

RECONSTRUCCIÓN DE IMAGEN ULTRARRÁPIDA MEDIANTE EL MÉTODO DE LA PSEUDO INVERSA

Alejandro López Montes

Directores: José Manuel Udías Moinelo y Joaquín López Herraiz

Curso 2015/2016



AGRADECIMIENTOS

A lo largo de todo este año, he tenido la suerte de poder trabajar tanto con profesores como con alumnos que realmente han dejado huella.

Cuando aún no tienes muy claro en qué consiste exactamente eso de investigar, la guía y el consejo de personas que llevan en esto mucho tiempo son fundamentales. Por eso, primeramente quiero agradecer a mis directores Joaquín y José Manuel que hayan sabido orientarme y empujarme dentro de este mundo apasionante que es la investigación y la física médica. Cuando empecé el año apenas sabía programar lo básico en Matlab, y Fortran me resultaba un mundo oscuro y lleno de horrores. Gracias en gran medida tanto a Joaquín como a José Manuel he ido superando mis miedos a la programación y he conseguido algo que hace un año no hubiera dicho que podría salir de mis manos: programar algo funcional y útil. Programación aparte, he podido aprender de ellos todos los conocimientos que día a día me repetían y que hoy me ayudan a entender mejor la reconstrucción de imagen.

Quiero agradecer a Oli que como siempre haya estado a mi lado todo este año, aguantándome hasta en los momentos más difíciles y ayudándome a superar todo lo que se ha puesto por delante.

A mi familia quiero agradecerles el intentar comprender qué hacía cuando me iba a trabajar y el haberme apoyado siempre a lo largo de este año y todos.

A mis compañeros del Grupo de Física Nuclear. A mis compañeros de despacho Víctor, Jaime y Pablo por todos esos buenos momentos que amenizan los días de trabajo. A todos mis compañeros del máster por este año y todos los viajes y experiencias que hemos disfrutado juntos. En general a todas las personas que me han apoyado y que de una manera o de otra han contribuido a que este trabajo haya podido ver la luz. Gracias.

ABSTRACT

The problem of the reconstruction of images from nuclear medicine has been studied and developed for many years. There have been used several methods with different results in resolution and computing time from analytical methods as FBP to iterative solutions where the variety of algorithms is enormous. Further, data acquired in 3D are often rebinned into 2D data sets in order to speed-up reconstruction. Thus the reconstruction of 3D data can be cast into two steps, first a rebinning of data into 2D sets, followed by reconstruction of the 2D slices. For many practical purposes, it is accurate enough to approximate both the rebinning and the 2D-reconstruction problems as a set of linear equations, that can be solved by means of the pseudo-inverse matrix. We show that with the pseudo-inverse rebinning, we can get results much better than with the usual SSRB recovering more information than with the most sophisticated FORE rebinning algorithm, and further the rebinning as well as the 2D reconstruction algorithms based on the pseudoinverse can be done in extremely short times.

Motivación

A lo largo de este trabajo se revisarán los actuales métodos de reconstrucción de imagen en PET y se propondrá el método de la pseudoinversa estudiando las mejoras que este introduce frente a utilizar métodos convencionales. Esperamos conseguir imágenes más resolutivas y de una forma mucho más rápida y eficiente que con ningún otro método y además esperamos mejorar también las formas de reducir los problemas de 3D en PET a varios problemas de 2D. Se comparará la eficacia de la pseudo inversa frente a SSRB que es el método más sencillo utilizado habitualmente en rebineado y se expondrá el software original que se ha implementado para obtener la matriz de respuesta del sistema.

El trabajo pasa por una introducción al PET para después discutir los actuales métodos de reconstrucción de imagen. Seguidamente se introducirá el concepto de pseudoinversa y la base matemática de la que partimos para después describir el problema de la reconstrucción en 3D. Finalmente se discutirá el código desarrollado a lo largo de este año y los resultados que se han obtenido con él.

El objetivo final del trabajo es crear las herramientas para poder hacer imagen en PET en tiempo real utilizando el método de la pseudo inversa.

Se utilizará para el trabajo la geometría del escáner para PET preclínico SUPERARGUS, fabricado por SEDECAL.

A lo largo del trabajo hemos utilizado códigos escritos tanto en FORTRAN como en MATLAB.

Índice

1. Introducción	1
1.1. Breve introducción a la física del PET y sus aplicaciones.	1
1.2. Interacción de los fotones con la materia.	2
1.3. Tratamiento de datos	3
1.4. Conceptos básicos de la reconstrucción de imagen.	4
1.5. Principales problemas y fuentes de ruido en PET.	5
1.6. Bibliografía	9
2. Métodos de reconstrucción de imagen 2D	11
2.1. Métodos analíticos (FBP).	11
2.2. Métodos iterativos (SART, OSEM, ISRA, OE).	12
2.3. Comportamiento frente a datos ruidosos de los métodos iterativos.	15
2.4. Bibliografía.	18
3. Pseudo-inversa.	20
3.1. Aproximación al problema de la pseudo-inversa.	20
3.2. Descomposición en valores singulares (SVD).	20
3.3. Pseudo-inversa a partir de SVD.Pseudo-inversa de Moore Penrose.	22
3.4. Reconstrucción de imagen con pseudo-inversa.	23
3.5. Bibliografía.	25
4. Reconstrucción de imagen en 3D.	27
4.1. Reducción del problema 3D a 2D.	27
4.2. Span. MRD. Michelograma.	27
4.3. SSRB.	30
4.4. Bibliografía.	31
5. Reconstrucción de datos PET basado en la pseudoinversa.	32
5.1. Objetivo de la investigación.	32
5.2. Matriz de respuesta del sistema. Geometría del escáner Superargus.	32
5.3. Código LUT. (LookUp Table.	34
5.4. Código rebineador	34
5.5. Código reconstructor de rodajas.	37
5.6. Resultados y discusión.	39
5.7. Aproximación a la solución directa del problema en 3D.	42
6. Conclusiones y expectativas de futuro.	43

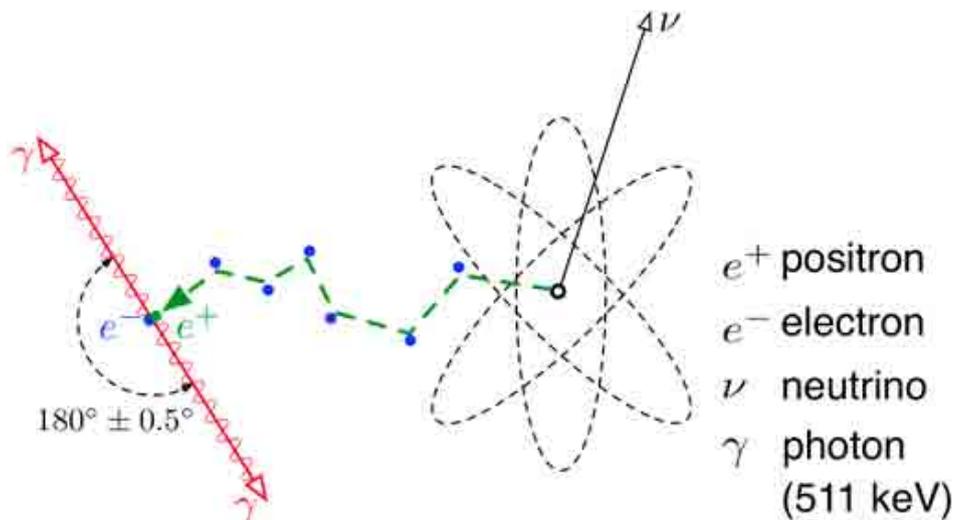
1. Introducción

1.1. Breve introducción a la física del PET y sus aplicaciones.

El PET (Positron Emission Tomography o Tomografía por Emisión de Positrones) es una disciplina de imagen médica basada la inoculación de un radiofármaco trazador que se distribuirá por el organismo y cuyas emisiones permitirán la obtención de la imagen.

Una de las singularidades que tiene esta técnica frente a otras como el CT (Computed Tomography), es que se trata de una imagen funcional y no anatómica, es decir, nos permite observar el funcionamiento del organismo y no simplemente su distribución anatómica. Se puede combinar esta prueba con CT o incluso últimamente con Resonancia Magnética Nuclear para una imagen completa tanto funcional como anatómica del organismo. El hecho de que se trate de una imagen funcional permite a los médicos la detección precoz de enfermedades tales como el cáncer o el alzheimer así como estudiar la viabilidad del miocardio en caso de infarto.

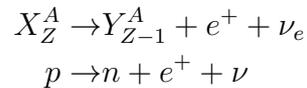
Las bases físicas del PET se basan en la aniquilación de electrón y un positrón. Cuando esto ocurre, ambas partículas dan lugar a dos fotones (o incluso tres, aunque con mucha menos probabilidad), cada uno de ellos con una energía de 511 keV (la energía en reposo tanto del positrón como del electrón). Por conservación del momento, ambos fotones salen en direcciones contrapuestas, por lo que estudiando la detección de fotones en coincidencia (detectados a la vez o en una ventana temporal muy pequeña) podemos saber la trayectoria que estos llevaron y así intentar buscar la posición donde se originaron (lugar donde se produjo la aniquilación). Todo el proceso de encontrar la posición donde se produjo la aniquilación es de lo que se encargan los métodos de reconstrucción de imagen.



Gráfica 1: Desintegración β^+ y aniquilación e^+ , e^-

Para generar los positrones dentro del organismo se utilizan emisores β^+ , esto es, se usan radioisótopos que decaigan beta positivo y que por tanto liberen positrones. El decaimiento β^+ o beta positivo es un proceso en el cual uno de los protones del núcleo decae en un neutrón

emitiendo un positrón (e^+ antipartícula del electrón) y un neutrino electrónico para conservar el número leptónico en la reacción.



Dado que la masa en reposo del neutrón es ligeramente superior a la del protón en reposo, esta reacción no puede darse para un protón libre. Sin embargo cuando se encuentra ligado a un núcleo, la masa efectiva de los nucleones disminuye por la interacción fuerte lo suficiente como para que esta reacción pueda tener lugar.

Estos radioisótopos se pueden generar en aceleradores de partículas cargadas bombardeando núcleos estables para conseguir otros inestables. En general, el tipo de acelerador más usado es el ciclotrón.

El radioisótopo más utilizado es el FDG en el que el ^{18}F se utiliza para marcar glucosa, molécula que absorben todos los procesos metabólicos que requieran energía y que es muy útil para detectar cáncer debido al acelerado metabolismo de las células cancerígenas. En cuanto a los isótopos y las aplicaciones de cada uno existe una amplia bibliografía que puede consultarse al final de la sección. No obstante no le dedicaremos más tiempo en este trabajo puesto que la descripción de los radioisótopos y su producción no es el objetivo de este proyecto.

Un escáner PET en humanos está formado por unos pocos anillos de cristales centelleadores, acoplados a tubos fotomultiplicadores, colocados alrededor del paciente para obtener imágenes de las distintas partes del cuerpo. En PET preclínico en animales pequeños, se pueden construir escáneres más complejos que engloben el tamaño completo del animal. En este trabajo, se usa el modelo de PET preclínico SUPERARGUS fabricado por la empresa SEDECAL. Un esquema global de este escáner puede verse en la sección 5.2 dedicada a la geometría del SUPERARGUS.

