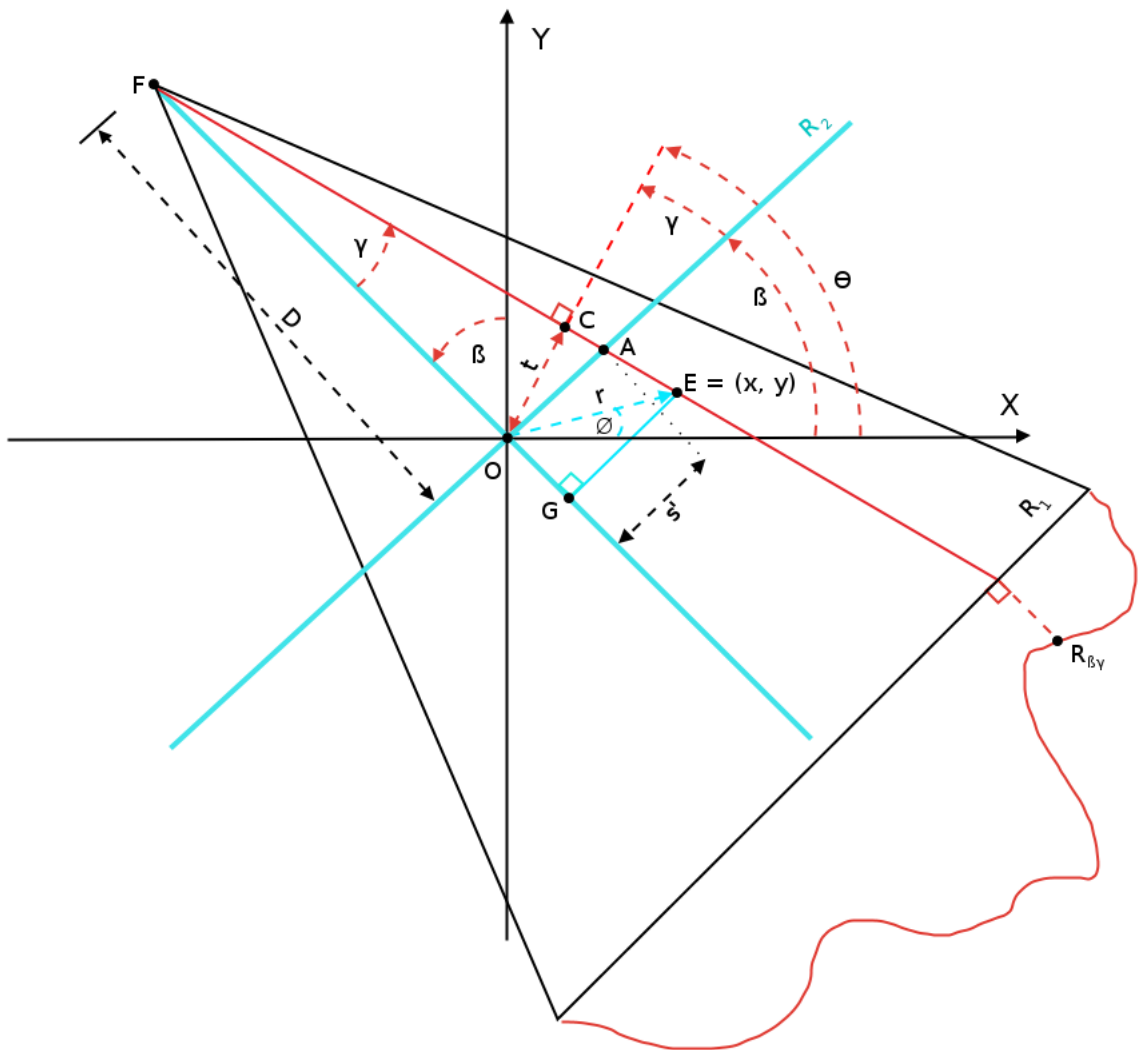


Figura 2.7: Haz de abanico con detector plano.

Expresamos ahora la función anterior en términos de coordenadas polares, r y ϕ . Teniendo que

$$x = r \cos \phi$$

$$y = r \sin \phi$$

sea $g(r, \phi) = f(r \cos \phi, r \sin \phi)$. Ahora t nos queda en términos de las nuevas variables:

$$t = r \cos \phi \cos \theta + r \sin \phi \sin \theta$$

sabiendo que $\cos(a - b) = \cos a \cos b + \sin a \sin b$, nos queda

$$t = r \cos(\theta - \phi) \quad (2.13)$$

combinando 2.10, 2.11 y 2.13, la función g queda:

$$g(r, \phi) = \frac{1}{2} \int_0^{2\pi} \int_{-t_m}^{t_m} p_\theta(t') q(r \cos(\theta - \phi) - t') dt' d\theta$$

luego, usando 2.9, tenemos que

$$g(r, \phi) = \frac{1}{2} \int_0^{2\pi} \int_{-s_m}^{s_m} p_{\beta+\gamma} \left(\frac{sD}{\sqrt{D^2 + s^2}} \right) q \left[r \cos \left(\beta + \tan^{-1} \left(\frac{s}{D} \right) - \phi \right) - \frac{sD}{\sqrt{D^2 + s^2}} \right] \frac{D^3}{\sqrt{(D^2 + s^2)^3}} ds d\beta$$

donde hemos reemplazado las variables t' y θ por s y β usando

$$dt' d\theta = \frac{D^3}{\sqrt{(D^2 + s^2)^3}} ds d\beta$$

El valor

$$p_{\beta+\gamma} \left(\frac{sD}{\sqrt{D^2 + s^2}} \right)$$

es simplemente el valor $R_\beta(s)$ de nuestro detector virtual R_2 . Nos queda entonces

$$g(r, \phi) = \frac{1}{2} \int_0^{2\pi} \int_{-s_m}^{s_m} R_\beta(s) q \left\{ r \cos \left[(\beta - \phi) + \tan^{-1} \left(\frac{s}{D} \right) \right] - \frac{sD}{\sqrt{D^2 + s^2}} \right\} \frac{D^3}{\sqrt{(D^2 + s^2)^3}} ds d\beta \quad (2.14)$$

Ahora trabajemos en el argumento del kernel q . Éste, sabiendo que $\cos(a + b) = \cos a \cos b - \sin a \sin b$, nos queda

$$r \left\{ \cos(\beta - \phi) \cos \left[\tan^{-1} \left(\frac{s}{D} \right) \right] - \sin(\beta - \phi) \sin \left[\tan^{-1} \left(\frac{s}{D} \right) \right] \right\} - \frac{sD}{\sqrt{D^2 + s^2}} \quad (2.15)$$

Usando ahora las siguientes igualdades trigonométricas:

$$\cos(\tan^{-1}(a)) = \frac{1}{\sqrt{1+a^2}} \quad \sin(\tan^{-1}(a)) = \frac{a}{\sqrt{1+a^2}}$$

la expresión 2.15 cambia a:

$$\frac{r \cos(\beta - \phi)}{\sqrt{1 + \left(\frac{s}{D}\right)^2}} - \frac{(s/D) r \sin(\beta - \phi)}{\sqrt{1 + \left(\frac{s}{D}\right)^2}} - \frac{sD}{\sqrt{D^2 + s^2}}$$

usando álgebra:

$$\frac{r D \cos(\beta - \phi)}{\sqrt{D^2 + s^2}} - \frac{s r \sin(\beta - \phi)}{\sqrt{D^2 + s^2}} - \frac{sD}{\sqrt{D^2 + s^2}}$$

quedando finalmente:

$$r \cos(\beta - \phi) \frac{D}{\sqrt{D^2 + s^2}} - (D + r \sin(\beta - \phi)) \frac{s}{\sqrt{D^2 + s^2}} \quad (2.16)$$

Se define ahora la cantidad U , que es el cociente entre la proyección del punto $E = (x, y)$ en el eje central del haz y el valor D , que es la distancia entre la fuente y el origen O :

$$\begin{aligned} U &= \frac{\overline{FG}}{D} \\ &= \frac{\overline{FO} + \overline{OG}}{D} \\ &= \frac{D + r \sin(\beta - \phi)}{D} \end{aligned} \quad (2.17)$$

Ahora definimos la distancia s' que es la distancia del punto A al eje central del haz (segmento \overline{OA}), con la cual se verifica la siguiente relación:

$$\frac{s'}{\overline{FO}} = \frac{\overline{EG}}{\overline{FG}}$$

el cual equivale a

$$s' = \frac{D}{D + r \sin(\beta - \phi)} r \cos(\beta - \phi)$$

o lo que es lo mismo

$$s' U = r \cos(\beta - \phi) \quad (2.18)$$

Luego, el argumento del kernel q se puede escribir a partir de 2.16, 2.17 y 2.18, quedando:

$$\frac{s' UD}{\sqrt{D^2 + s^2}} - \frac{s UD}{\sqrt{D^2 + s^2}} = (s' - s) \frac{UD}{\sqrt{D^2 + s^2}} \quad (2.19)$$

de 2.14 y 2.19 tenemos:

$$g(r, \phi) = \frac{1}{2} \int_0^{2\pi} \int_{-s_m}^{s_m} R_\beta(s) q \left[(s' - s) \frac{UD}{\sqrt{D^2 + s^2}} \right] \frac{D^3}{\sqrt{(D^2 + s^2)^3}} ds d\beta \quad (2.20)$$

el kernel q es el resultado de hacer la inversa de Fourier del filtro $|w|$ en el dominio de frecuencia:

$$q(t) = \int_{-\infty}^{\infty} |w| e^{j2\pi wt} dw$$

por lo tanto:

$$q \left[(s' - s) \frac{UD}{\sqrt{D^2 + s^2}} \right] = \int_{-\infty}^{\infty} |w| e^{j2\pi w (s' - s) (UD/\sqrt{D^2 + s^2})} dw$$

definiendo

$$w' = w \frac{UD}{\sqrt{D^2 + s^2}}$$

con

$$dw = \frac{\sqrt{D^2 + s^2}}{UD} dw'$$

nos queda

$$\begin{aligned}
q \left[(s' - s) \frac{UD}{\sqrt{D^2 + s^2}} \right] &= \frac{D^2 + s^2}{U^2 D^2} \int_{-\infty}^{\infty} |w'| e^{j2\pi (s' - s) w'} dw' \\
&= \frac{D^2 + s^2}{U^2 D^2} q(s' - s)
\end{aligned} \tag{2.21}$$

Haciendo la substitución de 2.21 en 2.20:

$$\begin{aligned}
g(r, \phi) &= \frac{1}{2} \int_0^{2\pi} \int_{-s_m}^{s_m} R_\beta(s) \frac{D^2 + s^2}{U^2 D^2} q(s' - s) \frac{D^3}{\sqrt{(D^2 + s^2)^3}} ds d\beta \\
&= \frac{1}{2} \int_0^{2\pi} \frac{D}{U^2} \int_{-s_m}^{s_m} \frac{R_\beta(s)}{\sqrt{D^2 + s^2}} q(s' - s) ds d\beta
\end{aligned} \tag{2.22}$$

En general, al igual que hicimos en los capítulos anteriores, durante la reconstrucción vamos a necesitar usar un filtro adicional que suavice al kernel q . Sea q' el kernel del filtro pasabajos. Entonces la forma final de la ecuación de reconstrucción para el punto ubicado en las coordenadas polares (r, θ) es:

$$g(r, \phi) = \frac{1}{2} \int_0^{2\pi} \frac{D}{U^2} (R'_\beta \star k)(s') d\beta \tag{2.23}$$

donde

$$R'_\beta(s) = \frac{R_\beta(s)}{\sqrt{D^2 + s^2}} \quad k(s) = q(s) \star q'(s)$$

$$U = \frac{D + r \sin(\beta - \phi)}{D}$$

2.1.8. Reconstrucción tomográfica con haz cónico

Para desarrollar correctamente el modelo matemático asociado al algoritmo de reconstrucción tomográfica de haz cónico, debemos partir primero de la ecuación usada para el haz de abanico, el cual es un caso particular del primero.

Como se aprecia en la figura 2.8, partiendo del sistema de coordenadas original, X, Y, Z , realizamos una rotación con un ángulo β , dando origen al sistema de coordenadas definido en los ejes A, B y C . En este caso, los ejes Y y B

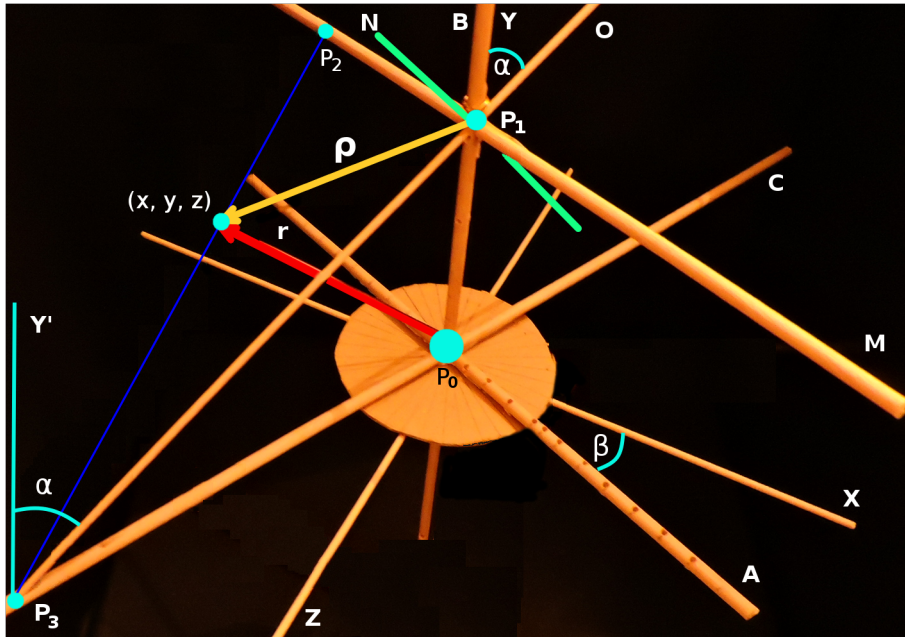


Figura 2.8: Sistemas de coordenadas para el modelo de haz cónico

coinciden.