1.2. Interacción de los fotones con la materia.

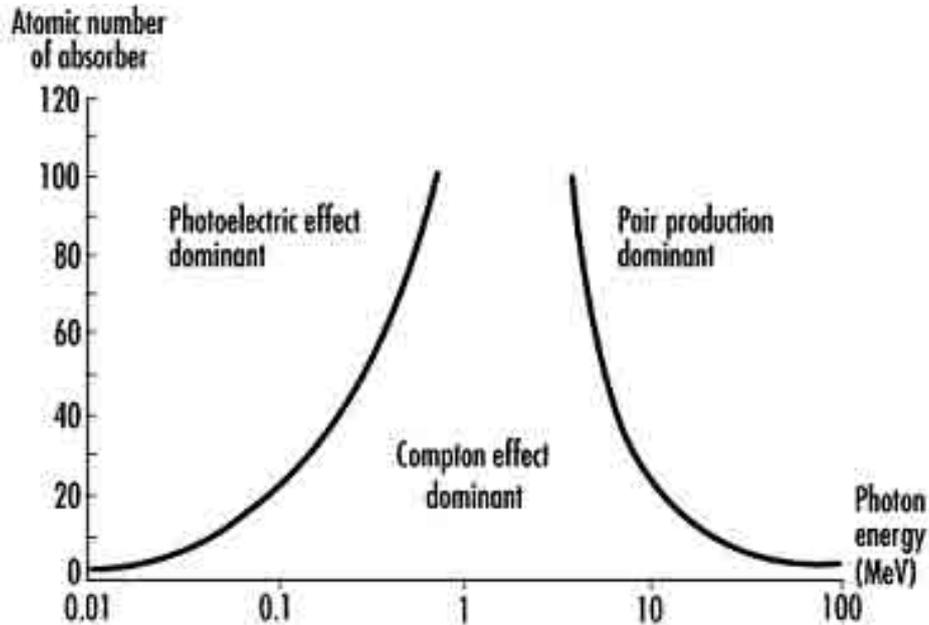
Para entender bien todo lo que ocurre en un escáner PET, es conveniente tener presentes las formas de interactuar de la radiación, en concreto de los fotones con la materia. Las principales formas en las que se da esta interacción son:

- *Efecto fotoeléctrico*: suele predominar para energías bajas del fotón y números atómicos grandes en el material donde se produce. Consiste en la absorción total de un fotón por parte de un átomo que para compensar el exceso de energía del fotón emite un electrón, a veces llamado fotoelectrón. La energía mínima que debe tener el fotón será la energía de ionización del átomo en cuestión y la energía de más que este lleve le corresponderá al electrón en forma de energía cinética. Este efecto es el que buscamos que se de en un detector para observar el fotopico de absorción que en el caso del PET buscamos entorno a la ventana energética de 511 keV.

- *Efecto Compton*: este efecto juega un papel importante en las fuentes de error que se discutirán en la sección 1.5. Consiste en un choque elástico de un fotón con, habitualmente, un

electrón. Se debe conservar tanto el momento lineal como la energía cinética del sistema por tratarse de una colisión elástica.

- *Creación de pares*: puede ocurrir que un fotón con energía superior a 1022 keV de lugar a un par $e^- e^+$ al verse afectado por las interacciones de la materia a su alrededor. Debido a que en PET trabajamos con energías por debajo de 511 keV, este fenómeno no es relevante.



Gráfica 2: Dominio de los diferentes procesos de interacción de los fotones con la materia.

1.3. Tratamiento de datos

Es importante almacenar los datos de las coincidencias de los cristales del escáner de manera que posteriormente sean fácilmente tratables. Existen varias formas de recoger los datos. Aquí expondremos las más utilizadas actualmente.

1) Modo lista:

Lo más simple, sería ir anotando sin más cada coincidencia en el orden de detección, dando información de los dos cristales implicados en la detección. Esto origina lo que se conoce como modo lista, es decir, un listado completo y detallado de todas las detecciones individualizadas. Es obvio que este modo origina ficheros de tamaños muy grandes. Estamos hablando de inoculaciones en humanos de varios cientos de MBq, y períodos de medición de entorno a la media hora. Eso supone varios millones de cuentas, por lo que trabajar con ficheros de ese calibre se hace costoso [1]. Es necesario entonces reducir el número de datos para poder trabajar cómodamente con ellos.

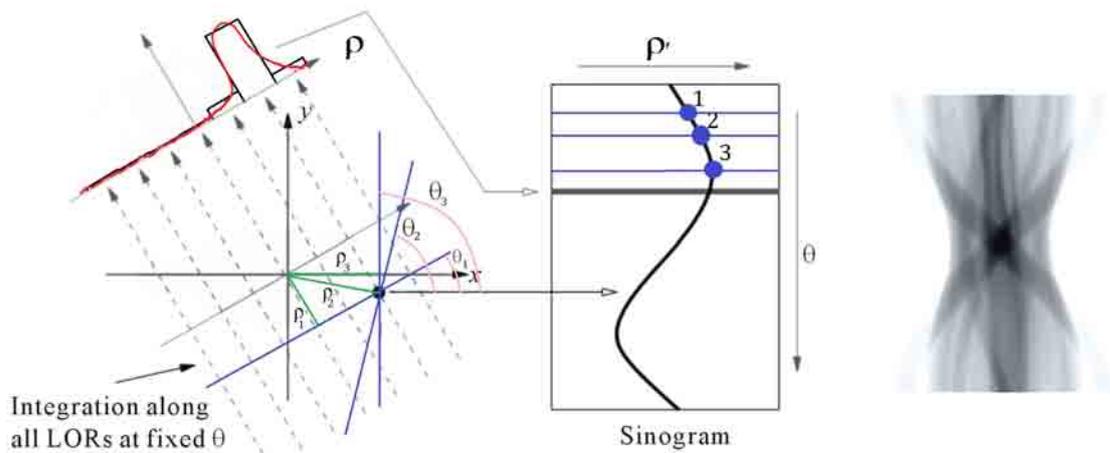
2) Histograma de LOR's:

Lo siguiente más intuitivo sería histogramar los datos del modo lista en base a las coordenadas de los cristales de interacción $(X1, Y1, Z1)$ y $(X2, Y2, Z2)$. Donde 1 y 2 serán los dos cristales

participantes en la coincidencia. Con esto ya nos reducimos bastante los datos, y además introducimos un concepto importantísimo en todo el proceso de la reconstrucción de imagen. Se trata del concepto de línea de respuesta (LOR de las siglas en inglés de Line Of Response). Consiste en la línea recta que uniría los dos cristales donde se produce la interacción. Esta línea atraviesa todo el interior del escáner y suponemos que la aniquilación que produjo los fotones de la coincidencia ha debido provenir de algún punto de esa línea. Podemos entonces reducir los datos que teníamos en el modo lista a número de cuentas en cada LOR, que constituye nuestro histograma de datos. A lo largo de este trabajo se han utilizado datos histogramados de esta manera, pues se ha considerado que es el formato con el que perdemos menos información de los que tienen un tamaño aceptable para computerizarlos.

3) Sinograma:

Existe otro método ampliamente utilizado sobre todo entre los métodos de reconstrucción analíticos que discutiremos en la Sección 2 de el presente trabajo. Esa forma de ordenar los datos se llama sinograma. No deja de ser también un histograma, sólo que en vez de histogramar los datos como LOR's frente al número de cuentas, lo hace ordenando las líneas de respuesta según dos parámetros: la distancia de la línea al origen de coordenadas, generalmente situado en el centro geométrico del escáner (ρ) y el ángulo que forma dicha línea con un eje de referencia (θ). El nombre le viene de que si representamos las coincidencias debidas a una fuente puntual no centrada, los máximos de cada LOR escrita en función de (ρ, θ) la imagen tiene forma de seno.



Gráfica 3: Sinograma

En la imagen la coordenada x' se refiere a ρ

1.4. Conceptos básicos de la reconstrucción de imagen.

El objetivo principal de la reconstrucción de imagen es el de llevar los datos adquiridos a la imagen de la distribución de la fuente de la que se obtuvieron. Para ello se utilizan algoritmos matemáticos.

Escribimos nuestros datos en un vector Y , otro vector al que llamaremos X contiene la información de la imagen. Ambos vectores, datos e imagen, vienen relacionados mediante una matriz

A a la que llamaremos matriz de respuesta del sistema (SMR, siglas en inglés de System Matrix Response). Esta matriz estará constituida por unos elementos a_{ij} que contendrán la información de la probabilidad de que la línea de respuesta i atraviese el elemento de la imagen j . En general, el proceso pasa por voxelizar o pixelizar la imagen para formar X . Esto es, discretizar el espacio de la imagen para poder asignar cómodamente y de manera unívoca una probabilidad $a_{i,j}$ de que la LOR i atraviese el vóxel (o píxel) j .

En base a esto tenemos un sistema de ecuaciones que resolver $AX = Y$. Las dimensiones de los datos y de la imagen hacen que el número de ecuaciones a resolver sea muy elevado. Por tanto la dificultad en la reconstrucción está en encontrar el proceso más óptimo de llegar a la solución correcta del sistema. Además de eso, en general estos sistemas no tienen una solución única, por lo que tendremos que encontrar aquella que más se acerque al resultado real.

La forma de llegar a esa solución se ha tratado ampliamente a lo largo del pasado siglo, y aún a día de hoy aparecen métodos nuevos o mejoras a los ya existentes. En la sección 2 presentaremos los métodos más utilizados actualmente antes de describir en la sección 3 el que se ha tratado mayoritariamente a lo largo de este problema que es el de la pseudoinversa.

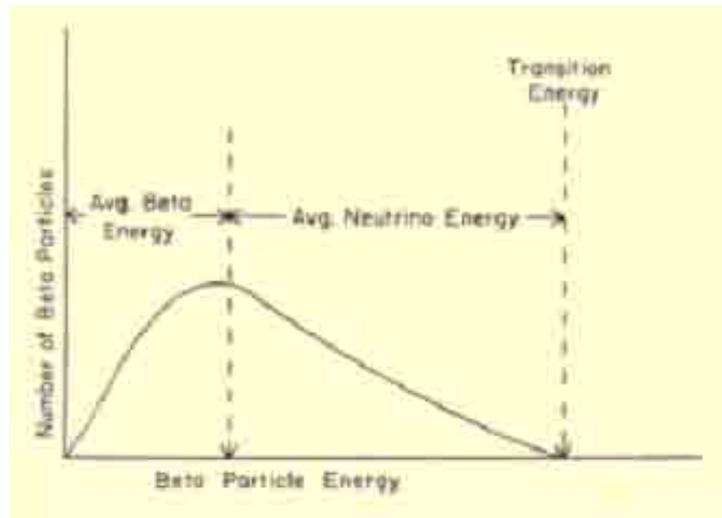
1.5. Principales problemas y fuentes de ruido en PET.

Si los datos adquiridos fuesen datos perfectos, la solución del sistema anterior resultaría mucho más sencilla y se obtendría una imagen bastante aproximada de la fuente. Sin embargo, esta no es la situación habitual. De hecho en general, las fuentes de error a la hora de medir emborronan y estropean los datos, lo que dificulta llegar a la solución real del sistema y no a una imagen irreal.

Existen varias fuentes de error en PET siendo algunas la no colinearidad de los fotones y el rango del positrón. Existen otras fuentes de error que también comentaremos brevemente pero comenzaremos por estas dos las cuales encontramos más relevantes.

En la sección 1.1, se indicó que los positrones producidos en la desintegración β^+ de los núcleos se aniquilaban con los electrones que encontraban en la materia. Siendo un poco más precisos, el proceso de aniquilación electrón – positrón, se produce cuando el centro de masas del sistema se encuentra prácticamente en reposo.

Al producirse la desintegración, al tratarse de tres cuerpos emitidos como producto, el espectro de energía del positrón no es discreto como en el caso de las desintegraciones α o γ , sino continuo.