En el punto P_3 , ubicamos la fuente de rayos X , que emite fotones hacia el plano AB , paralelo a un detector imaginario ubicado a cierta distancia del origen (punto P_0) y perpendicular al eje C . La muestra u objeto que está siendo irradiada se ubica en alguna posición del espacio entre la fuente de rayos y el detector. Sea (x, y, z) un punto de ese objeto. Nos interesa saber, por supuesto, el nivel de atenuación de ese objeto en ese punto. Dicho punto es atravesado por el rayo indicado en color azul, cuyo segmento se define con los puntos P_3 y P_2 , siendo P_2 la intersección entre el rayo azul y el plano AB .

Ahora tomamos el eje A y lo desplazamos a través del eje B hasta intersectar el punto P_2 , generando el eje M . Tenemos un nuevo punto, P_1 , que junto al punto P_3 definen el eje O . Además, definimos el eje N , perpendicular a ambos ejes M y O . Como consecuencia tenemos definido un nuevo sistema de coordenadas a partir de los ejes M, N, O . En este nuevo sistema de coordenadas definimos el vector ρ (en color naranja), que indica la posición del punto (x, y, z) para dicho sistema. El mismo punto puede ser identificado con el vector r (en color rojo), definido con respecto al sistema de coordenadas de los ejes A, B y C .

Ahora definimos los siguientes segmentos:

$$\begin{aligned}\gamma &= \overline{P_0P_1} \\ D &= \overline{P_0P_3} \\ E &= \overline{P_1P_3} \\ s' &= \overline{P_1P_2}\end{aligned}$$

Por último, llamemos \hat{x} , \hat{y} , \hat{z} , \hat{a} , \hat{b} , \hat{c} , \hat{m} , \hat{n} y \hat{o} , a los versores ubicados en los ejes X , Y , Z , A , B , C , M , N y O , respectivamente.

La idea es entonces usar el modelo de haz de abanico sobre el sistema de coordenadas M , N , O .

En particular, cuando $\gamma = 0$, es decir, cuando el eje M coincide con el eje A , los vectores ρ y r coinciden y podemos escribir el valor de atenuación del punto (x, y, z) , el cual llamamos $h(\rho)$, como:

$$h(\rho) = \frac{1}{2} \int_0^{2\pi} \frac{D}{U^2} (R'_\beta \star k)(s') d\beta$$

$$R'_\beta(s) = \frac{R_\beta(s, \gamma = 0)}{\sqrt{D^2 + s^2}} \quad k(s) = q(s) \star q'(s)$$

$$U = \frac{D + \rho \cdot \hat{o}}{D}$$

$$s' = \frac{D(\rho \cdot \hat{m})}{D + \rho \cdot \hat{o}}$$

Sea α el ángulo entre los vectores \hat{y} y \hat{o} . Si definimos un nuevo eje, llamado Y' , paralelo al eje Y que intersecta a los ejes C y O , el mismo ángulo α se forma entre Y' y O . Tenemos que

$$|\hat{y} \times \hat{o}| = |\hat{y}| \cdot |\hat{o}| \cdot \sin \alpha = \sin \alpha = \frac{D}{\sqrt{D^2 + \gamma^2}} \quad (2.24)$$

Ahora, una rotación infinitesimal $\delta\beta$ alrededor del eje Y , corresponde a una rotación $\delta\eta$ alrededor del eje N . Teniendo en cuenta 2.24 tenemos

$$\begin{aligned}\delta\hat{o} &= \delta\beta \hat{y} \times \hat{o} \\ &= \delta\beta \frac{D}{\sqrt{D^2 + \gamma^2}} \hat{m}\end{aligned} \quad (2.25)$$

Por otro lado, también se cumple que

$$\begin{aligned}\delta\hat{\mathbf{o}} &= \delta\eta \hat{\mathbf{n}} \times \hat{\mathbf{o}} \\ &= \delta\eta \hat{\mathbf{m}}\end{aligned}\quad (2.26)$$

igualando 2.25 y 2.26 nos queda:

$$\delta\eta = \frac{\delta\beta D}{\sqrt{D^2 + \gamma^2}} \quad (2.27)$$

Como la nueva distancia desde la fuente al origen es:

$$E = \sqrt{D^2 + \gamma^2}$$

entonces 2.27 se puede escribir como

$$\delta\eta = \frac{D}{E} \delta\beta \quad (2.28)$$

Ahora consideremos $\gamma \neq 0$. Se puede verificar que el vector \mathbf{r} se puede expresar como

$$\mathbf{r} = \boldsymbol{\rho} + \gamma \hat{\mathbf{y}}$$

La ecuación para el haz de abanico que queda a una altura γ en el eje Y es:

$$h(\mathbf{r}) = h(\boldsymbol{\rho} + \gamma \hat{\mathbf{y}}) = \frac{1}{2} \int_0^{2\pi} \frac{E}{V^2} (S'_\eta \star k)(s') d\eta$$

$$S'_\eta(s, \gamma) = \frac{S_\eta(s, \gamma)}{\sqrt{E^2 + s^2}} \quad k(s) = q(s) \star q'(s)$$

$$V = \frac{E + \boldsymbol{\rho} \cdot \hat{\mathbf{o}}}{E}$$

$$s' = \frac{E(\boldsymbol{\rho} \cdot \hat{\mathbf{m}})}{E + \boldsymbol{\rho} \cdot \hat{\mathbf{o}}}$$

Por otro lado, las tres igualdades

$$\boldsymbol{\rho} \cdot \hat{\mathbf{o}} = \frac{E}{D} \mathbf{r} \cdot \hat{\mathbf{c}}$$

$$\sqrt{E^2 + s^2} = \sqrt{D^2 + \gamma^2 + s^2} \quad (2.29)$$

$$\boldsymbol{\rho} \cdot \hat{\mathbf{m}} = \mathbf{r} \cdot \hat{\mathbf{a}}$$

junto con 2.28 implican que

$$h(\mathbf{r}) = \frac{1}{2} \int_0^{2\pi} \frac{E}{V^2} (S'_\beta \star k)(s', \gamma) \frac{D}{E} d\beta$$

$$S'_\beta(s, \gamma) = \frac{S_\beta(s, \gamma)}{\sqrt{D^2 + \gamma^2 + s^2}} \quad k(s, \gamma) = q(s, \gamma) \star q'(s, \gamma)$$

$$V = \frac{D + \mathbf{r} \cdot \hat{\mathbf{c}}}{D} \quad s' = \frac{\mathbf{r} \cdot \hat{\mathbf{a}}}{V} \quad \gamma = \frac{yD}{D + \mathbf{r} \cdot \hat{\mathbf{c}}}$$

Dado que $\mathbf{r} = (x, y, z)$, $\hat{\mathbf{c}} = (-\sin \beta, 0, \cos \beta)$ y $\hat{\mathbf{a}} = (\cos \beta, 0, \sin \beta)$, la forma final de las ecuaciones es la siguiente:

$$h(x, y, z) = \frac{1}{2} \int_0^{2\pi} \frac{D}{V^2} (S'_\beta \star k)(s', \gamma) d\beta$$

$$S'_\beta(s, \gamma) = \frac{S_\beta(s, \gamma)}{\sqrt{D^2 + \gamma^2 + s^2}} \quad k(s, \gamma) = q(s, \gamma) \star q'(s, \gamma)$$

$$V = \frac{D - x \sin \beta + z \cos \beta}{D} \quad s' = \frac{x \cos \beta + z \sin \beta}{V} \quad \gamma = \frac{yD}{D - x \sin \beta + z \cos \beta}$$

2.2. Implementación

En las siguientes subsecciones resumimos las implementaciones de software que llevan a cabo las simulaciones de proyecciones radiográficas, el conversor de estas proyecciones a formato estándar DICOM, y finalmente el algoritmo de reconstrucción tomográfica basado en las ecuaciones desarrolladas anteriormente.

2.2.1. *simrx* - PENELOPE

El código *simrx* es un simulador de proyecciones radiográficas, el cual utiliza un código de simulación Monte Carlo de transporte de energía llamado PENELOPE (Penetration and energy loss of Positrons and Electrons). Desarrollado en FORTRAN 77 por la Universidad de Barcelona, los módulos de PENELOPE utilizados por *simrx* son **Pengeom** y **Penelope**, ambos en color azul mostrados en la figura 2.9 En toda simulación de proyecciones radiográficas

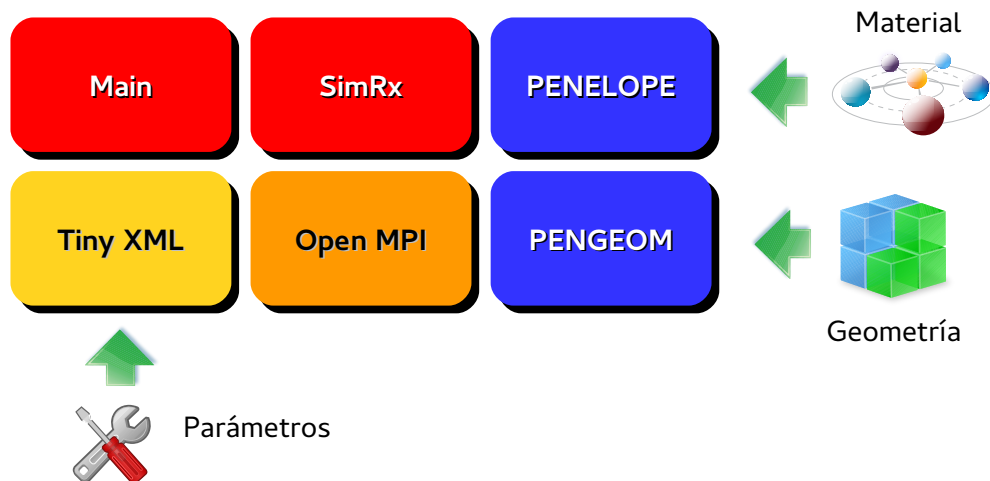


Figura 2.9: Diagrama de módulos de simrx

tendremos 3 elementos principales: la fuente de rayos X, el detector, y el objeto de estudio. Este último se define en un archivo usando superficies cuádricas, las cuales son interpretadas por el módulo Pengeom. El objeto de estudio está compuesto por distintos materiales, que son interpretados por el módulo Penelope. Este módulo también contiene el generador de números aleatorios, además de tener implementado la simulación de los distintos fenómenos físicos que se producen en la interacción entre radiación y materia, como el efecto fotoeléctrico, Compton, producción de pares, etc.

Nuestra implementación consistió en reutilizar estos dos módulos y crear otros nuevos encargados de simular el proceso entero de ir rotando la muestra para obtener proyecciones radiográficas a distintos ángulos. Entre ellos, **SimRx**, código en FORTRAN 90, es el encargado de simular fotones y dispararlos hacia el objeto de estudio. Coordina las llamadas a las distintas funciones del módulo Penelope y orquesta la simulación completa.

Como las simulaciones de este tipo son computacionalmente intensivas, se decidió ejecutar la simulación en un entorno HPC (por sus siglas en inglés: High Performance Computing), que consiste en uno o varios clusters de procesadores para

trabajar con computación distribuída, reduciendo el tiempo de cómputo. Dado entonces un conjunto de procesadores, se simula cada proyección radiográfica, una en cada procesador. El módulo encargado de paralelizar el cómputo entre distintos procesadores es **Main**, implementado en C++. Este utiliza la librería **OpenMPI** (implementación open source del protocolo Message Passing Interface), desarrollada para distribuir y ejecutar procesos en paralelo en un cluster de procesadores.

Finalmente, el módulo **TinyXML** es el encargado de leer el archivo de configuración de *simrx*, el cual tiene formato XML y dicta el funcionamiento del simulador, e incluye los siguientes parámetros:

- Tipo de haz (cónico o paralelo), energía de los fotones, parámetros físicos del haz (distancias, ángulos, etc)
- Nombres de los archivos de materiales y geometría
- Angulo de rotación total del objeto, si es de haz cónico normalmente de 360 grados, y cantidad de proyecciones a realizar.
- Características y distancia del detector a la fuente de rayos.
- Parámetros específicos pasados al módulo Penelope (ver [26]).

Las imágenes radiográficas obtenidas se guardan en formato crudo (RAW). Estos se deben pasar al formato estándar usado en medicina llamado DICOM.

2.2.2. DICOM

Como se describe en [25], el estándar DICOM no solamente define el formato y estructura de un archivo con datos de imágenes médicas, sino que además establece un protocolo de comunicación entre distintos dispositivos (tomógrafos, computadoras centrales, servidores de archivos, etc), para el intercambio correcto y eficiente de datos entre ellos.

En nuestro caso, básicamente nos interesa poder tomar las imágenes en formato RAW que genera el simulador *simrx* y guardarlas en archivos DICOM. Para ello, se desarrolló una herramienta de conversión, llamada **LIIFAMIRx Raw to DICOM conversor**. En la figura 2.10 se muestra una captura de pantalla del conversor en pleno funcionamiento, luego de haber elegido el conjunto de datos a convertir.

El funcionamiento es simple: se elige el directorio en disco donde se encuentran los archivos generados por *simrx*, que tienen extensión **srx**, y el conversor



Figura 2.10: Captura de pantalla del conversor RAW-DICOM

automáticamente extrae información de los parámetros usados en la simulación por medio del archivo de parámetros que se usa en simrx.

En la captura de pantalla vemos algunos de esos parámetros, como ser las características del haz, del detector, distancias, geometría, resolución de la imagen final, energía del haz de fotones en eV, etc.

Este conversor fué implementado en Qt/C++.

De manera muy resumida, podemos decir que el formato DICOM consta de dos partes principales: un encabezado y los datos de imagen propiamente dichos. En el encabezado se guardan los parámetros mencionados anteriormente, como así también datos específicos explicados en [25], como ser si la imagen es positiva o negativa, cantidad de bits usados para codificar el nivel de grises en la imagen, etc.

La imagen, ubicada al final del archivo, puede guardarse nuevamente en formato

RAW, o usar algún formato de compresión de imágenes, como ser JPEG.

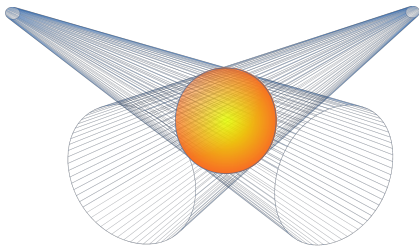
Para la implementación del conversor, se utilizó la librería open source **DCMTK (DICOM Toolkit)**.

2.2.3. Algoritmos de reconstrucción tomográfica

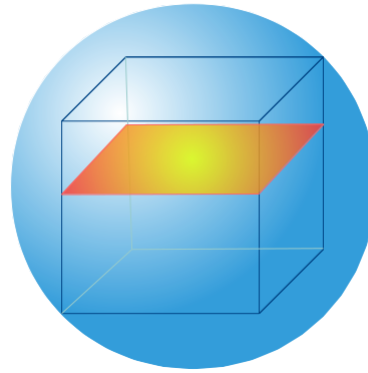
Una vez que simulamos las proyecciones radiográficas con *simrx* y convertimos los archivos RAW a DICOM, nos queda reconstruir la tomografía. Implementamos entonces el algoritmo FDK, basado en el modelo matemático desarrollado en la sección 2.1.8, utilizando **MATLAB®**. Este software puede interpretar archivos DICOM directamente, y de esta manera obtener todos los parámetros físicos y de simulación guardados en el encabezado de estos archivos. La función de reconstrucción tomográfica que implementamos utiliza los siguientes parámetros de entrada:

- Número de slice a reconstruir
- Tipo de filtro a utilizar y parámetros del filtro (ver sección 2.1.5).
- Tipo de interpolación: vecino mas cercano o lineal

Durante la reconstrucción tomográfica, el algoritmo implementado tiene en consideración solamente los puntos con los que se cuenta la totalidad de los datos, es decir, si en un rango de ángulos, tenemos datos de intensidad para un punto, pero para ciertos ángulos esta información no está presente, entonces la atenuación para ese punto no se computa. Debido a que la fuente de rayos X va girando alrededor del objeto de estudio, y que el haz tiene geometría cónica, nos queda entonces que los puntos candidatos a computar se encuentran en una esfera (figura 2.11 a). La implementación que realizamos reduce aún mas el conjunto de puntos a computar acotándose al cubo circunscrito en la esfera anterior (figura 2.11 b). A ese cubo se lo divide en capas, que son los slices. Una ejecución del algoritmo reconstruye uno o un conjunto contiguo de estos slices.



(a) Esfera definida por la intersección de conos



(b) Los puntos que se computan son los que pertenecen a un slice dentro del cubo

Figura 2.11: Puntos considerados para la reconstrucción

Capítulo 3

Resultados

3.1. *simrx*

En la figura 3.1 se pueden apreciar 4 imágenes que corresponden a proyecciones a distintos ángulos resultantes de una simulación de *simrx* consistente en el siguiente escenario: un cubo de agua de 6 cm de lado, un cilindro de silicio de 2 cm de diámetro y 2 cm de largo, ubicado dentro del cubo de agua, desplazado del centro de los tres ejes. Las imágenes son positivas, es decir, más blanco indica más fotones, más oscuro, menos fotones llegaron al detector. La simulación se realizó con un haz monocromático de geometría cónica con 28 grados de apertura, de 50 keV, cuya fuente está ubicada a 20 cm del origen del sistema de coordenadas, y el detector está ubicado a 19 cm del mismo centro, en dirección contraria, encerrando ambos al cubo y cilindro. El barrido se hizo tomando 360 proyecciones separadas a 1 grado cada una. Se generaron 10^9 fotones para este ejemplo particular.

Se realizó una medición del coeficiente de atenuación lineal de ambos materiales, agua y silicio, tomando sus valores de intensidad obtenidos en el detector virtual de *simrx*, y los valores de intensidad inicial provenientes de la fuente de rayos X, utilizando luego 2.1. Se compararon estos coeficientes con los valores teóricos para ese nivel de energía (50keV). En la tabla 3.1 se realiza la comparación de valores, dando como resultado un error relativo menor al 2%. Parte de este error se explica por el scattering: al tener radiación dispersa, se cuentan más fotones de los que deberían llegar, por lo que da la apariencia de tener menos atenuación en el material, lo cual es consistente con lo que se observa en la tabla 3.1, en la cual los valores de atenuación en la simulación son menores a los teóricos.

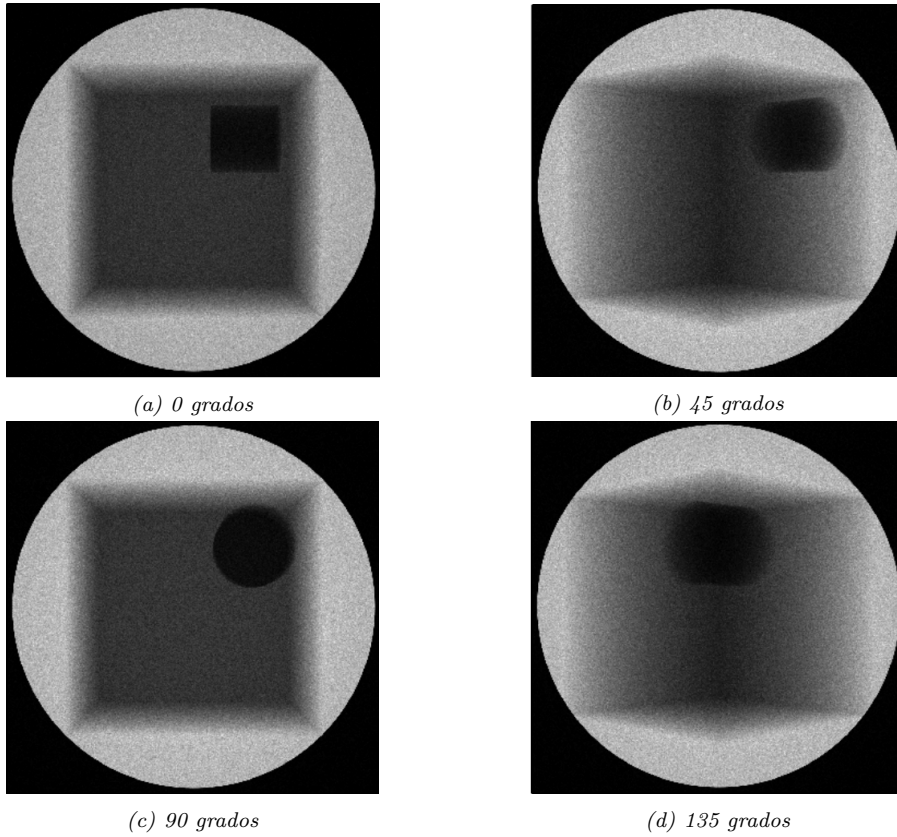


Figura 3.1: Imágenes obtenidas con *simrx*

	Agua	Silicio
Medido	0.2245 cm ⁻¹	1.0045 cm ⁻¹
Teórico	0.2269 cm ⁻¹	1.0208 cm ⁻¹

Tabla 3.1: Valores de atenuación lineal, medidos vs. teóricos

3.2. Haz paralelo

3.2.1. Introducción

Se realizaron tres experimentos con haz paralelo. En el primero se trabaja con un escenario artificial generado con Matlab consistente en un simple cubo. En el segundo, se trabaja con el fantoma de Shepp-Logan en 3-D, y por último, se utiliza una simulación de una barra de silicio producida por *simrx*.

3.2.2. Reconstrucción del cubo

Este experimento consiste en generar una geometría artificial de un cubo y luego obtener 180 proyecciones generadas con haz paralelo. Esto se logra con la función **radon** de Matlab. En la figura 3.2 se observa el sinograma con tres perfiles de ejemplo, que corresponden a las proyecciones a 0, 22 y 45 grados.

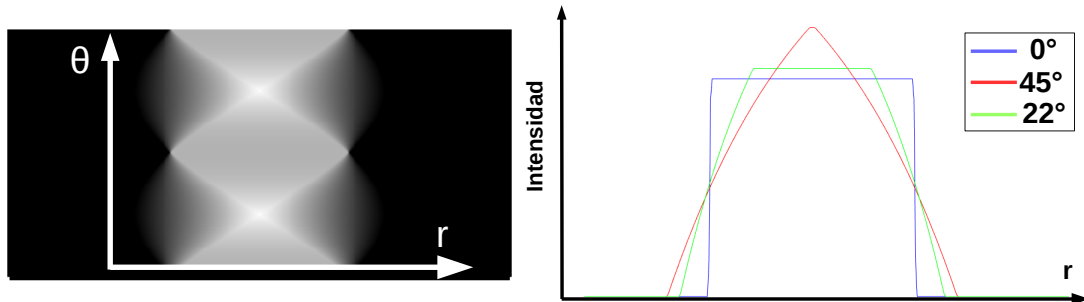


Figura 3.2: Sinograma (izq.) y tres perfiles (der.)

Se realizó la reconstrucción del slice 128. En la figura 3.3 se observan la imagen original y tres reconstrucciones, una con filtrado Ram-Lak solo, otra con Ram-Lak mas Hann con corte en 0.8, y otra imagen con los mismos filtros pero con corte en 0.5.

Finalmente, en la figura 3.4 tenemos dos cortes de las imágenes generadas con Ram-Lak y con Hann de 0.5. Se observa como el filtro Hann suaviza el ruido en comparación si se usa solamente Ram-Lak durante la reconstrucción.

3.2.3. Shepp-Logan

En la figura 3.5 se observan 2 de las 180 proyecciones del fantoma Shepp-Logan generadas con la función *radon* de Matlab. La utilización de esta función genera, al igual que en el caso del fantoma del cubo, cierto grado de ruido, que se ve reflejada al momento de realizar las distintas reconstrucciones. El fantoma tiene dimensiones de $256 \times 256 \times 256$ píxeles. Se realizó la reconstrucción del slice 145. En 3.6(a) se aprecia la imagen original, en (b) la reconstrucción usando el vecino mas cercano, en (c) la reconstrucción usando interpolación lineal, mientras que en (d) se agrega un filtro Hann de corte 0.8 al caso anterior. En las imágenes reconstruídas (figura 3.6) parece no haber diferencia a simple vista, pero examinando los perfiles, realizando un corte horizontal en el medio de cada imagen, se observa con claridad la diferencia en el valor de la magnitud de la relación señal ruido.