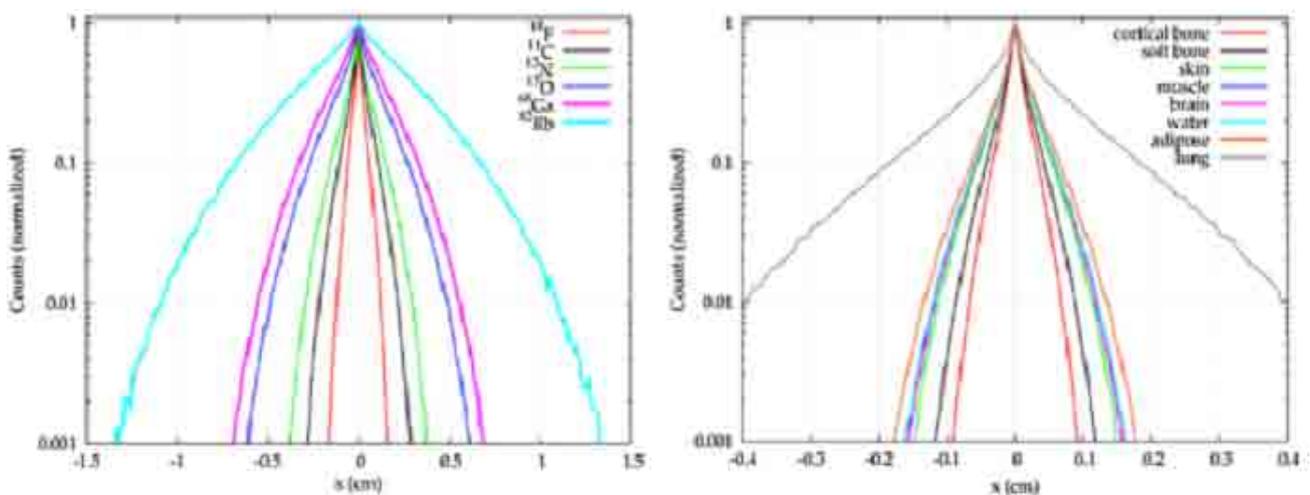


Gráfica 4: Energía promedio positrón

Debido a que la sección eficaz de aniquilación del positrón es máxima cuando el sistema e^-e^+ está en reposo, el positrón tiene que perder primero la práctica totalidad de la energía cinética que le correspondiera en el decaimiento. Hasta que eso ocurre, la partícula habrá recorrido una cierta distancia, y por lo tanto el lugar en el que se producirán los dos fotones no coincidirá con el lugar donde se encuentra el radioisótopo emisor. Esta distancia es lo que se conoce como rango del positrón. Esto genera un emborronamiento en la imagen y pérdida de resolución.

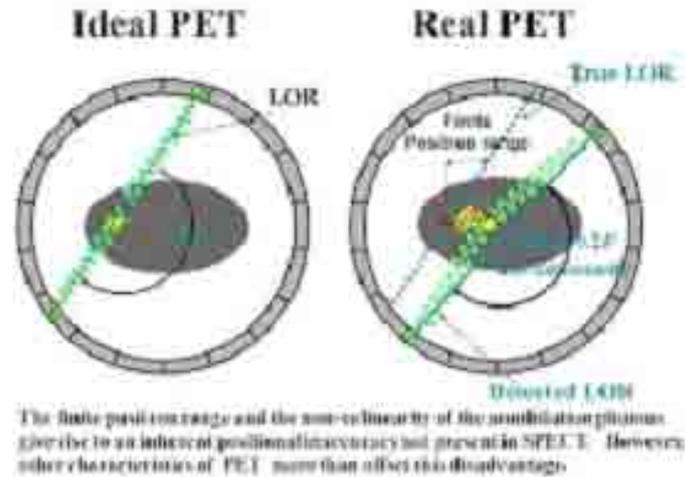
Existen diversas formas de corregir este efecto, aunque en ningún caso podremos eliminarlo por completo. Este problema puede encontrarse ampliamente detallado en [1].

La corrección del rango del e^+ se basa en intentar modelizar la forma que tiene la probabilidad de encontrar la aniquilación a una distancia concreta de la fuente. En función del tipo de isótopo utilizado y del material biológico en el que se produzca la emisión, esta probabilidad puede variar sustancialmente.



Gráfica 5: Rango del positrón esperado

Aunque el positrón en su recorrido pierda la energía cinética de la reacción, puede ocurrir que, en el momento de la aniquilación, el sistema $e^- e^+$ tenga un cierto momento lineal. El efecto que esto producirá es que los dos fotones producidos no saldrán con un ángulo exacto de 180 grados. Este efecto puede provocar diferencias angulares que pueden llegar incluso a medio grado [7]. Esto puede provocar que los cristales de interacción sean distintos de los esperados y que por lo tanto la LOR que asignamos no sea la LOR real que debiera corresponderle a la desintegración.



Gráfica 6: No colinearidad de los fotones.

Existen métodos que intentan recomponer los datos para eliminar parte del ruido, esto es, deconvolucionar la imagen hasta una sin ruido. En [2] hay una sección dedicada a este caso donde se discuten las ventajas y desventajas de deconvolucionar la imagen.

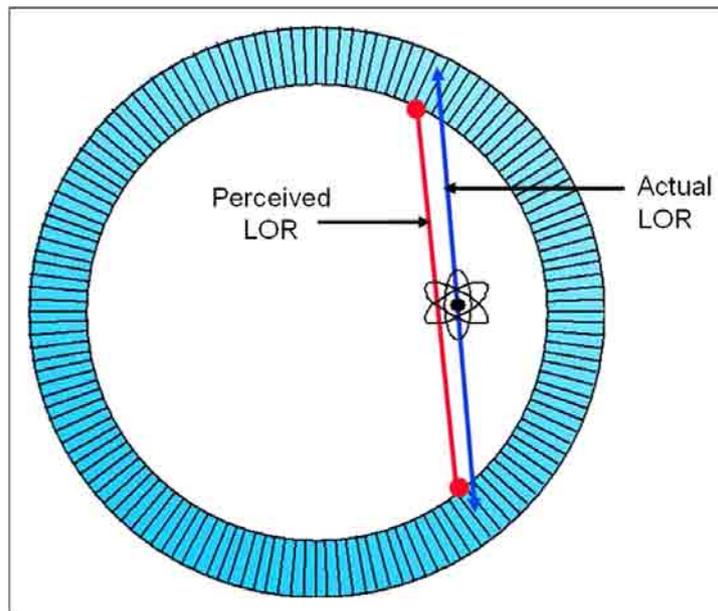
Aparte de las ya comentadas, hay otras fuentes de error en la adquisición de datos que también introducen emborronamiento en nuestras imágenes.

Otro efecto que se da, es la posible interacción Compton de los fotones resultantes de la aniquilación. Existe una probabilidad, que vendrá determinada por la densidad electrónica del material, de que los fotones interactúen Compton con el medio, desviándose de su trayectoria y por lo tanto llegando al cristal que no le correspondería. Esto da origen a falsas coincidencias que también emborronan nuestra imagen. El efecto Compton también puede tener lugar en el propio cristal del escáner, en ese caso el fotón dejaría en el cristal correcto una pequeña parte de su energía y podría volver a interactuar con los cristales de alrededor originando más ruido en nuestra imagen. Además, también pueden darse interacciones antes de llegar al cristal lo que produce un efecto de atenuación de los fotones. [6]

Además de los efectos debidos a la física del problema que acabamos de comentar existen otros efectos puramente aleatorios, como es la posibilidad de coincidencias aleatorias. A veces ocurre que se detectan en coincidencia no dos, sino tres o más fotones que provienen de distintas aniquilaciones. En estos casos resulta más conveniente descartar la cuenta para evitar añadir más ruido a la imagen. También podría ocurrir que se detecten sólo dos fotones en coincidencia pero

dado que la eficiencia de los detectores no es del 100% provengan de aniquilaciones distintas. Este proceso sí resulta inevitable y también habrá que tenerlo en cuenta al reconstruir la imagen.

Por cuestiones de geometría y construcción del escáner, también podemos encontrar otras fuentes de error. Es el caso de la profundidad de interacción. Cada cristal del escáner tiene unas ciertas dimensiones, de anchura, altura y profundidad. La asignación del cristal de la interacción se hace mediante un proceso conocido como lógica de Anger, sobre este tema puede consultarse [8] y [9]. Ahora bien, en cuanto a la profundidad del cristal a la que haya interactuado la partícula, la línea de respuesta que le asignemos puede cambiar considerablemente.



Gráfica 7: Profundidad de interacción.

Para tratar con este problema, es frecuente construir cristales centelleadores con al menos dos capas de diferente material, de manera que se pueda determinar a cuál de los dos materiales, y por tanto a qué profundidad, le corresponde la coincidencia. Esto es lo que se conoce como phoswich. Para que el phoswich funcione, los dos cristales deben tener características de detección distintas, para que la señal electrónica que generen sean diferenciables. También en [8] y [9] se puede encontrar más información al respecto.

Por último mencionaremos problemas en la adquisición de datos debidos al montaje experimental. En general suelen tratarse de problemas debidos a los detectores, como el tiempo muerto que puede ocasionar que perdamos cuentas que lleguen muy próximas entre sí o el efecto conocido como pile-up, o apilamiento, que ocurre cuando en una misma ventana temporal de medición se contabilizan más de una cuenta, sumando la energía que le correspondería a cada una e identificándola como un fotón de mucha más energía de la real. Esto hace que si estamos mirando en una ventana de energía de entorno a 511 keV, que es donde esperamos encontrar nuestros fotones, una cuenta colocada al doble de energía por el detector no será contabilizada.

En general, lo que finalmente ocurre es que todas estas fuentes de error emborronan nuestra

imagen. Para intentar tenerlas en cuenta, se asignan líneas de respuesta de un cierto ancho, esto es, algo así como tubos de respuesta para cada par de cristales.

Otro problema que encontramos pero el cual no se tiene en cuenta añadiendo grosor a las líneas de respuesta es el hecho de que no todos los cristales del escáner tienen la misma eficiencia. Esto hace que pueda haber regiones de la imagen más muestreadas que otras debido a diferencias de eficiencia en las líneas de respuesta que las atraviesan. Para solucionar este problema se suele hacer una imagen de una fuente uniforme que nos indique las diferencias de sensibilidad de cada elemento de la imagen, una vez tenemos nuestra imagen, dividimos las cuentas de cada elemento por su sensibilidad. Esto se conoce como normalización. [5]

1.6. Bibliografía

Referencias

- [1] JACOBO CAL GONZÁLEZ. *Positron range and prompt gamma modeling in PET imaging*.
<http://eprints.ucm.es/28460/1/T35753.pdf>
- [2] JOAQUÍN LÓPEZ HERRAIZ. *Técnicas avanzadas de reconstrucción de imagen nuclear PET, X-CT y SPECT*.
http://nuclear.fis.ucm.es/webgrupo_2007/TRABAJOS%20GRUPO/Trabajos%20fin%20master/MASTER_JOAQUIN_LOPEZ_REV-baja.pdf
- [3] KHALED M. A. ABUSHAB. *Simulation and Image Reconstruction of Clinical TOF-PET Scanners*.
http://nuclear.fis.ucm.es/webgrupo_2014/trabajos-investigacion-grupo/Tesis/KHALED%20ABUSHAB/THESIS%20_29-1-2013.pdf
- [4] ELENA HERRANZ MUELAS. *Formulación de modelos dinámicos de distribución temporal de fármacos en animales de laboratorio y contrastación con datos adquiridos en PET*.
http://nuclear.fis.ucm.es/webgrupo_2007/TRABAJOS%20GRUPO/Trabajos%20fin%20master/Elena%20Herranz/HerranzMuelas_Elena.pdf
- [5] ESTHER VICENTE TORRICO. *Caracterización, mejora y diseño de escáneres PET preclínicos*.
http://nuclear.fis.ucm.es/webgrupo_2014/trabajos-investigacion-grupo/Tesis/ESTHER%20VICENTE%20TORRICO/EVICENTE_THESIS_FINAL.pdf
- [6] MARÍA PINTO MONEDERO. *Evaluación de semiconductores como detectores de radiación para PET. Detectores de CZT*.
http://nuclear.fis.ucm.es/webgrupo_2007/TRABAJOS%20GRUPO/Trabajos%20fin%20master/Maria%20Pinto/PintoMonedero_Maria.pdf
- [7] C.ROSS SCHMIDTLEIN. DEPARTMENT OF MEDICAL PHYSICS. NUCLEAR IMAGING PHYSICS GROUP. MEMORIAL SLOAN-KETTERING CANCER CENTER. *Basic PET imaging physics: Illustrated via Monte Carlo*.
<http://www.aapm.org/meetings/amos2/pdf/49-14363-4210-81.pdf>