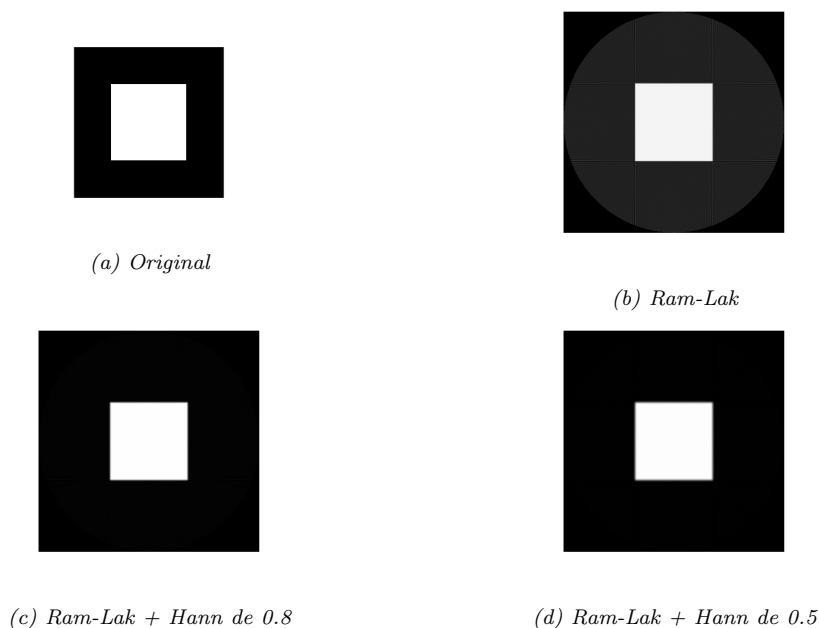


Figura 3.3: Imágenes: original y diversas reconstrucciones.

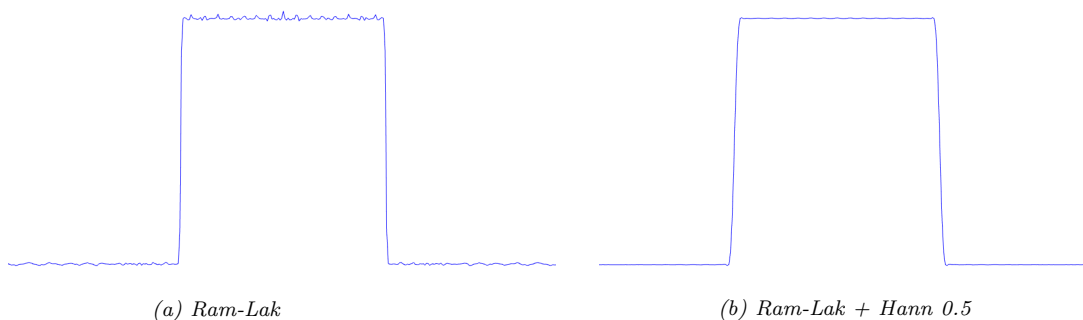


Figura 3.4: Perfiles de las imágenes 3.3 (b) y (d)

3.2.4. Barra de silicio simulada

Como ejemplo final de reconstrucción de haz paralelo, utilizamos como material de trabajo una simulación generada por *simrx* la cual consiste en una barra rectangular de silicio de 2 cm de ancho, 1 cm de profundidad y 10 cm de altura, centrada en el origen del sistema de coordenadas.

La simulación se realizó con 10^9 fotones, lo cual es una cantidad pequeña en relación a la superficie del detector de rayos ($15\text{ cm} \times 15\text{ cm}$). En este caso el resultado es una señal con un nivel alto de ruido. Esto da como resultado imágenes cuyo perfil se puede apreciar en la figura 3.8 (en color azul).

Como consecuencia, la reconstrucción de cualquier slice produce resultados con

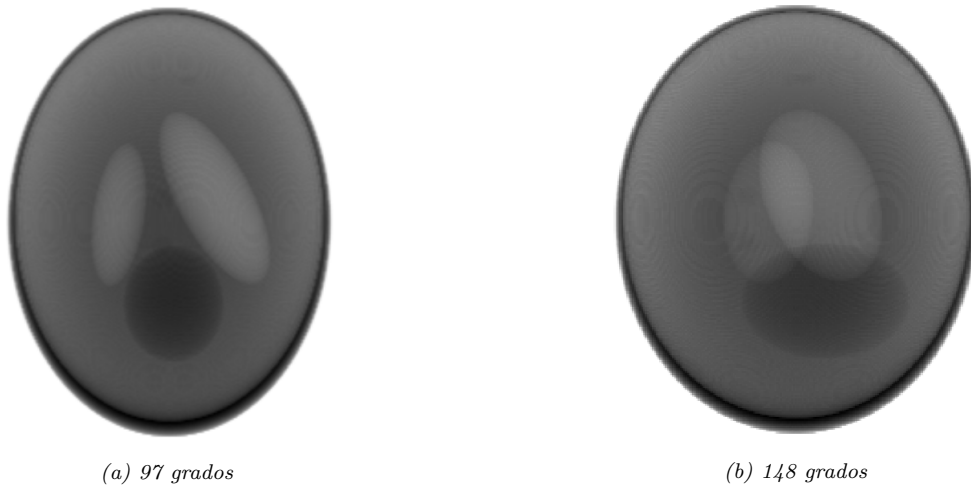


Figura 3.5: Dos proyecciones del fantoma Shepp-Logan, en negativo

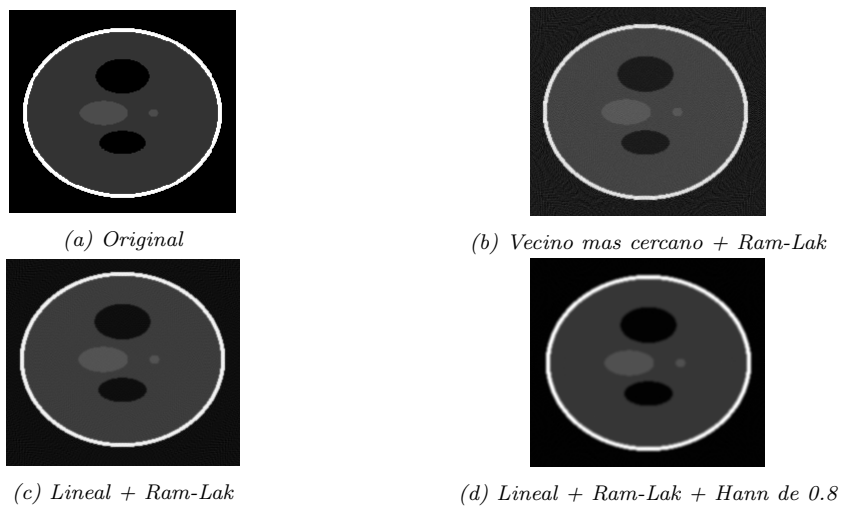
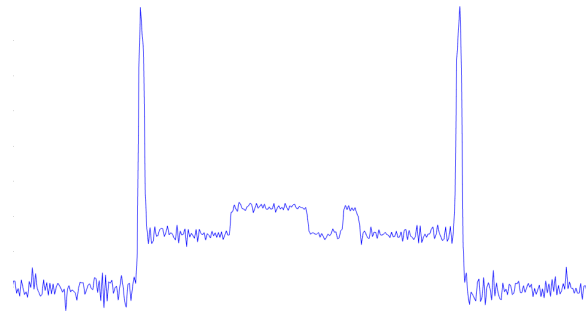


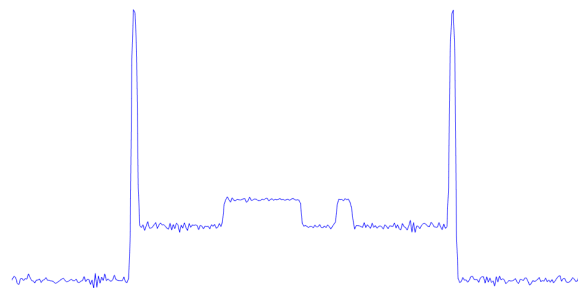
Figura 3.6: Imágenes: original y diversas reconstrucciones

ruido apreciable (señal azul en figura 3.9)

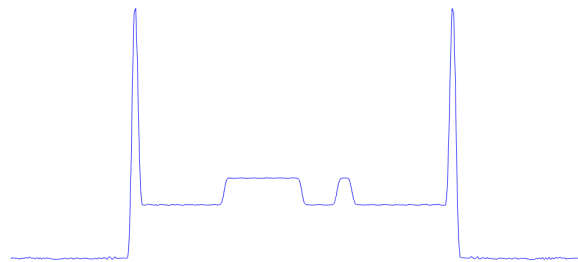
Para reducir este ruido, lo que se hizo fué filtrar las proyecciones usando wavelets Haar de nivel 5. El inconveniente es que tenemos 180 proyecciones, con lo cual resulta costoso diseñar tal cantidad de filtros. Sin embargo, podemos hacer la siguiente aproximación: ya que la proyección n -ésima “se parece” a la proyección $n + 1$ y $n - 1$, y a sus vecinos mas cercanos, dividimos el total de proyecciones en paquetes de 10, tomando la quinta proyección de cada paquete, diseñando un filtro para ella, y aplicando este filtro a todas las proyecciones del



(a) Perfil central de la figura 3.6b



(b) Perfil central de la figura 3.6c



(c) Perfil central de la figura 3.6d

Figura 3.7: Perfiles de las imágenes 3.6

bloque. Tenemos entonces un total de 18 filtros. Como ejemplo, en la figura 3.8, vemos la señal naranja que es el resultado de filtrar la señal azul.

El resultado sin usar wavelets se observa en la figura 3.10 (a). Usando el filtrado wavelet tenemos el resultado de la figura 3.10 (b), que aunque es mas uniforme y menos ruidosa, presenta sin embargo algunos artefactos bien marcados. Estos son ligeramente reducidos si usamos un filtro Butterworth de orden $n = 1$ y frecuencia de corte $wc = 0,8$, como se ve en la figura 3.10 (c).

Los perfiles centrales de las figuras 3.10 (a) y 3.10 (c), se muestran en la figura 3.8, señales azul y naranja, respectivamente.

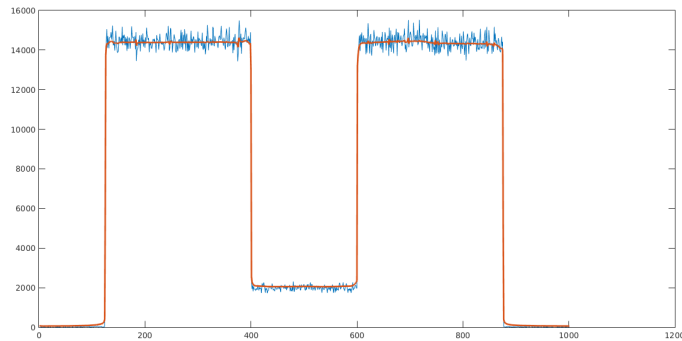


Figura 3.8: Perfiles de intensidad: original (azul), filtrado wavelet (naranja)

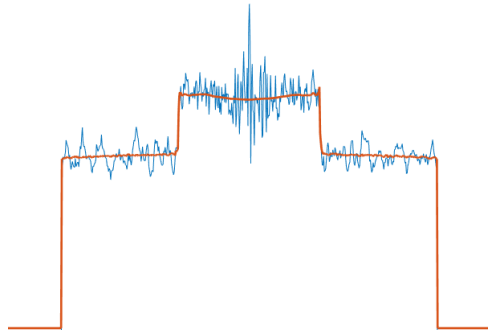


Figura 3.9: Perfiles de las imágenes reconstruidas: sin filtrado (azul), con filtrado wavelet (naranja)

3.3. Haz de abanico

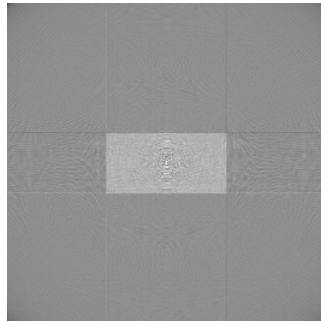
3.3.1. Introducción

Se realizó la prueba de la implementación en Matlab del algoritmo de reconstrucción con haz de abanico usando el fantoma de Shepp-Logan de tamaño 128×128 píxeles (fig. 3.11) (a).

Para obtener el sinograma resultante de aplicar un haz de abanico en un fantoma, se utiliza la función **fanbeam** de Matlab (fig. 3.11) (b).

3.3.2. Resultados

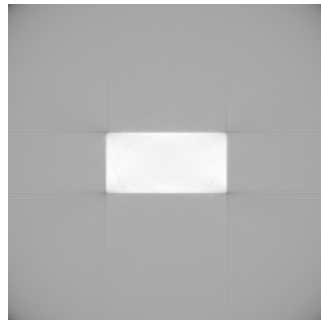
En la figura 3.12 se presentan dos reconstrucciones del fantoma anterior, una utilizando el valor de atenuación del vecino más cercano, y la otra usando interpolación lineal. Por las características de la imagen original (baja resolución y bordes con serrucho), no se utilizó el filtro pasa-bajos adicional ni se recortó la



(a) *Reconstrucción sin filtrado wavelet*



(b) *Reconstrucción con filtrado wavelet*



(c) *Reconstrucción con filtrado wavelet + Butterworth con $n = 1$ y $wc = 0,8$*

Figura 3.10: Tres reconstrucciones de la barra simulada con $simrx$

frecuencia máxima en el filtro de Ram-Lak.

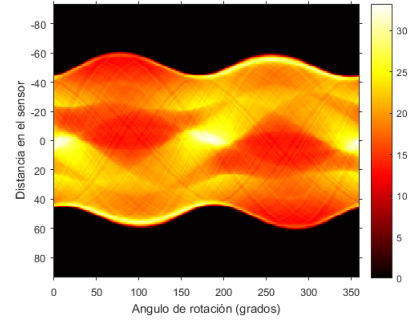
3.4. Haz cónico

3.4.1. Introducción

Para la parte mas importante del trabajo, que es la reconstrucción tomográfica de haz cónico usando la implementación del algoritmo FDK, se analizaron tres casos.