- [8] JACOBO CAL GONZÁLEZ. *Mejora de la identificación del cristal de interacción en escáneres PET de alta resolución mediante simulaciones.*
http://nuclear.fis.ucm.es/webgrupo_2014/trabajos-investigacion-grupo/trabajos-acad-dirigidos-licen-fisicas/trabajo_dirigido-jacobo.pdf
- [9] CATHERINE MURPHY-O'CONNOR CANTILLO. *Mejora de la identificación del cristal de interacción en escáneres PET de alta resolución.*
http://nuclear.fis.ucm.es/webgrupo_2014/trabajos-investigacion-grupo/trabajos-acad-dirigidos-licen-fisicas/trabajo-catherine-julio7-2006.pdf
- [10] GLENN F. KNOLL. *Radiation Detection And Measurements.*
- [11] GRÁFICA 1 <https://es.wikipedia.org/wiki/Archivo:Annihilation.png>
- [12] GRÁFICA 2 <https://bnnas.files.wordpress.com/2015/11/1.gif>
- [13] GRÁFICA 3 <http://file.scirp.org/Html/13-9101545%5Cef2da7bc-79d7-4778-8640-119e6510d1eb.jpg>
- [14] GRÁFICA 4 https://upload.wikimedia.org/wikipedia/commons/thumb/8/85/Beta_decay_spectrum.png/300px-Beta_decay_spectrum.png
- [15] GRÁFICA 5 <http://iopscience.iop.org/0031-9155/58/15/5127/downloadFigure/figure/pmb462478f5>
- [16] GRÁFICA 6 http://images.slideplayer.com/16/4896073/slides/slide_48.jpg
- [17] GRÁFICA 7 <http://tech.snmjournals.org/content/37/2/63/F1.large.jpg>

2. Métodos de reconstrucción de imagen 2D

2.1. Métodos analíticos (FBP).

Los métodos de reconstrucción analíticos son en principio los métodos más rápidos de reconstrucción de imagen. El más común es FBP, el cual se basa en una transformada de Fourier bidimensional que se conoce como transformada de Radon. Utiliza los datos recogidos previamente en un sinograma y los retroproyecta (de ahí el nombre Filtered Back Projection o retroproyección filtrada) utilizando la transformada de Fourier bidimensional para llevar los datos a la imagen.

El principal problema que tienen los métodos analíticos es que se comportan mal para datos incompletos como por ejemplo escáneres con huecos entre los detectores, y al hacer la reconstrucción generan artefactos en la imagen. Hay que diseñar por tanto programas de rellenado de huecos o similares para evitarlo.

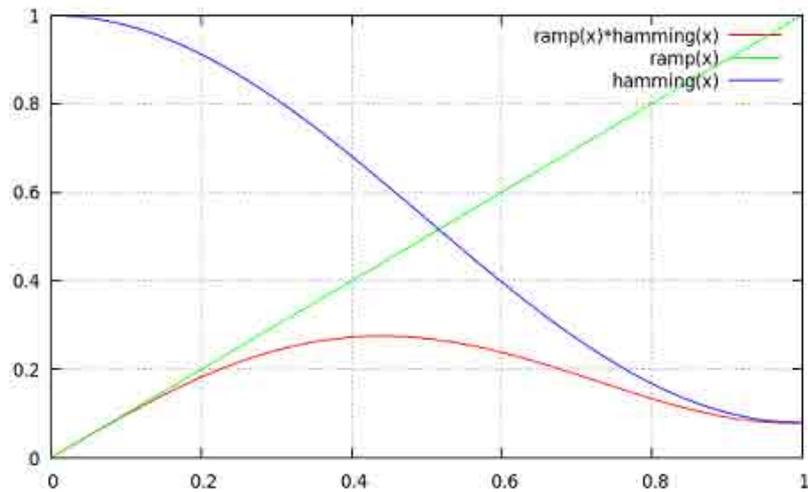
En general las imágenes que se generan con FBP son poco resolutivas comparándolas con otros métodos iterativos.

A la hora de utilizar métodos analíticos hay que prestar especial cuidado a temas de muestreo. En lo que se basa el principio matemático de la FBP es en que, mediante una transformada de Fourier en dos dimensiones, podríamos llevar la información del objeto al espacio de los datos:

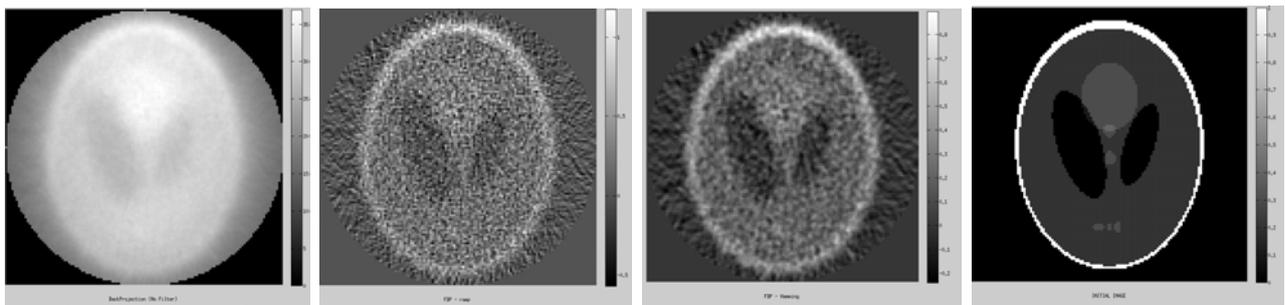
$$f(\rho, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin(\theta - \rho)) dx dy$$

Este espacio de ρ, θ resulta tener el aspecto de un sinograma. Por lo tanto si tenemos el sinograma completo de un objeto, se puede encontrar una expresión que nos lleve de nuevo a la imagen. Al hecho de pasar del objeto al sinograma le llamaremos proyectar y al del proceso contrario retroproyectar. Hay que puntualizar que a la hora de retroproyectar estamos tomando una serie de datos discretos. A la hora de reconstruir una serie de datos, podemos muestrear a distintas frecuencias. No todas ellas son válidas a la hora de reconstruir la imagen, de hecho habrá una frecuencia máxima a partir de la cual estaremos obteniendo resultados irreales, asignándoles a x, y unas frecuencias que no le corresponden. Este fenómeno se llama aliasing y el teorema de Nyquist determina que la frecuencia máxima de muestreo deberá ser la mitad de la frecuencia de Nyquist. En [1] se puede encontrar una formulación matemática detallada del problema.

Además, en una reconstrucción por retroproyección, las frecuencias bajas suelen quedar muy muestreadas mientras que las frecuencias altas tienen menos muestreo, se hace necesario utilizar una serie de filtros como por ejemplo el filtro rampa que además de cortar el muestreo antes de la frecuencia máxima le asigna pesos menores a las frecuencias más bajas y mayores a las más altas. Para suavizar el ruido que es predominante a frecuencias altas se utilizan otros filtros como el filtro Hamming o el filtro sinc [2], [3].



Gráfica 8: Filtros rampa y Hamming , así como el producto de ambos



((a)) Reconstrucción sin filtrar. ((b)) Filtro rampa ((c)) Filtro Hamming ((d)) Imagen inicial

Gráfica 9: Imágenes reconstruidas con FBP

Para obtener mejores resultados en FBP se pueden hacer correcciones en los sinogramas para mejorar la imagen suavizando el ruido antes de la reconstrucción. [2]

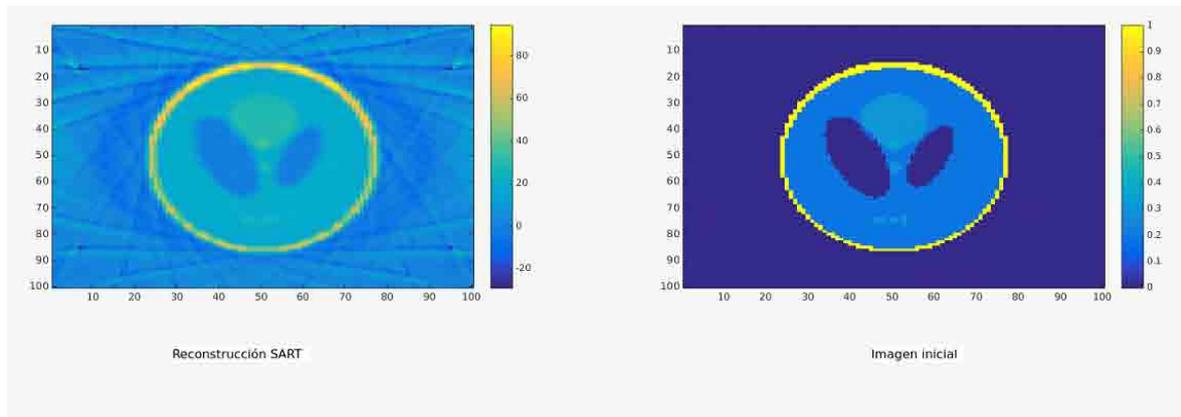
2.2. Métodos iterativos (SART, OSEM, ISRA, OE).

Los métodos analíticos tienen la ventaja de ser muy rápidos, sin embargo están muy limitados en cuanto a resolución. Los métodos iterativos se van acercando progresivamente a la solución para llegar a obtener una imagen lo más próxima posible a la real.

En el código iterativo podemos incluir información dentro de la matriz del sistema que usaremos en la reconstrucción, información que, en una FBP, al ser un método analítico, no podríamos utilizar. Existe un gran número de algoritmos que se pueden implementar para realizar una reconstrucción iterativa. Aquí recogemos algunos de ellos:

- 1) **SART** (Simultaneous Algebraic Reconstruction Technique)

Este método se basa en una sucesión de proyecciones y retroproyecciones de la imagen comparando el resultado. Parte de una imagen inicial simple, por ejemplo una imagen de ceros y a cada proyección compara con los datos del sinograma o histograma, después corrige los datos mediante operaciones algebraicas, suma y resta, y vuelve a retroproyectar los datos nuevos formando la siguiente imagen. Cada una de las proyecciones-retroproyecciones de la imagen se denomina iteración.



Gráfica 10: Reconstrucción SART. A la izquierda una reconstrucción con SART de la proyección de la imagen de la derecha. Se han utilizado 1000 iteraciones.

$$X_{i+1} = X_i + \frac{A' \cdot (Y - A \cdot X_i)}{A' \cdot N}$$

Donde la matriz N lleva incluidos parámetros tanto de normalización como de sensibilidad de los detectores. En principio si fuese uniforme para todos los cristales, la matriz N sería simplemente una fila de unos. Por su parte A' es la matriz transpuesta del sistema que relaciona líneas de respuesta con vóxeles de la imagen.

La solución de este algoritmo iterativo se obtiene de resolver el sistema $A\Delta X = Y$ mediante la minimización del problema de métodos cuadrados:

$$\sum_i \left(\sum_j a_{ij} \cdot x_j - y_i \right)^2 = 0 \quad (1)$$

El problema ampliamente descrito puede encontrarse en [1].

2) **EM-ML** (Expectation Maximitation – Maximum Likelihood)

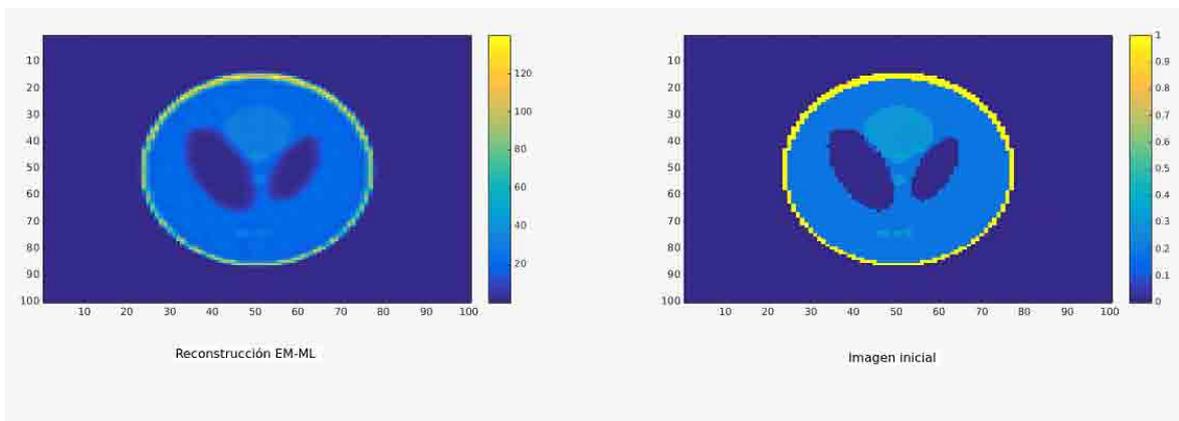
El proceso es muy parecido al anterior en el sentido de que también se basa en una sucesión de proyecciones y retroproyecciones. No obstante la forma en la que corregimos los datos cambia, dado que este método no se fundamenta en la resolución del problema de mínimos cuadrados sino en la maximización de la función likelihood (función de máxima verosimilitud), método estadístico bayesiano ampliamente conocido. Puede verse en [2] la obtención del algoritmo usado en este método. El algoritmo que maximiza esta función máxima verosimilitud se conoce como

Expectation Maximitation (EM). Finalmente la expresión queda:

$$X_{i+1} = X_i \cdot \frac{A' \cdot (Y/A \cdot X_i)}{A' \cdot N}$$

En este caso, la imagen inicial X_0 no puede elegirse nula, por lo que cogemos otra sencilla, por ejemplo un vector de unos.

Habitualmente suele utilizarse este algoritmo de reconstrucción de imagen con una implementación que reduce el número de iteraciones necesarias para converger a la solución haciendo varias actualizaciones (subsets) de la imagen en cada iteración. Esta implementación se llama OSEM (Ordered Subsets Expectation Maximitation). [2]



Gráfica 11: A la izquierda reconstrucción EMLL de la proyección de la imagen de la derecha.

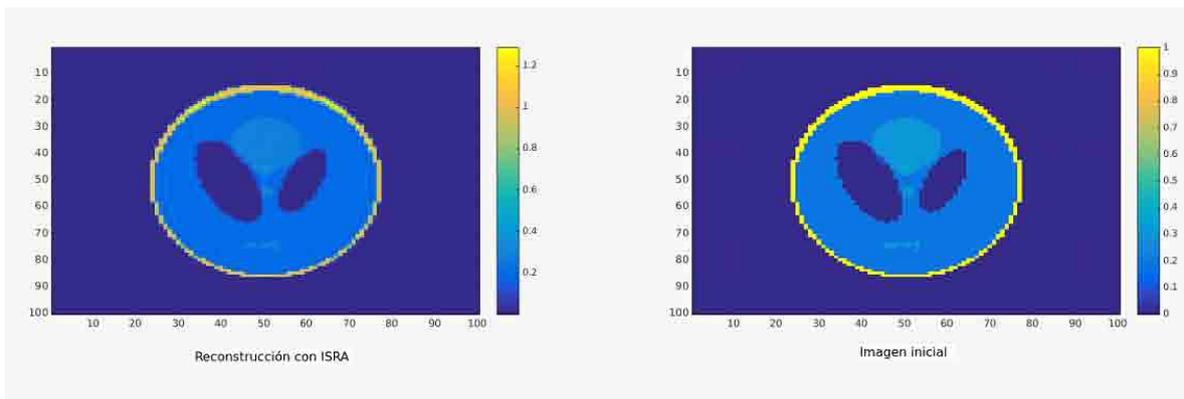
3) ISRA (Image Space Reconstruction Algorithm)

Frente a OSEM se tiene también otro algoritmo que en algunas circunstancias puede resultar más conveniente debido a que supone un menor coste computacional.

Parte también de una imagen sencilla, por ejemplo un vector de unos. Se tiene previamente el cálculo de la retroproyección de los datos, que se haría con $A' \cdot Y$. Esto facilita los cálculos posteriores pues no hay que hacer este producto en cada iteración. Nótese que hacer $A' \cdot Y$ equivaldrá a hacer una retroproyección sin filtrar de la imagen (ver sección 2.1.). A cada iteración, la imagen es corregida por la relación entre los datos retroproyectados y la retroproyección de la proyección de la imagen de la iteración anterior. Esto es:

$$X_{i+1} = X_i \cdot \frac{M}{B \cdot X_i}$$

Con $M = A'Y$ previamente calculada y guardada y con $B = A'A$ también calculada previamente. En esencia, el algoritmo se parece bastante a OSEM, salvo que evita tener que hacer el cálculo de las proyecciones y retroproyecciones a cada iteración. En [4] puede verse una comparativa entre los métodos OSEM e ISRA.



Gráfica 12: A la izquierda reconstrucción con ISRA de las proyecciones de la imagen de la derecha.

4) OE (Origin Ensemble)

Además de los métodos ya expuestos, hemos querido hablar de otro método cuyo fundamento matemático es diferente. Se trata de Origin Ensemble. Es un algoritmo puramente bayesiano. Pasa por construir las LORs y a cada una asignarle el número de cuentas que le corresponde. Una vez ha hecho eso recorre un bucle pasando por todas las cuentas y mueve cada una a otro punto dentro de la misma línea de respuesta a la que pertenecía. Al final la hipótesis de la que partimos es que la cuenta asociada a esa LOR concreta tiene que haber venido de alguna parte de dicha línea. Con la matriz de respuesta del sistema previamente se ha asignado una sensibilidad a cada vóxel de la línea, esto es, la probabilidad de que una cuenta provenga de ese vóxel concreto sumada a todas las LORs. El algoritmo, que es de carácter estocástico, como puede verse en [5] acaba asignando una probabilidad de que cada cuenta provenga de un vóxel dado usando la información del resto de cuentas.

La imagen inicial de la que parte es una imagen completamente aleatoria pero manteniendo el número de cuentas que le corresponde a cada línea de respuesta.

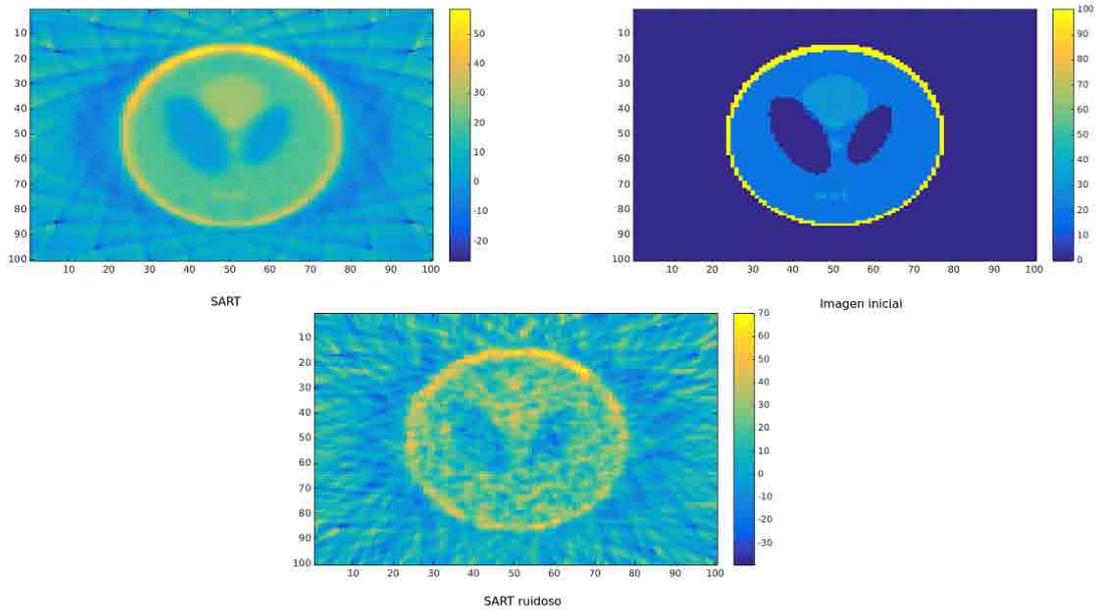
Este método puede ser muy útil a la hora de hacer time of flight (TOF) en PET. [6],[7].

2.3. Comportamiento frente a datos ruidosos de los métodos iterativos.

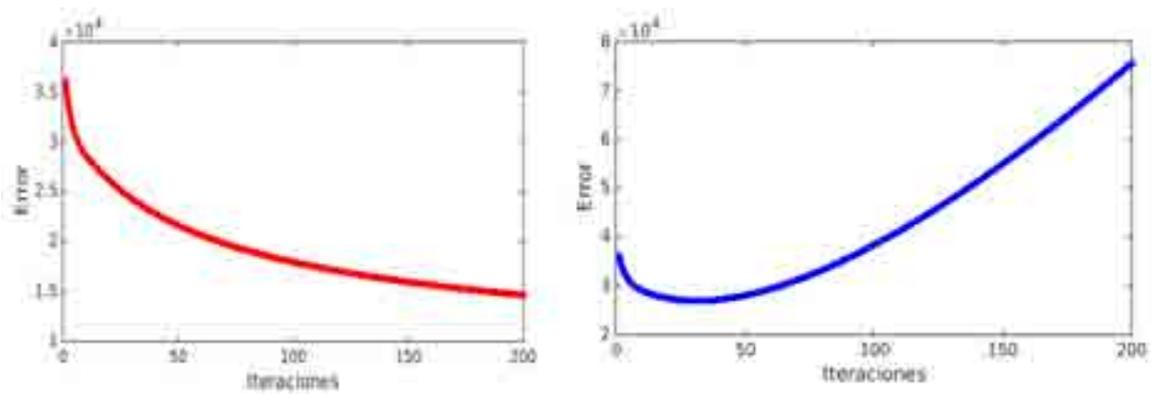
En esta sección hemos descrito una serie de algoritmos de reconstrucción iterativa dando una idea de cómo funcionan éstos e ilustrándolos con imágenes reconstruidas con dichos algoritmos (salvo en el caso de OE que resulta más complejo) utilizando la matriz de respuesta del sistema obtenida por los códigos desarrollados durante este Trabajo de Fin de Máster que se plantean en la sección 5.

En todo caso no hemos descrito como se comportan frente al ruido. En los casos anteriores hemos obtenido unos datos ideales simplemente proyectando mediante la matriz del sistema. Las imágenes obtenidas se puede observar que no eran perfectas debido a que en la matriz de res-

puesta del sistema se le ha dado un cierto ancho a las líneas de respuesta para tener en cuenta los efectos físicos descritos en la sección 1.4. y que además se trata de un escáner realista con huecos entre los detectores y demás efectos geométricos. Podemos generar las imágenes anteriores añadiendo ruido para ver como se comportan:



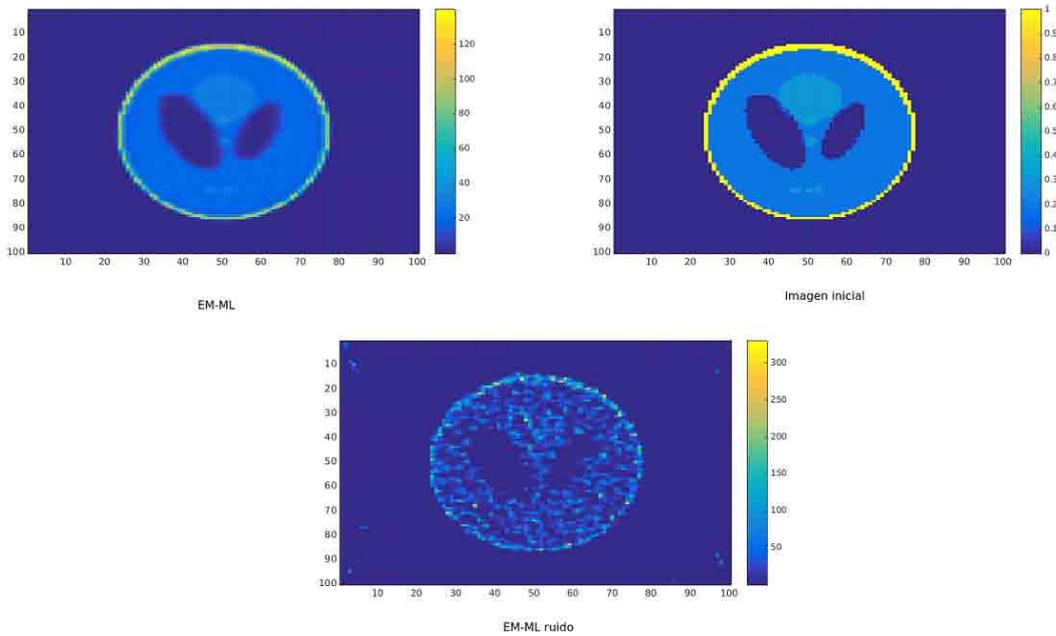
Gráfica 13: SART con ruido abajo frente a SART sin ruido arriba a la izquierda. Arriba a la derecha imagen inicial para comparar