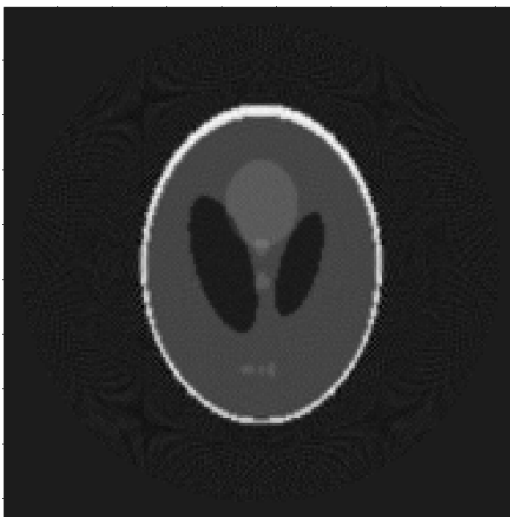


(a) Imagen original

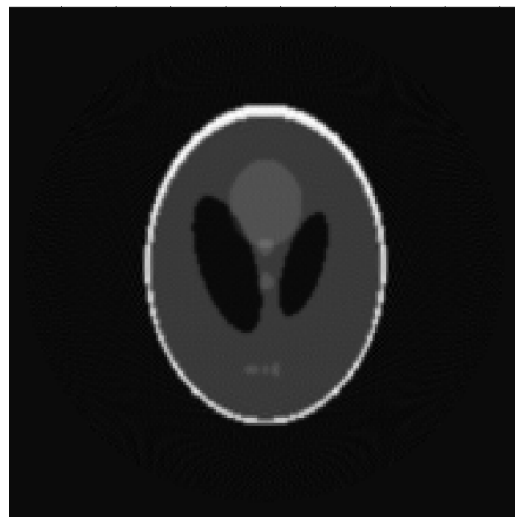


(b) Sinograma

Figura 3.11: Fantoma de Shepp-Logan y su sinograma de haz de abanico



(a) Vecino mas cercano



(b) Con interpolación lineal

Figura 3.12: Reconstrucciones del fantoma.

Para el primero, se utilizó como fantoma un cubo, cuya simplicidad es útil en la práctica para poder realizar un testeo básico del algoritmo FDK, permitiendo identificar errores de programación de manera mucho mas efectiva que si se usara un fantoma de geometría mas compleja.

Para el segundo, se utilizó, al igual que en las pruebas de haz de abanico del capítulo anterior, el fantoma de Shepp-Logan. En esta prueba, con el objeto de tener un criterio para medir la calidad de las imágenes obtenidas, se utiliza como método de comparación el valor conocido como *error relativo medio* (en inglés, mean relative error o MRE).

Por último, se realizaron reconstrucciones de una geometría simulada con *simrx*, con imágenes de dos simulaciones distintas, una con 10^9 fotones y la otra con 10^{10} fotones. La geometría consiste en un cubo de agua de 4 cm de lado en cuyo interior se encuentra un cilindro de silicio de 1 cm de diámetro y 1 cm de largo. La simulación fué realizada a 50 KeV.

3.4.2. Reconstrucción del cubo

El fantoma utilizado en esta prueba fué generada artificialmente en Matlab con un cubo de $501 \times 501 \times 501$ voxels, en donde se encuentra inmerso otro cubo concéntrico de $143 \times 143 \times 143$.

El valor asignado al cubo interno es simplemente 1, mientras que los voxels que lo rodean y que pertenecen al cubo externo tienen valor 0.

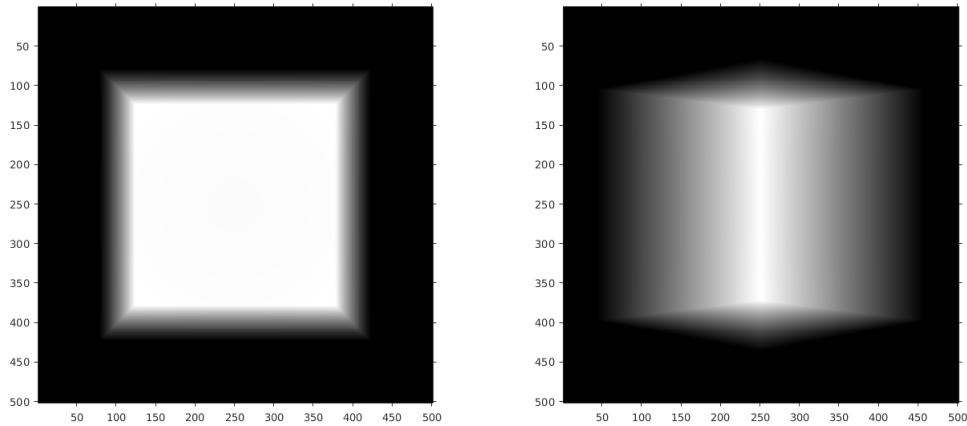
Para generar las 360 proyecciones con haz cónico a partir de este fantoma, se utilizaron scripts implementados por [18]. En la figura 3.13 se observan dos proyecciones obtenidas con estos scripts, la proyección a 0 grados y la de 45 grados. En las imágenes, las unidades están en píxeles.

En la figura 3.14(a) se ve el slice central original. En la figura (b) se muestra una reconstrucción usando nuestra implementación del FDK en Matlab con el método del vecino mas cercano, sin filtro adicional. En (c) se muestra la reconstrucción realizada con interpolación lineal, sin filtro adicional, mientras que (d) es el mismo procedimiento que (c) pero con el agregado de un filtro Hann de 0.8 de frecuencia de corte.

Como a simple vista no se distinguen las diferencias entre estas imágenes, es necesario inspeccionar los perfiles.

En la figura 3.15 (a) se muestran los perfiles centrales de los tres procedimientos de reconstrucción probados. Se aprecia, de manera cualitativa, que el nivel de ruido es menor con el método que usa interpolación lineal + el filtro Hann (señal en color naranja).

En la figura 3.15 (b) se grafican todos los perfiles superpuestos del slice central



(a) A 0 grados

(b) A 45 grados

Figura 3.13: Dos proyecciones del cubo generados con Matlab

reconstruido con el filtro Hann. De nuevo, a nivel cualitativo, se observa que el nivel de señal, que representa el nivel de atenuación del objeto estudiado, se mantiene bastante uniforme a lo largo de toda la geometría.

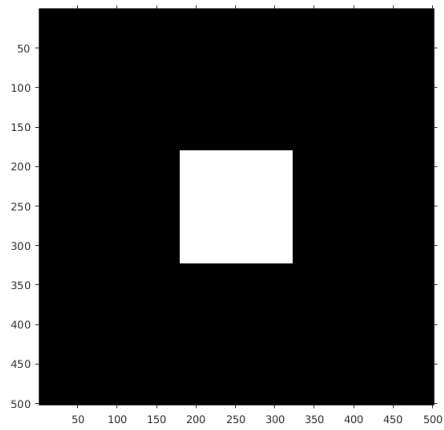
Un resultado importante es que la geometría resultante en los slices reconstruidos por esta implementación del FDK son correctas, es decir, tienen las dimensiones equivalentes con respecto al fantoma original.

Por otro lado, en este ejemplo no podemos comparar los niveles de señal ya que esta es binaria, es decir, tiene solo dos niveles. En los próximos ejemplos de Shepp-Logan y el fantoma simulado vamos a poder realizar este tipo de comparación.

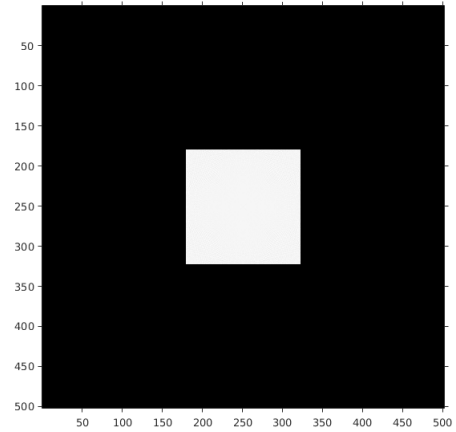
3.4.3. Reconstrucción del fantoma Shepp-Logan

Para esta prueba, se generó con Matlab un fantoma de Shepp-Logan tridimensional, de las cuales se obtuvieron 360 proyecciones usando [18]. A partir de allí se analizó la calidad de las imágenes reconstruidas con nuestra implementación del algoritmo FDK utilizando como criterio el valor RME (relative mean error, en inglés), el cual se define de esta manera:

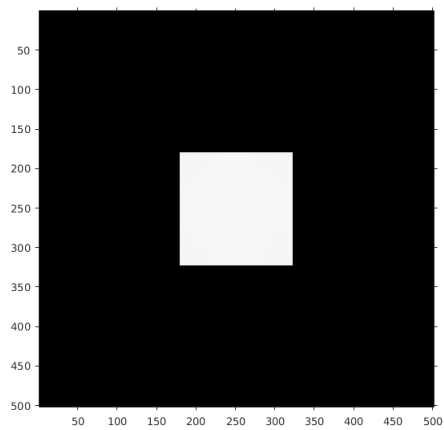
$$\text{RME} = \frac{1}{n} \sum_{x=1}^n \left| \frac{\mu'(x) - \mu_{id}(x)}{\mu_{id}(x)} \right|$$



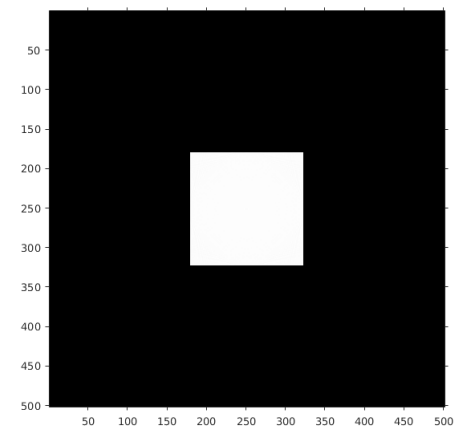
(a) Original



(b) Vecino mas cercano



(c) Interpolación lineal



(d) Interpolación lineal + Hann de 0.8

Figura 3.14: Slice central original + tres reconstrucciones

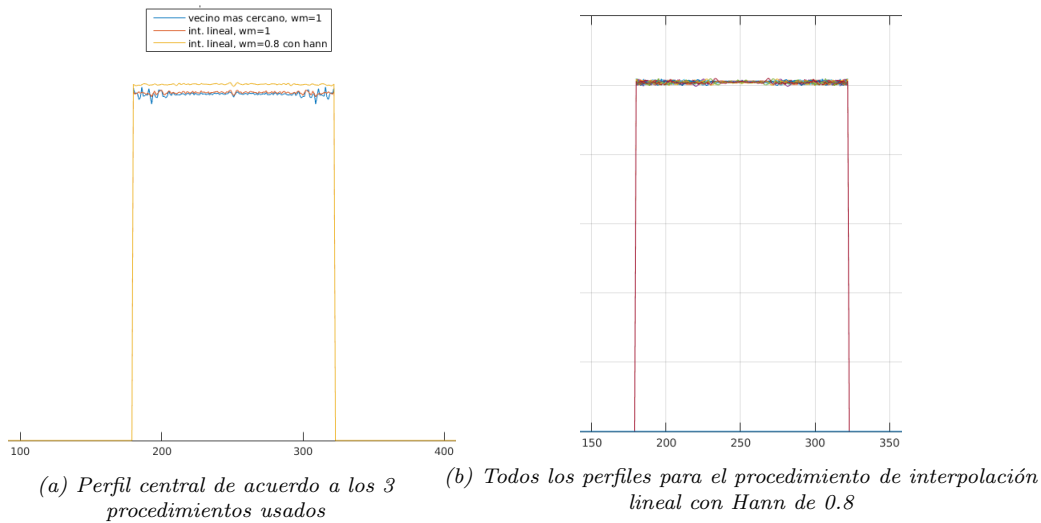


Figura 3.15: Perfiles de los slices reconstruidos

donde n es la cantidad de muestras de la señal (número de píxeles en un perfil de un slice), y $\mu_{id}(x)$ es el valor de atenuación del perfil ideal (tomado del fantoma original) en el punto x . La cantidad $\mu'(x)$ se define como:

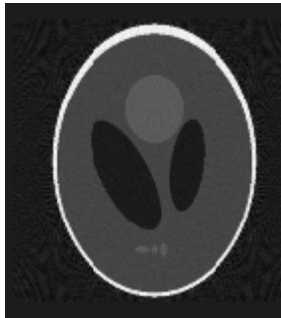
$$\mu'(x) = \mu(x) \frac{\bar{\mu}}{\bar{\mu}_{id}}$$

donde $\mu(x)$ el valor de atenuación del perfil en el píxel x tomado del slice reconstruido por el algoritmo FDK, mientras que $\bar{\mu}$ es la media de la señal reconstruida y $\bar{\mu}_{id}$ es la media del perfil ideal. La señal $\mu'(x)$ tiene el mismo valor medio que la señal ideal y es igual a $\bar{\mu}_{id}$.