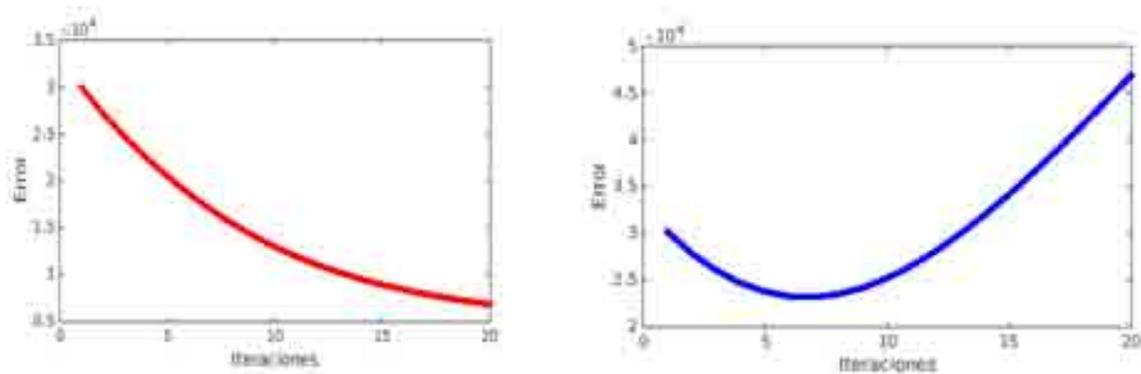
Gráfica 14: Comportamiento del error frente al número de iteraciones en SART. En la figura de la izquierda para el caso sin ruido decae el error con el número de iteraciones. En la figura de la derecha para el caso ruidoso, el error crece con el número de iteraciones

Comparando con la imagen sin ruido la diferencia es bastante clara. En la gráfica de error frente a número de iteraciones vemos cuánto nos vamos acercando o alejando en cada iteración a la solución real del problema. Es importante ver que a partir de un cierto número de iteraciones,

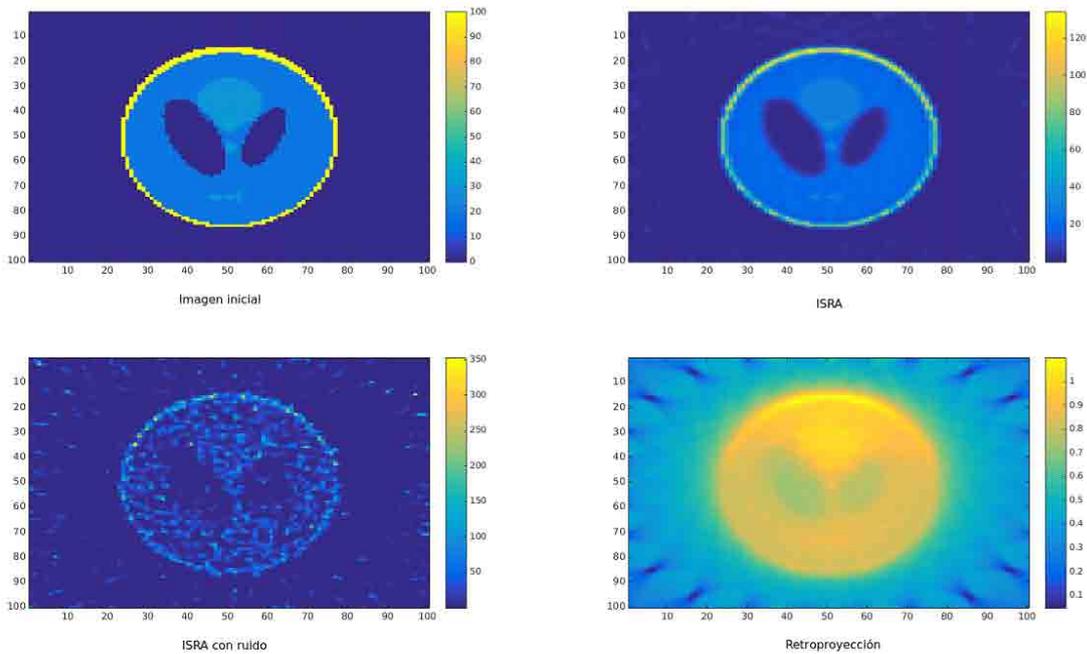
para el caso ruidoso, acaba predominando el ruido. Por eso es importante tener en cuenta que a partir de un cierto punto no sólo no ganamos resolución según seguimos iterando sino que multiplicamos el ruido. Podemos observar lo mismo para el caso de EM-ML e ISRA. En la imagen de EM-ML se ve claramente como si elegimos demasiadas iteraciones acabamos estropeando la imagen.



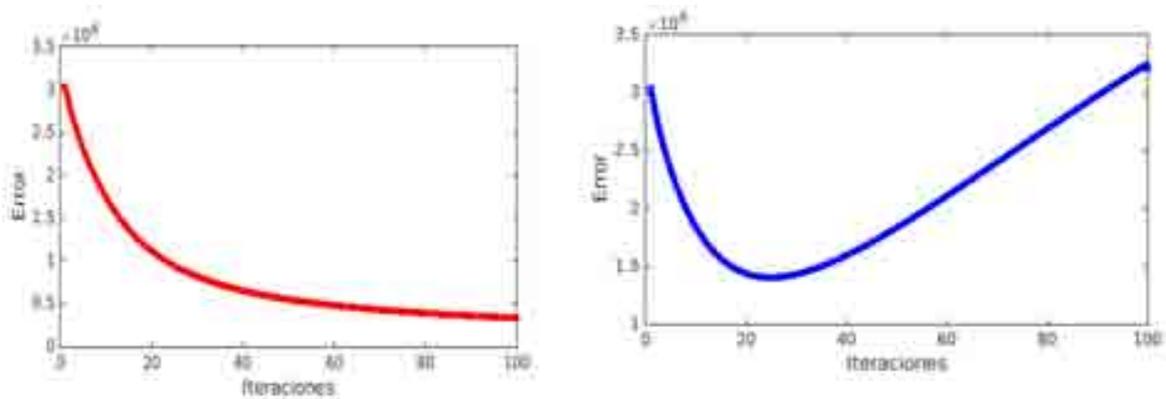
Gráfica 15: EM-ML con ruido abajo y sin ruido arriba a la izquierda. 100 iteraciones. Arriba a la derecha se presenta la imagen inicial para poder comparar.



Gráfica 16: Error frente a iteraciones. EM-ML. A la izquierda podemos ver como los datos sin ruido se acercan a la solución mientras que para datos ruidosos el comportamiento diverge



Gráfica 17: Reconstrucción con ISRA. Abajo a la derecha podemos ver una retroproyección sin filtrar de la imagen original que se puede ver arriba a la izquierda. Arriba a la derecha la reconstrucción sin ruido usando ISRA y abajo a la izquierda una reconstrucción con ruido usando 100 iteraciones.



Gráfica 18: Comportamiento del error frente a las iteraciones en ISRA. Se puede ver como se va reduciendo para datos sin ruido y como a partir de cierto punto para datos ruidosos empeora.

2.4. Bibliografía.

Referencias

- [1] FRANK NATTERER. *The Mathematics of Computerized Tomography*.
- [2] JOAQUÍN LÓPEZ HERRAIZ. *Técnicas avanzadas de reconstrucción de imagen nuclear PET, X-CT y SPECT*.

http://nuclear.fis.ucm.es/webgrupo_2007/TRABAJOS%20GRUPO/Trabajos%20fin%20master/MASTER_JOAQUIN_LOPEZ_REV-baja.pdf

- [3] R. W. HAMMING. *Digital Filters*.
- [4] G.E. B. ARCHER AND D. M. TITTERINGTON. UNIVERSITY OF GLASGOW. *The iterative Image Space Reconstrucion Algorithm (ISRA) as an alternative to the EM algorithm for solving positive linear inverse problems*.
- [5] ARKADIUSZ SITEK. *Statistical Computing in Nuclear Imaging*.
- [6] CHRISTIAN WÜLKER, ARKADIUSZ SITEK , AND SVEN PREVRHAL. *Time Of Flight- PET image reconstruction using origin ensembles*.
- [7] J. L. HERRAIZ AND A. SITEK. *Técnicas avanzadas de reconstrucción de imagen nuclear PET, X-CT y SPECT*.
- [8] MICHEL DEFRISE, PAUL E KINAHAN AND CHRISTIAN J MICHEL. *Image Reconstruction Algorithms in PET*.
<http://eknygos.lsmuni.lt/springer/370/63-91.pdf>

3. Pseudo-inversa.

3.1. Aproximación al problema de la pseudo-inversa.

Partamos de un sistema de ecuaciones de la forma $AX = Y$ tal como se indica en 1.3. Es decir que la matriz de probabilidades A , a la que llamamos matriz de respuesta del sistema, al multiplicarla por los datos de la imagen nos determina el número de cuentas que deben provenir de una determinada LOR, esto es, Y . El cálculo de esta matriz se trata en la sección 5.2 dedicada a ella. De momento supongamos que esta matriz es conocida. El problema entonces, como en todos los demás casos de reconstrucción, será llegar a la solución más aproximada a la realidad para X , donde almacenamos la imagen en sí.

Si supiéramos que la matriz A fuese cuadrada e invertible, el problema se reduciría trivialmente al cálculo de la inversa de A de modo que $X = A^{(-1)}Y$. No obstante, en general nuestra matriz A para empezar no será cuadrada, sino que tendrá una dimensión $m \times n$ donde m será el número de LORs del sistema y n el número de vóxeles de la imagen. Además aunque se diese el caso de que fuera cuadrada ello no garantizaría que fuera invertible, pues podríamos encontrar que la matriz tuviera alguna fila o columna linealmente dependiente de las demás. Resumiendo, en general no podemos resolver el sistema con el simple cálculo de la matriz inversa dado que esta no existe en la mayor parte de los casos. Para resolverlo pues, necesitaremos obtener una matriz que cumpla las mismas propiedades que la matriz inversa y que sea generalizable a todos los casos.

Este problema matemático fue resuelto por E.H. Moore en 1920 y después también por R. Penrose en 1955, que obtuvieron lo que denominaron una matriz pseudoinversa o inversa generalizada. Conocemos esta matriz como Pseudo-inversa de Moore-Penrose y aún a día de hoy sigue siendo el método más utilizado para llegar a una solución a nuestro problema [4].

Lo que pretende este método es poder resolver el sistema $\|X - A^+Y\| = 0$ minimizando el resultado, esto es, queremos llegar al vector X que más se aproxime a la solución del sistema minimizando el problema de mínimos cuadrados siendo A^+ lo que entendemos por inversa generalizada[5].

Existen muchos métodos para llegar a la solución de la matriz pseudo inversa como el método de la diagonalización QR [6]. No obstante, cuando hablamos del cálculo computacional de dicha matriz, el método más utilizado pasa por un proceso llamado descomposición en valores singulares de la matriz (SVD siglas del inglés Singular Value Decomposition).

3.2. Descomposición en valores singulares (SVD).

El principio matemático de la descomposición en valores singulares de una matriz consiste en la expresión de la misma como un producto de varias matrices que cumplan ciertas propiedades:

$$A = \Sigma V^T \tag{2}$$

Hay que matizar que, en este trabajo, estamos tratando con matrices pertenecientes al cuerpo de los reales, si no fuese así, la traspuesta habría que entenderla como conjugación hermítica.

A es una matriz de dimensiones arbitrarias $m \times n$ y Σ una matriz diagonal de la misma dimensión que cuyos elementos de la diagonal principal distintos de cero se conocen como “valores singulares” de A .

Las matrices U y V estarán formadas por una base de autovectores ortonormalizados de las matrices cuadradas y simétricas AA^T y $A^T A$ respectivamente. Si tenemos una matriz cuadrada simétrica con un número finito de elementos reales como en nuestro caso, esta matriz entonces es diagonalizable con autovalores reales y podemos construir una base ortonormal de autovectores de dicha matriz. Esto se conoce como teorema de la descomposición espectral. Puede verse una prueba de este teorema en [7].

$$(AA^T)_{(m \times m)} = U_{(m \times m)} S_{(m \times m)} U_{(m \times m)}^T, \quad (A^T A)_{(n \times n)} = V_{(n \times n)} S'_{(n \times n)} V_{(n \times n)}^T \quad (3)$$

Del teorema de la descomposición de la unidad tenemos que U y V serán dos matrices ortonormales formadas por los autovectores de AA^T y $A^T A$ respectivamente y S y S' serán matrices diagonales con los respectivos autovalores en la diagonal asociados a los autovectores que forman U y V respectivamente. De (1) tendremos:

$$A^T = (U \Sigma V^T)^T = V^{TT} \Sigma^T U^T \quad (4)$$

Hay que tener en cuenta algunas propiedades de las matrices ortonormales y de las simétricas [8]:

1. Si U es una matriz ortogonal, entonces $U^T = U^{-1}$ con lo que $UU^T = U^T U = I$ (matriz identidad).
2. De lo anterior se deduce $U^{TT} = U$.
3. Y por tanto también $\det(U) = \pm 1 \rightarrow \det(UU^T) = \det(U) \cdot \det(U^T) = (\det(U))^2 = \det(I) = 1$
4. Si A es una matriz $m \times n$ de rango $r < n$ y $r < m$ entonces tanto $A^T A$ como AA^T serán semidefinidas positivas (todos sus autovalores son mayores o iguales a cero) y además, sus autovalores no nulos coincidirán.

De (2) se sigue que:

$$AA^T = U \Sigma V^T V \Sigma^T U^T = U \Sigma \Sigma^T U^T \quad A^T A = V \Sigma^T U^T U \Sigma V^T = V \Sigma^T \Sigma V^T \quad (5)$$

De aquí se deduce que $S = \Sigma \Sigma^T$ y $S' = \Sigma^T \Sigma$, con lo que siendo tanto S como S' como Σ diagonales, si denotamos a los elementos de Σ como (σ_{ij}) y a los de S y S' respectivamente (s_{ij}) y (s'_{ij}) , podemos escribir las relaciones: $s_{ii} = \sigma_{ii} \cdot \sigma_{ii}$ y $s'_{jj} = \sigma_{jj} \cdot \sigma_{jj}$ con (s_{ik}) y $(s'_{lj}) = 0$ para cada $i \neq k$ y $l \neq j$ respectivamente.

Esto nos lleva a que los valores singulares σ_{ij} con $i = j$ serán las raíces cuadradas de los autovalores de AA^T y de $A^T A$, que serán iguales a excepción de elementos nulos.

El estudio de la SVD nos lleva también a poder determinar cuál será el rango de la matriz A . Volviendo a tomar (1) es inmediato llegar a:

$$A = U \Sigma V^T \rightarrow U^T A V = \Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r, 0, 0, \dots)$$

Con lo que podemos determinar que el rango r de la matriz A será el número de valores en la diagonal de Σ distintos de cero, o lo que es lo mismo, el número de valores singulares de A . En general, cuando hablamos de matrices cuyo rango no coincide con $\min(m, n)$ siendo $m \times n$ la dimensión de la matriz ($r < \min(m, n)$), decimos que existe una deficiencia de rango. Esto suele darse en la mayor parte de los casos y complica el problema de calcular la pseudo-inversa. Cuando no hay una deficiencia de rango decimos que la matriz es de rango completo bien por filas si $r = m = \min(m, n)$ o por columnas si $r = n = \min(m, n)$.

El cálculo computacional de la SVD resulta la parte más costosa a la hora de construir la matriz pseudo-inversa. Un estudio detallado de cómo se puede llegar a computerizar este método puede encontrarse en [5] y [6]. En este trabajo por cuestión de tiempo se utilizó el algoritmo que tiene implementado el programa Matlab.

3.3. Pseudo-inversa a partir de SVD. Pseudo-inversa de Moore Penrose.

Llamaremos matriz inversa generalizada o matriz pseudoinversa, o pseudo-inversa de Moore-Penrose, en honor a los primeros que dieron una descripción matemática de ésta, a aquella matriz A^+ que teniendo el sistema $AX=Y$ resuelva $X=A^+ Y$. Por supuesto si la matriz A es invertible, la inversa generalizada debe coincidir con la inversa habitual. Este tipo de inversa generalizada cumple las siguientes condiciones:

1. $AA^+A = A$
2. $A^+AA^+ = A^+$
3. $(AA^+)^T = AA^+$
4. $(A^+A)^T = A^+A$

ref. [9][2][3]

Además, se puede demostrar la existencia y unicidad de la inversa generalizada para cualquier matriz. [4] Una vez hemos definido las propiedades de la pseudoinversa veamos cómo obtenerla a partir de la SVD.

Si ya tenemos descompuesta la matriz mediante valores singulares, podemos calcularla a partir de [4]:

$$A = U\Sigma V^T \longrightarrow A^+ = V^{-1}\Sigma^+(U^T)^{-1}$$

Ahora bien, siendo U y V matrices ortonormales sabemos que cumplen $U^{-1} = U^T$, $V^{-1} = V^T$ con lo que

$$A^+ = V^T \Sigma^+ U$$

Por su parte, la matriz Σ^+ se construye a partir de la matriz Σ , trasponiendo dicha matriz y calculando los inversos (recíprocos) de los valores singulares de A (σ_{ii} distintos de cero), con lo que $\Sigma^+ = \text{diag}(1/\sigma_1, 1/\sigma_2, \dots, 1/\sigma_r, 0, 0, \dots)$.

Aquí surge nuestro mayor problema con la pseudo-inversa. Al inicio de la sección, dijimos que el objetivo de definir la matriz pseudoinversa era el de poder resolver el sistema de ecuaciones generalizado $AX = Y$ como $X = A + Y$, podemos escribir esto mismo como $X = V^T \Sigma^+ UY$. Es fácil darse cuenta de que si tuviéramos algún valor singular muy cercano a cero, al hacer su recíproco obtendríamos un valor muy grande. Este hecho supone que una pequeña variación en el vector Y supondrá una variación enorme en el vector X . En el caso de la imagen médica, esto se traduce en que un ruido pequeño en los datos provoca unas fluctuaciones enormes en la imagen. [5]

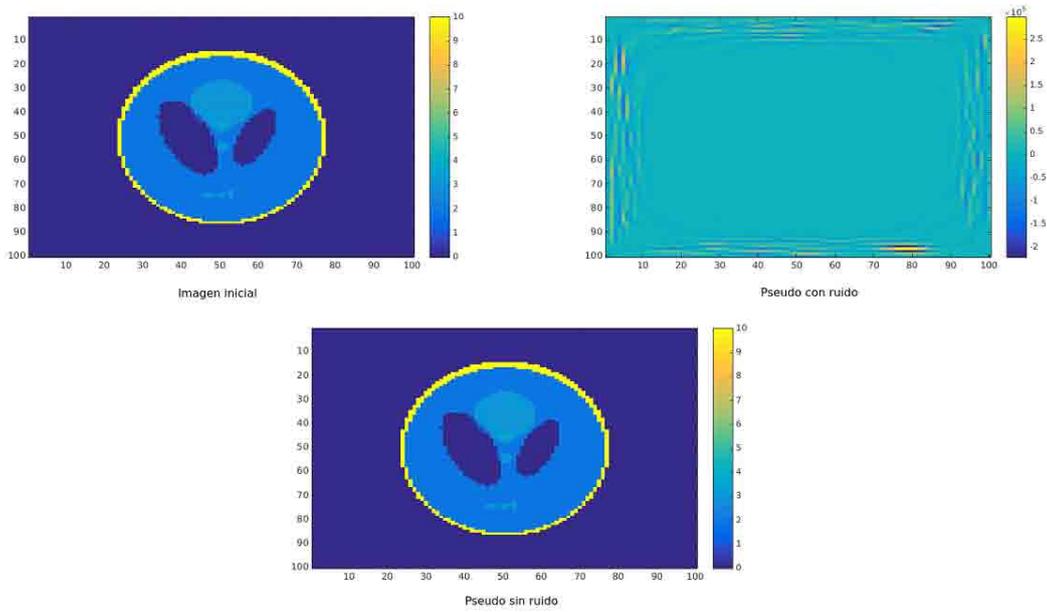
Surge entonces la necesidad de despreciar ciertos valores cuando son mucho más pequeños que el resto de valores singulares y hacerlos cero para evitar el error. El límite que establecemos a partir del cual hacemos cero los valores singulares es lo que se conoce como la tolerancia de la pseudo-inversa.

3.4. Reconstrucción de imagen con pseudo-inversa.

Hace años, el hacer imagen usando esta técnica se planteaba impensable debido a los tamaños tan grandes que tiene la matriz de respuesta del sistema (del orden de 1 millón x 1 millón de elementos). No obstante la computación ha avanzado considerablemente en estos últimos años y lo que antes era imposible ahora es tratable.

A pesar del coste computacional que supone el cálculo de la pseudo-inversa y la SVD, la pseudo-inversa se postula como un método realmente prometedor para hacer imagen ultra rápida, pues una vez hemos construido la matriz del sistema y la hemos invertido, podemos utilizar el mismo resultado para reconstruir infinidad de imágenes con un simple producto de matrices, mucho más rápido que los actuales métodos iterativos y mucho más resolutivo que los métodos analíticos al poder introducir modelos más realistas y físicos en la matriz. Para el caso de los analíticos necesitamos una geometría ideal.

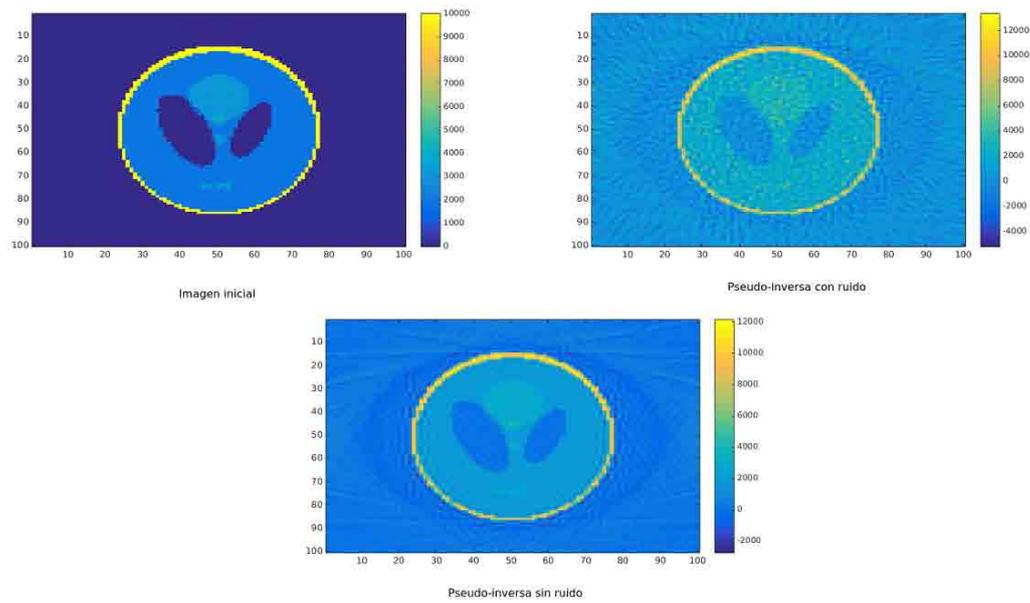
A la hora de hacer imagen con pseudo-inversa, es fundamental tener en cuenta la tolerancia que fijamos, podemos variar esta tolerancia en función del ruido que tenga la imagen para conseguir imágenes mejores. Así, en datos con poco ruido utilizaremos una tolerancia prácticamente nula mientras que cuanto mayor sea el ruido, mayor deberá ser también la tolerancia para evitar llegar a resultados del sistema muy alejados del real. Para ilustrar esto veamos un par de imágenes simples reconstruidas por Matlab con la función `pinv` que hace la pseudoinversa de una matriz, con una tolerancia muy baja para imágenes con y sin ruido:



Gráfica 19: Imágenes obtenidas con Matlab usando pseudoinversa con una tolerancia baja. Arriba a la izquierda vemos la imagen inicial de referencia. A la derecha no vemos nada debido a la pérdida de información por el ruido. Abajo vemos reconstrucción sin ruido del phantom de Matlab.