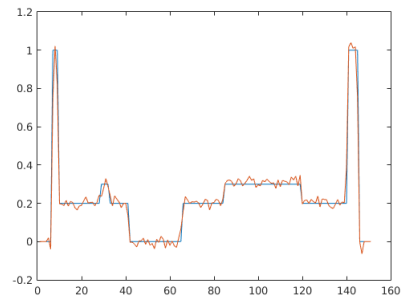
Para las pruebas se utilizó el slice central del fantoma. De la gran cantidad de combinaciones de parámetros que se especifican durante la reconstrucción, se eligieron cuatro imágenes para presentar en esta sección. En la tabla 3.2 se presentan los resultados de estas cuatro pruebas. El valor de RME mas bajo, es el que resulta ser de la reconstrucción con interpolación lineal usando un filtro Butterworth de orden $n = 1$ y frecuencia de corte $wc = 0,8$.

Prueba nro.	Selección de valor	wm	filtro	RME
1	vecino + cercano	1	-	0.019
2	interpolación lineal	1	-	0.014
3	interpolación lineal	0.8	Hann	0.013
4	interpolación lineal	1	Butterworth n=1 wc=0.8	0.012

Tabla 3.2: Valores de RME para comparar la calidad de las imágenes



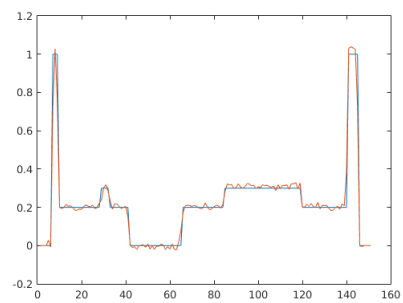
(a) Vecino mas cercano, sin filtro



(b) Perfil central de (a)



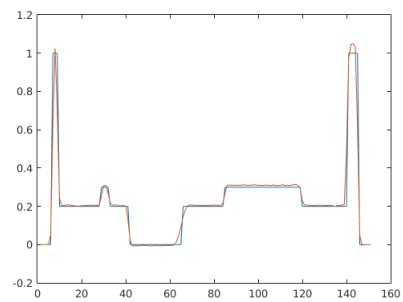
(c) Interpolación lineal sin filtro



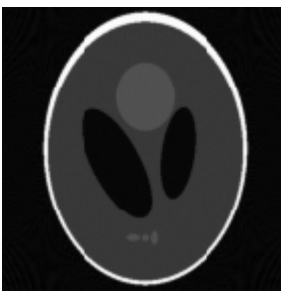
(d) Perfil central de (c)



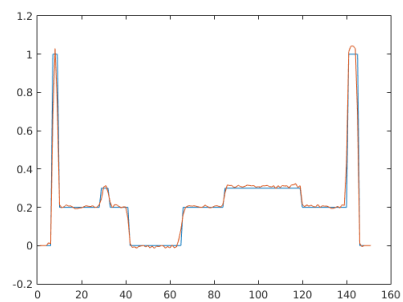
(e) Interpolación lineal + Hann de 0.8



(f) Perfil central de (e)



(g) Interpolación lineal + Butterworth con $n=1$ y $wc=0.8$



(h) Perfil central de (g)

Figura 3.16: Reconstrucciones y sus perfiles centrales verticales

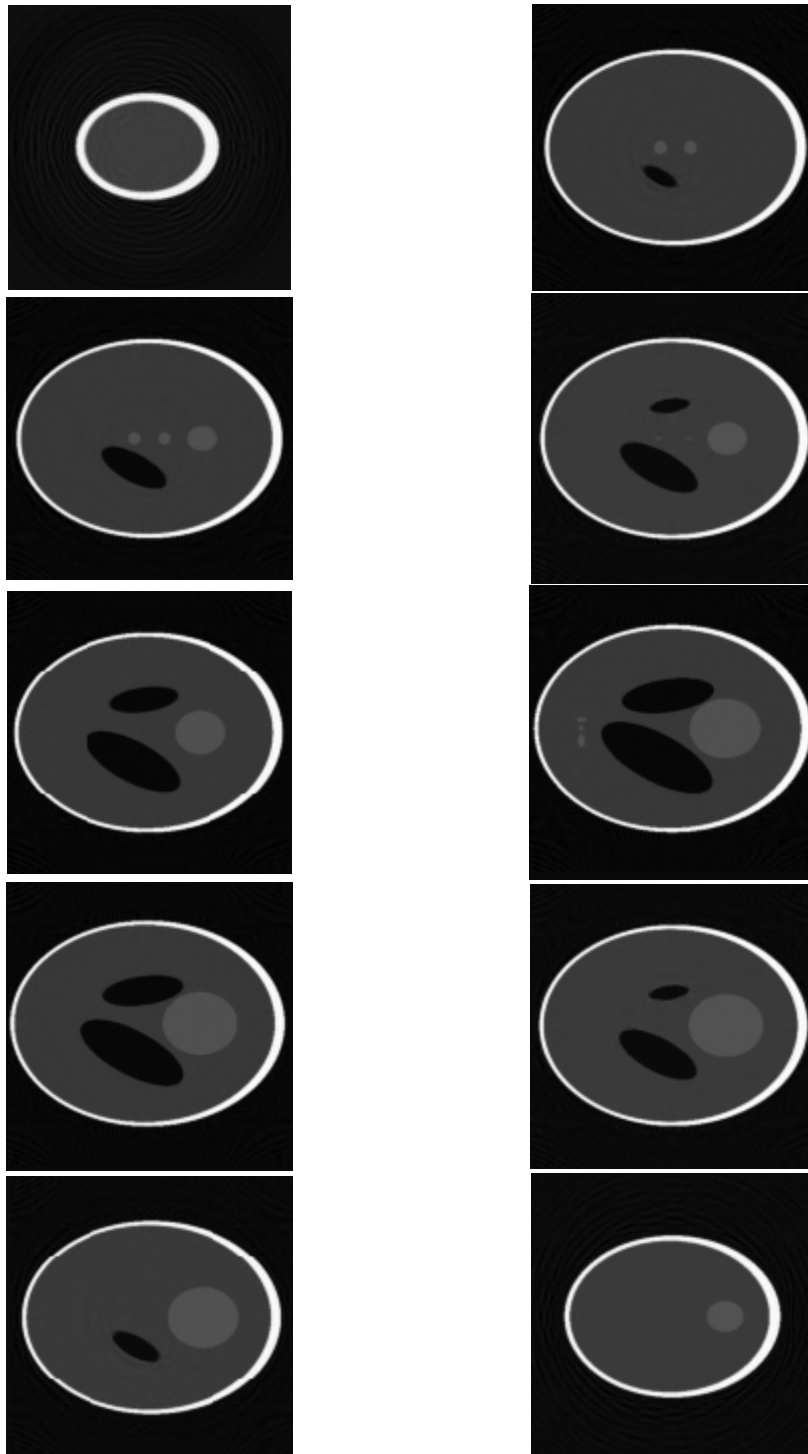


Figura 3.17: Varios slices reconstruidos del fantoma

En la figura 3.16 se observan las imágenes resultantes de las pruebas, cada una junto con un corte central horizontal comparando la señal ideal (en azul), con la obtenida en la reconstrucción (en naranja). En todas las ejecuciones se obtuvo un error relativo inferior al 2%. Finalmente, en la figura 3.17 se presentan reconstrucciones de varios slices del fantoma.

3.4.4. Reconstrucción de la simulación cubo-cilindro

En esta prueba tenemos como fantoma un cubo de agua de 4 cm de lado, en cuyo interior se encuentra un cilindro de 1 cm de diámetro y 1 cm de largo, cuyo eje tiene como centro $(x, z) = (1,5, 1,5)$ cm, tal como se muestra en la figura 3.18.

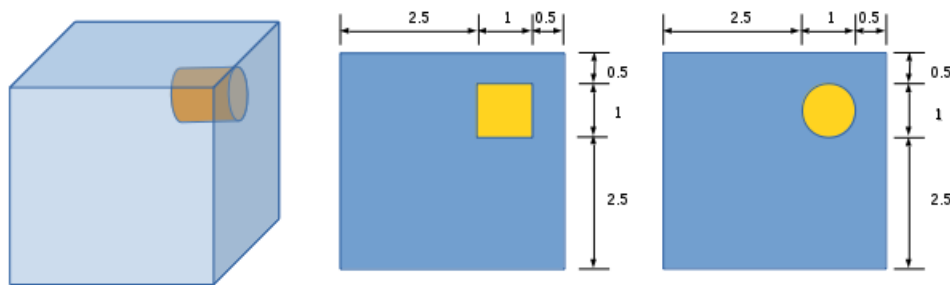


Figura 3.18: Geometría del fantoma (izquierda). Vista paralela al plano xz (centro). Vista paralela al plano yz (derecha). Las medidas están en cm.

Las proyecciones radiográficas para este fantoma fueron obtenidas del simulador *simrx*. Se ejecutaron dos versiones: una con 10^9 fotones, y la otra con 10^{10} fotones. Ambas ejecuciones se realizaron con los siguientes parámetros:

En la figura 3.19 se pueden apreciar 4 proyecciones de las 360 generadas por *simrx*.

Para las pruebas se eligió reconstruir el slice que pasa por el centro del cilindro, tal como se muestra en la figura 3.20.

Para ver la diferencia entre las simulaciones con 10^9 y 10^{10} fotones, tomamos la proyección a cero grados de ambas y trazamos un perfil que pasa por el centro del cilindro de silicio, tal como se muestra en la figura 3.21(a). La señal azul

Parámetro	Valor
Cantidad de proyecciones	360
Energía	50 keV
Tipo de haz	Cónico, 28°, monocromático
Distancia entre fuente de rayos y origen	20 cm
Distancia entre fuente de rayos y detector	39 cm
Tamaño detector	20 × 20 cm
Resolución detector	501 × 501 píxels

Tabla 3.3: Parámetros de la simulación generadas por *simræ*

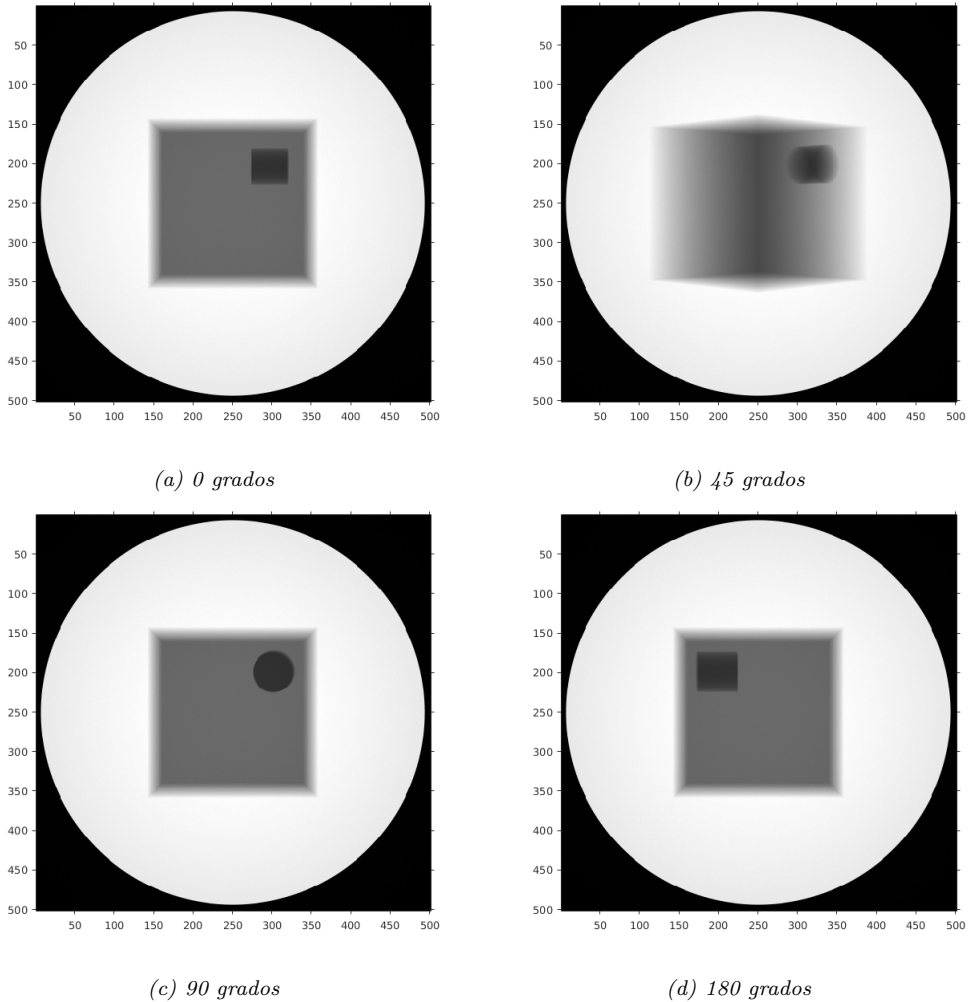


Figura 3.19: Algunas proyecciones del fantoma cubo-cilindro

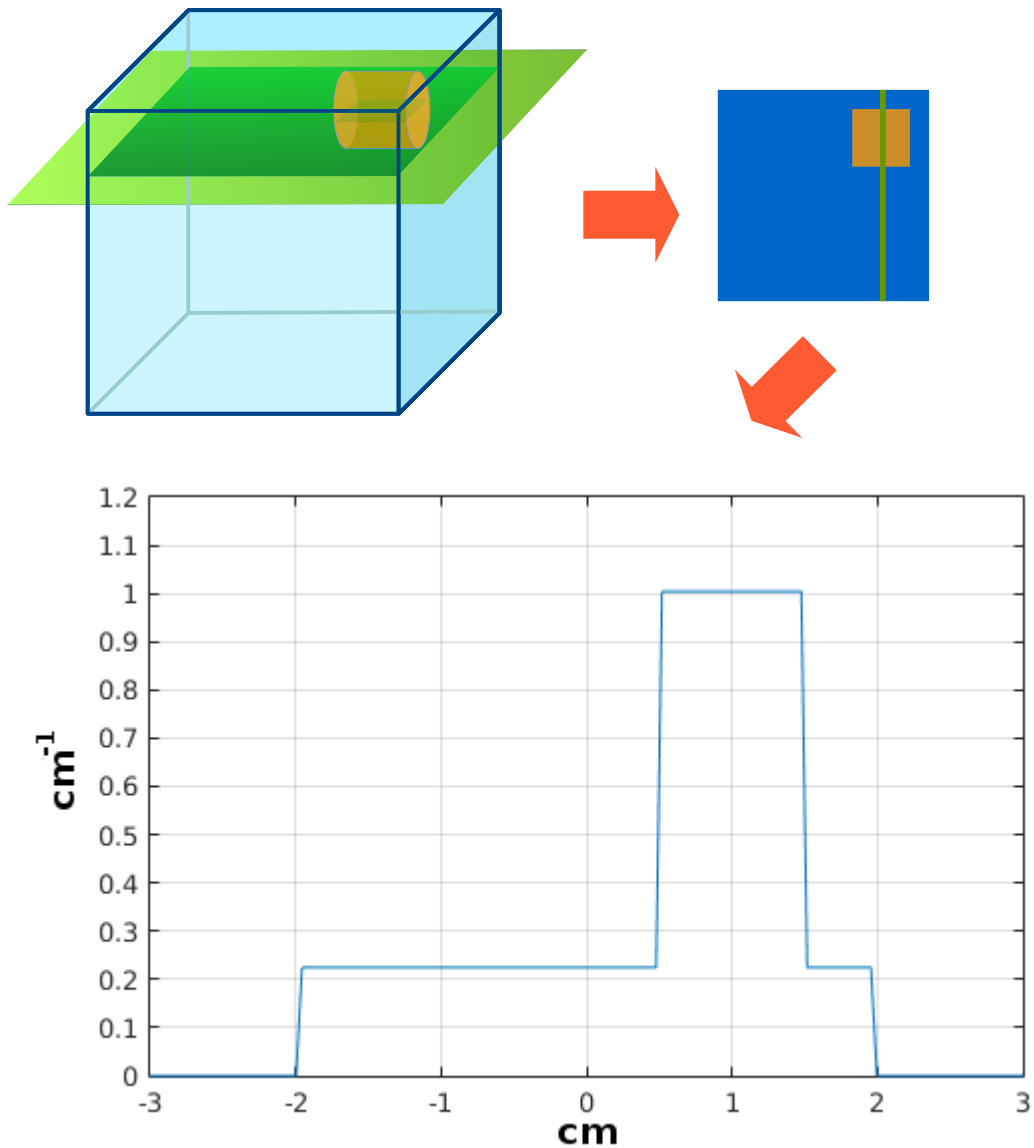


Figura 3.20: Slice reconstruido marcado con el plano verde (arriba, izquierda). Se obtiene la imagen mostrada arriba a la derecha. Allí, se vuelve a realizar otro corte (línea vertical verde) para ver el perfil de la señal (abajo, centro).

corresponde a la simulación con 10^9 fotones, la cual, como es de esperar, es mas ruidosa que la de 10^{10} fotones (señal naranja).

Para mostrar que el número de fotones afecta el resultado final de la reconstrucción tomográfica, realizamos dos ejecuciones del FDK con vecino mas cercano, frecuencia de corte 1 (recordar que esto significa que $k = k_{max}$, ver sección

2.1.5), sin filtro pasabajos, obteniendo los resultados mostrados en 3.21(c). La señal azul tiene una medida de RME de 0.037, mientras que la señal naranja tiene un RME de 0.010.

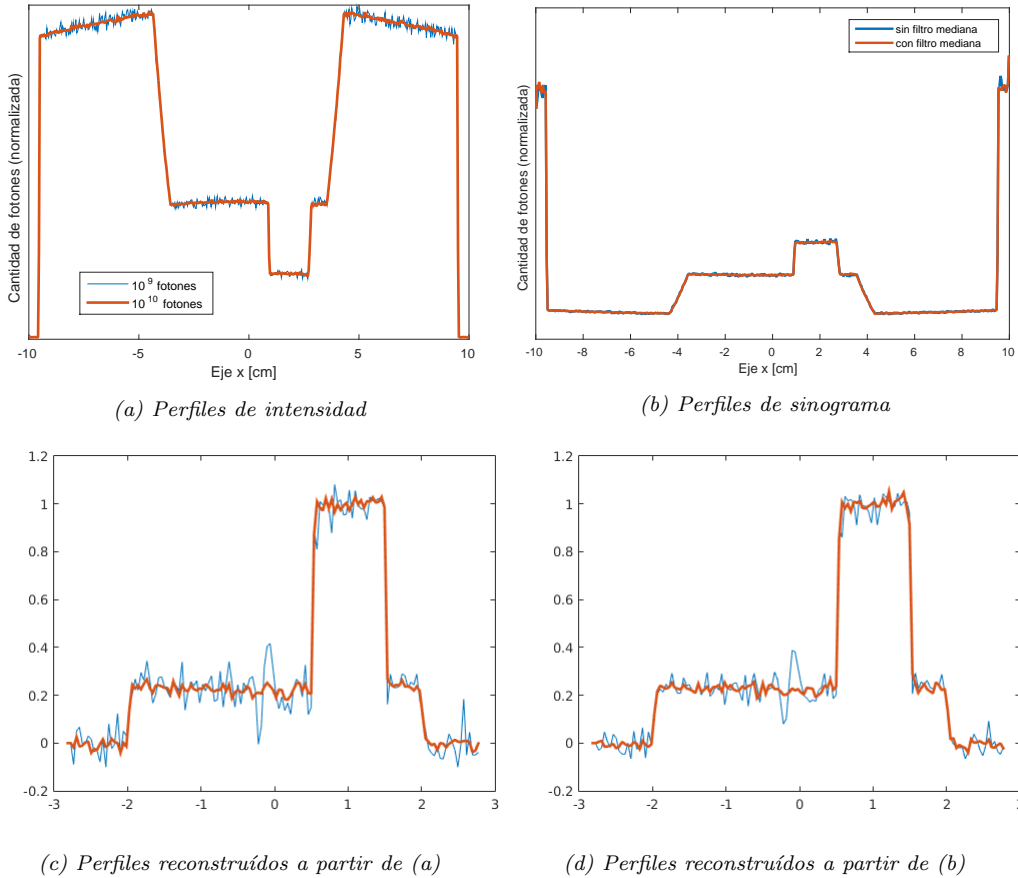


Figura 3.21: En (a) y (b) tenemos perfiles de una proyección. En (c) y (d) perfiles de una reconstrucción a partir de los datos de (a) y (b).

Otra característica que afecta la calidad de la reconstrucción tomográfica es el filtrado del sinograma antes de efectivamente ejecutar el algoritmo FDK. En la figura 3.21(b) tenemos dos perfiles de la misma proyección con igual cantidad de fotones. La señal azul se muestra tal cual, sin filtrado, mientras que la señal naranja consiste en tomar la señal anterior y tratarla con un filtro de mediana. El resultado de las reconstrucciones con ambas opciones se muestra en la figura 3.21 (d). En ella, la señal azul, con $RME=0.025$, corresponde a haber usado el sinograma original, mientras que la señal naranja, con $RME=0.009$, menos rui-

dosa, corresponde al sinograma filtrado.

En la figura 3.22 se observa la reconstrucción con el algoritmo FDK del slice que pasa por el centro del cilindro. La imagen está ampliada para ver en detalle. Se observa que la geometría generada es correcta: la sección del cubo de agua mide 4 cm, mientras que la sección del cilindro mide 1 cm, en ambos ejes x y z .

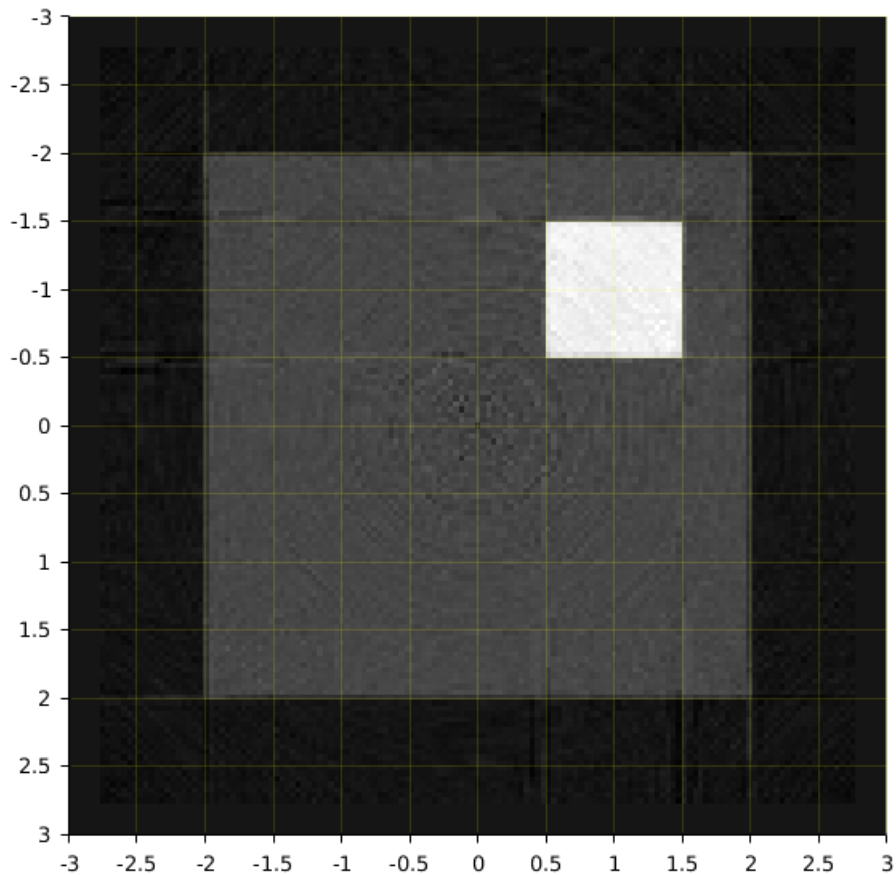


Figura 3.22: Geometría del slice que pasa por el centro del cilindro. Las medidas en ambos ejes están en cm.

Se ejecutaron varias pruebas con nuestro algoritmo FDK, variando los parámetros de filtros, frecuencias de corte, etc. Entre todas ellas se eligieron 4 pruebas, cuyos resultados fueron analizados calculando el valor RME. Además de esto, se agregó otro criterio adicional: el grado de correlación entre la señal ideal y la señal reconstruida. A pesar de que el coeficiente estadístico de correlación utiliza dos variables aleatorias, y la señal ideal (muestreada), estrictamente hablando, no es

una variable aleatoria, aún así se encontró que este coeficiente es de utilidad. El grado de correlación entre dos señales será mayor cuando el valor de su coeficiente de correlación se acerque mas a 1.

Las imágenes reconstruidas junto con un perfil indicado por la línea verde en la figura 3.20 del slice, se muestran en la figura 3.23. Allí, las señales azules corresponden al perfil ideal, mientras que las señales naranjas son las obtenidas con el algoritmo FDK.

En la tabla 3.4 se muestran los resultados de las 4 pruebas.

En la columna de RME se encuentra el error relativo medio entre la señal ideal y la obtenida con el algoritmo FDK, y en la columna r se encuentra el coeficiente de correlación entre ambas señales.

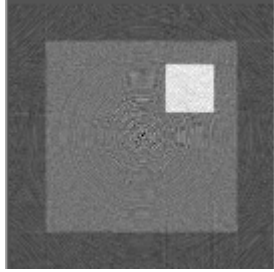
La columna FM indica si se ha utilizado filtro de mediana en el sinograma.

De acuerdo con lo discutido en la sección 3.1 se toman como valores de atenuación lineal para el agua $0,2245 \text{ cm}^{-1}$ y para el silicio $1,0045 \text{ cm}^{-1}$.

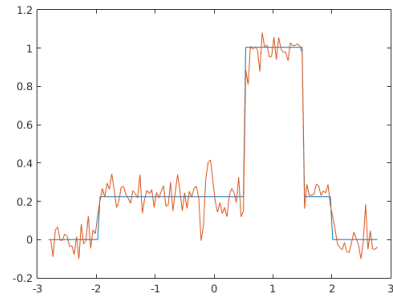
Notar que las pruebas 3 y 4 son, desde el punto de vista cualitativo, muy similares. Desde el punto de vista cuantitativo, para los valores de RME, la prueba 4 sería la imagen de mejor calidad, mientras que usando el criterio de correlación (columna r), tenemos que la prueba 3 resulta en la mejor imagen.

Prueba	Fotones	FM	Selección de valor	wm	filtro	RME	r
1	10^9	No	vecino + cercano	1	-	0.03706	0.6403
2	10^{10}	No	interpolación lineal	1	-	0.00825	0.7369
3	10^{10}	Si	interpolación lineal	0.8	Hamming	0.00553	0.7410
4	10^{10}	Si	interpolación lineal	1	BT n=1 wc=0.8	0.00551	0.7403

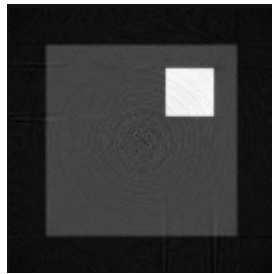
Tabla 3.4: Valores de RME para comparar la calidad de las imágenes



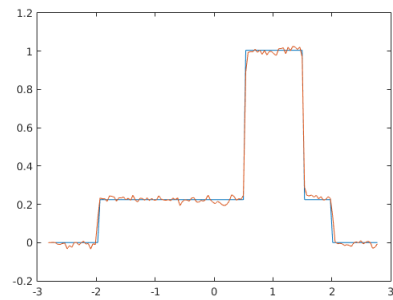
(a) Vecino mas cercano, sin filtro



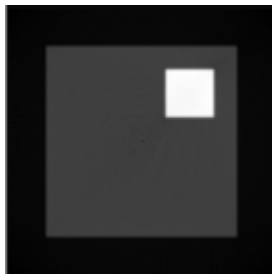
(b) Perfil de (a)



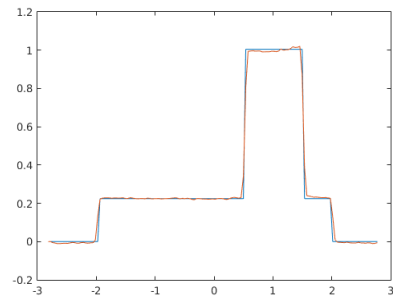
(c) Interpolación lineal, sin filtro



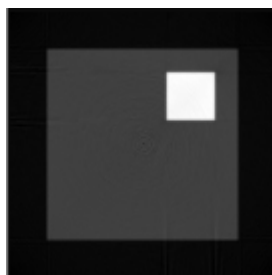
(d) Perfil de (c)



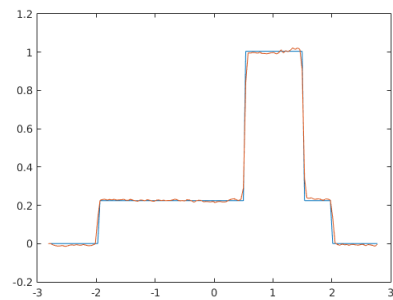
(e) Interpolación lineal + Hamming de 0.8



(f) Perfil de (e)



(g) Interpolación lineal + Butterworth de $n=1$ y $wc=0.8$



(h) Perfil de (g)

Figura 3.23: Reconstrucciones y sus perfiles centrales verticales.

Capítulo 4

Conclusiones

El algoritmo FDK, diseñado para reconstrucción tomográfica de haz cónico, basado en la redefinición del modelo matemático basado en la transformada Radon y luego del modelo de haz de abanico, ha probado ser muy simple de entender y de implementar en una computadora. Esta simplicidad también se obtuvo gracias a que las imágenes de prueba eran o bien generadas artificialmente con Matlab, o bien creadas con el simulador simrx, las cuales abstraen una gran cantidad de fenómenos que sí están presentes en imágenes tomadas con un tomógrafo real, que tiene haz policromático y que generan artefactos como el endurecimiento del haz, entre otros.

En cuanto a correctitud de la implementación del algoritmo, se hizo necesario construir fantomas simples junto con sus proyecciones de atenuación usando Matlab para poder testear el algoritmo FDK directamente y de forma aislada, actividad conocida como pruebas de unidad (unit testing en inglés). El testeo incluye además, lo que en el mundo del desarrollo de software se conoce como pruebas de integración, en donde se ejercita la cadena de procesamiento completa (simrx → conversor RAW a DICOM → FDK).

El punto más importante, la correctitud en las medidas y la calidad de las imágenes: en los ejemplos ejecutados, se constató que las medidas en la geometría de los fantomas se conservan, mientras que como se vió en los resultados, se verificó la calidad de las imágenes usando como medida el error relativo entre las señales de las imágenes reconstruidas versus las señales teóricas y se constataron las hipótesis, las cuales establecen errores menores al 2% para simrx y 1% para el FDK.

A pesar de la relativa simplicidad de los algoritmos implementados, la desventaja que poseen es la gran cantidad de parámetros que presentan sus códigos,

los cuales se deben configurar antes de cada ejecución. El simulador simrx y el algoritmo FDK presentan, pensando de manera abstracta, un panel de control con infinidad de combinaciones de configuración, con lo cual probar todos los casos es imposible para la capacidad de una sola persona, en este caso el autor del presente trabajo. Para poder realizar una cobertura de casos de prueba mayor y de mejor calidad, se necesitaría un equipo de varios profesionales de software para cubrir el esfuerzo requerido.

Por último, consideremos el tiempo de ejecución de los algoritmos. Con el FDK el tiempo de reconstrucción de un solo slice es de aproximadamente 15 segundos, con lo cual hacer una reconstrucción volumétrica entera lleva un tiempo razonable. La dificultad surge con simrx, que al ser computacionalmente intensivo, se hizo necesario ejecutarlo en un cluster, paralelizando las tareas para tener los resultados en un tiempo mucho menor.

En base a las conclusiones anteriores, los puntos a mejorar que podrían incluirse en futuros trabajos, son:

1. Refactorizar simrx para aumentar la performance de ejecución. En un caso extremo, reimplementarlo para usar GPGPU, tal como CUDA.
2. Incorporar en las simulaciones de simrx características no presentes en la actualidad, como haz policromático. Esto provocará la aparición de ciertos artefactos como el endurecimiento del haz, con lo que habrá que implementar un código para su corrección.
3. Reimplementar FDK en Qt/C++ y CUDA. Qt logrará que el operador del software tenga una experiencia de usuario satisfactoria, ya que esto implica ofrecerle una interface de usuario gráfica amigable y fácil de usar. CUDA logrará bajar el tiempo de cómputo y de esta manera aumentará la productividad y comodidad del usuario.
4. Terminar de implementar el sistema completo para poder utilizarlo con imágenes radiográficas tomadas del tomógrafo real con el objetivo de poder comenzar la creación de productos y utilidades para otras instituciones científicas y la industria.
5. Mejorar los criterios de análisis para comparar calidad entre imágenes.
6. Aumentar la capacidad del equipo de trabajo, es decir, incorporar mas personas para realizar testeo del código.

Bibliografía

- [1] Altschuler, M. D. *et al.* Mathematical aspects of image reconstruction from projections. Department of Computer Science Tech. Rep. MIPG56. State University of New York at Buffalo, Buffalo, N.Y, May 1981.
- [2] Altschuler, M. D. *et al.* Demonstration of a software package for the reconstruction of the dynamically changing structure of the human heart from cone-beam x-ray projections. Department of Computer Science Tech. Rep. MIPG32. State University of New York at Buffalo, Buffalo, N.Y, August 1979.
- [3] Altschuler, M. D. *et al.* Fully three dimensional reconstruction from cone-beam sources. Proceedings of the Conference on Pattern Recognition and Image Processing. IEEE Computer Society, Long Beach, Calif., pp. 194-199. 1978
- [4] Barney, B. Message Passing Interface (MPI). Online tutorial from Lawrence Livermore National Laboratory.
- [5] Calderón, J. Notas del los cursos de posgrado *Imágenes I y II*. Maestría en Análisis y Procesamiento de Imágenes. Universidad Nacional de Córdoba. 2012.
- [6] Colsher, J. G. Iterative three-dimensional image reconstruction from tomographic projections. *Comput. Graphics Image Processing*, 6, 513-537. 1977.
- [7] Curry, D. A. *UNIX Systems Programming for SVR4*. O'Reilly. ISBN: 1-56592-163-1
- [8] Deitel, P. and Deitel, H. *C++, cómo programar*, novena edición. Pearson. ISBN: 978-607-32-2739-1
- [9] Edminister, J. A. *Circuitos Eléctricos*, segunda edición. ISBN: 968-451-582-0.

- [10] Epstein, C. L. Introduction to the Mathematics of Medical Imaging. Second Edition, 2008. SIAM Books. ISBN: 978-0-89871-642-9
- [11] Feldkamp, L. A., Davis, L. C., and Kress, J. W. Practical cone-beam algorithm. Ford Motor Company, Dearborn, Michigan. 1984.
- [12] Ghom, A. G. (Editor). Textbook of Oral Radiology. Elseiver , 2008. ISBN: 978-81-312-1148-9
- [13] Gonzalez, R. C. and Woods, R. R. Digital Image Processing, second edition. ISBN: 978-0-201-18075-8.
- [14] Gonzalez, R. C., Woods, R. R. and Eddins, S. L. Digital Image Processing using Matlab, second edition. ISBN: 978-0-9820854-0-0
- [15] Heinz, R. Implementation of a Cone-Beam Reconstruction Algorithm in Matlab. Final report. Department of Physics and Astronomy, University of Canterbury, New Zealand.
- [16] Kak, A. C. and Slaney, M. Principles of Computerized Tomographic Imaging. Primero impreso por IEEE Press, 1988. Luego por SIAM, 2001. ISBN: 978-0898714944.
- [17] Kowalski, G. Multislice reconstruction from twin-cone beam scanning. IEEE Trans. Nucl. Sci. NS-26, 2895-2903. 1979.
- [18] Kyungsang, K. 3D Cone beam CT (CBCT) projection backprojection FDK, iterative reconstruction Matlab examples. A Matlab pack with several algorithms for tomographic reconstruction. Published in Matlab Central File Exchange web site. Version used: May 17th, 2015.
- [19] Larson, R., Hostetler, R. and Edwards, B. H. Cálculo esencial. CENGAGE Learning. ISBN: 607-481-269-1
- [20] Lewitt, R. M. and McKay, M. R. Description of a software package for computing cone-beam x-ray projections of time-varying structures, and for dynamic three-dimensional image reconstruction. Department of Computer Science Tech. Rep. MIPG45. State University of New York at Buffalo, Buffalo, N. Y. May 1980.
- [21] Minerbo, G. N. Convolutional Reconstruction from cone-beam projection data. IEEE Trans. Nucl. Sci. NS-26, 2682-2684. 1979.
- [22] Mixed Programming: Using C/C++ and Fortran together. Online tutorial.

- [23] Online Fortran 77 Tutorial. Standford University.
- [24] Online Fortran 90 Tutorial. Standford University.
- [25] Pianykh, O. S.. Digital Imaging and Communications in Medicine (DICOM). A practical introduction and survival guide. Second Edition. Springer. ISBN: 978-3-642-10849-5.
- [26] Salvat, F., Fernandez-Varea, J. M. and Sempau, J. PENELOPE-2011: A Code System for Monte Carlo Simulation of Electron and Photon Transport. Workshop Proceedings. Barcelona, Spain. 4-7 July 2011. Nuclear Energy Agency
- [27] Sauchelli, V. H. Teoría de señales y sistemas lineales. Segunda edición. Editorial Universitas Córdoba.
- [28] Schlindwein, M.. Iterative three-dimensional reconstruction from twin-cone beam projections. IEEE Trans. Nucl. Sci. NS-25 1135-1143. 1978.
- [29] Smith, S. W. The Scientist and Engineer's Guide to Digital Signal Processing. Versión libre y gratuita online.
- [30] Suetens, P. Fundamentals of medical imaging. Cambridge University Press, segunda edición, 2009. ISBN: 978-0-521-51915-1.
- [31] Tolstov, G. P. Fourier Series. ISBN: 978-0-486-63317-6
- [32] Valente, M. Notas del curso de posgrado en *Procesos físicos en la formación de imágenes radiológicas de uso médico*, Maestría en Análisis y Procesamiento de Imágenes. Universidad Nacional de Córdoba. 2013.
- [33] Valente, M. Notas del curso de posgrado en *Métodos determinísticos y estocásticos para modelar imágenes radiológicas*. Maestría en Análisis y Procesamiento de Imágenes. Universidad Nacional de Córdoba. 2013.
- [34] Ver referencias contenidas en G. T. Herman. Image Reconstruction from Projections. Academic, New York. 1980.
- [35] Webb Steve (editor). The Physics of medical imaging. Institute of Cancer Research and Royal Marsden Hospital, Sutton, Surrey.
- [36] Wolovick, N. y Bederián, C. Notas del Curso de Computación Paralela 2012. Facultad de Matemática, Astronomía y Física. Universidad Nacional de Córdoba.