En Figure ?? se puede observar la imagen inicial de la que partimos, en este caso un phantom de matlab. De esta imagen, proyectando con la matriz de respuesta se obtienen los datos con los que se han reconstruido las otras dos imágenes. Estas imágenes se han reconstruido mediante pseudo inversa con una tolerancia muy pequeña para ambas. La imagen sin ruido sale perfecta dado que al no existir ruido y al estar utilizando la misma matriz para proyectar y retroproyectar apenas se ve afectada, pero en la imagen ruidosa se ha dejado de ver absolutamente nada debido a que la tolerancia era prácticamente cero y una variación por ruido de los datos nos ha llevado a este resultado.

Sin embargo, aumentando la tolerancia, podemos llegar a reconstruir una buena imagen para ambos casos, si bien a costa de perder cierta información y que por lo tanto la imagen perfecta y sin ruido de antes se vea ahora afectada también.



Gráfica 20: Imágenes obtenidas con Matlab usando pseudo inversa con una tolerancia alta. Fijamos más valores singulares como cero y por tanto la imagen se ve nebis afectada por el ruido. Arriba a la derecha la imagen ruidosa, abajo sin ruido que ha perdido resolución frente a gráfica ?? porque al aumentar la tolerancia perdemos algunos datos. Arriba a la izquierda la imagen inicial.

Como se puede observar en Figure 19 y en Figure 20, la imagen reconstruida con pseudoinversa puede contener valores negativos, y eso es algo que hay que tener muy presente si se quieren evitar errores a la hora de realizar ciertas reconstrucciones.

Para entender y ver la colocación de los detectores en el escáner se puede ver la sección 5.5, pues esa es la matriz que se ha venido utilizando para reconstruir tanto con pseudo inversa como con los métodos iterativos.

3.5. Bibliografía.

Referencias

- [1] https://www.cs.bgu.ac.il/~na131/wiki.files/SVD_application_paper.pdf
- [2] https://inst.eecs.berkeley.edu/~ee127a/book/login/def_pseudo_inv.html
- [3] <http://www.math.ucla.edu/~laub/33a.2.12s/mppseudoinverse.pdf>
- [4] JOÃO CARLOS ALVES BARATA. MAHIR SALEH HUSSEIN. BRAZ J PHYS (2012) 42:146–165 DOI 10.1007/s13538-011-0052-z. *The Moore–Penrose Pseudoinverse: A Tutorial Review of the Theory*.
- [5] G. GOLUB AND W. KAHAN. *Calculating the singular values and pseudo-inverse of a matrix*.

- [6] GENE H. GOLUB. CHARLES F. VAN LOAN. *Matrix Computations*.
- [7] <http://web.mit.edu/jorloff/www/18.03-esg/notes/symmetricMatrices.pdf>
- [8] <http://www.ugr.es/~falvarez/matrices.pdf>
- [9] JAMES, M. (JUNE 1978). "The generalised inverse". *Mathematical Gazette*. 62: 109–114. doi:10.2307/3617665.
- [10] E. E. OSBORNE. SPACE TECHNOLOGY LABORATORIES. LOS ANGELES. CALIFORNIA. DOI: 10.1145/321088.321100 *On Least Squares Solutions of Linear Equations*.
- [11] J.L.HERRAIZ, J.J.VAQUERO, J.M.UDÍAS *FBP Reconstruction of Sinograms with Gaps Basen on the Inversion of a Perturbed Matrix*

4. Reconstrucción de imagen en 3D.

4.1. Reducción del problema 3D a 2D.

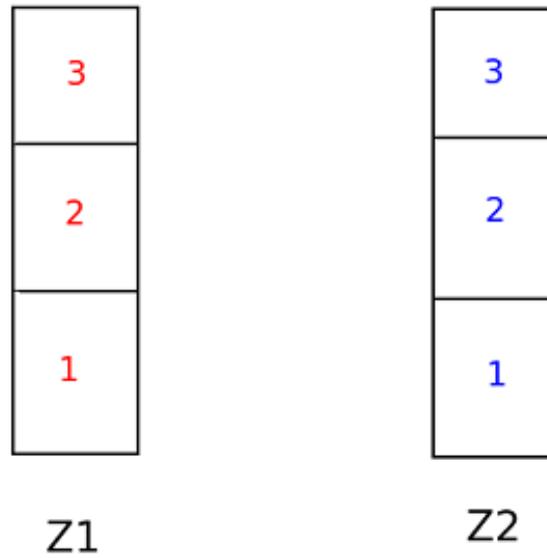
Cuando tenemos un problema de reconstrucción en 3D, la situación se complica con respecto a la reconstrucción 2D. En general se hace muy difícil abordar el problema tridimensional de manera directa, por lo que se suele optar por una reducción del problema a muchos problemas de dos dimensiones. Es decir, vamos a separar la imagen global en un número finito de cortes o rodajas en la coordenada z de un escáner con forma cilíndrica. Es lo que llamamos rebineado en z .

La manera más simple y la más común de hacer este rebineado se llama SSRB (Single Slice ReBinning), existen otras más sofisticadas como FORE (FOurier REbinning). En este trabajo hemos tratado de implementar una forma mejor de rebineado utilizando la propia información de la matriz de respuesta del sistema y a partir de la pseudoinversa. De momento, antes de hablar más en profundidad de los métodos de rebineado, es necesario hablar de la manera en que los datos se almacenan tridimensionalmente.

4.2. Span. MRD. Michelograma.

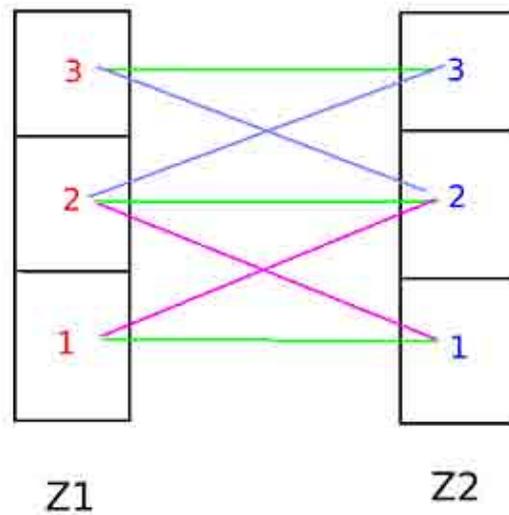
En la sección 1.2, hemos visto como se almacenan los datos bidimensionales para la reconstrucción de la imagen. El modo lista y el histograma de LORs pueden generalizarse sin mayor problema para el caso tridimensional. El caso del sinograma, aunque también se podría generalizar a una tercera dimensión, resulta mucho menos claro.

A la hora de tener o no en cuenta las coincidencias generadas por cristales enfrentados hay que considerar dos conceptos que a veces resultan confusos. Para tratar de explicarlos con más claridad imaginemos un escáner en el que tenemos tres cristales enfrentados a otros tres:



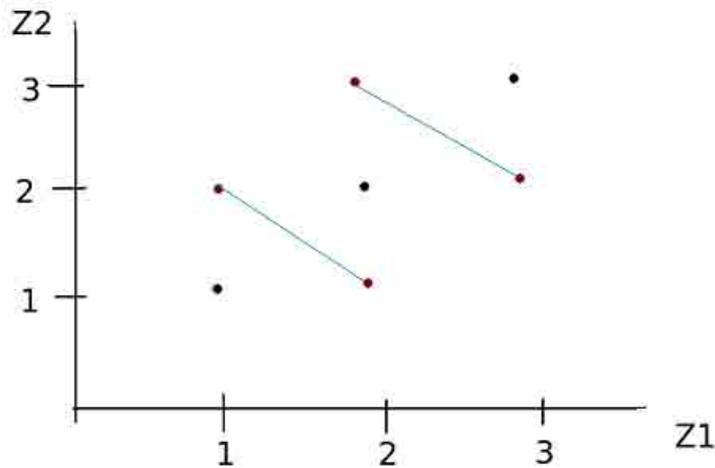
Gráfica 21: Escáner simplificado

Lo que se conoce como máxima diferencia de anillos o MRD del inglés Maximum Ring Difference será la máxima inclinación que permitiremos de una línea de respuesta para tenerla en cuenta, esto es, la diferencia máxima que permitimos entre el número de cristal de $Z2$ y el número de cristal de $Z1$. Así, en nuestro ejemplo, una MRD de 1 nos permitiría considerar las cuentas de $Z1 = 1$ con $Z2 = 2$ y $Z2 = 1$ así como $Z1 = 2$ con $Z2 = 1, 2$ ó 3 y $Z1 = 3$ con $Z2 = 2$ ó 3 . Una MRD=0 equivaldría a tomar sólo las cuentas de los cristales enfrentados. Llamamos a estas coincidencias directas. Supongamos que estamos permitiendo una MRD de 1, por lo que consideramos las coincidencias cruzadas entre anillos así como las directas. A las coincidencias del cristal 2 en $Z1$ con el cristal 1 en $Z2$ y a las del cristal 1 en $Z1$ con las del cristal 2 en $Z2$ las podemos asignar a un mismo punto. Esto es, podemos sumar ambas coincidencias por decirlo de alguna forma provenientes de LORs con igual pendiente:



Gráfica 22: MRD=1

Una forma práctica de expresar esto es con una representación llamada Michelograma, muy útil a la hora de considerar todas las posibles combinaciones de cristales.

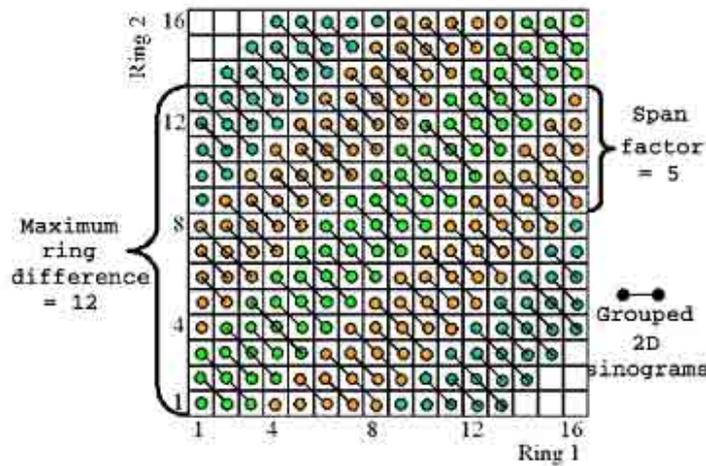


Gráfica 23: Michelograma MRD=1. 3 cristales

Dado que estamos combinando puntos donde sumamos un solo dato (los directos) con otros donde estamos sumando dos (los cruzados) decimos que tenemos $span = 2 + 1 = 3$. Luego el span nos da una idea de cuantos puntos, o mejor dicho, cuantas LORs sumamos o combinamos a la hora de interpretar los datos.

En Figure 4 puede verse un Michelograma más detallado con una MRD y un span más altos.

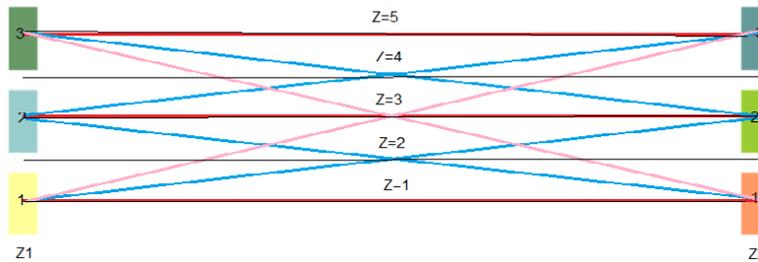
En [1] puede verse una descripción detallada del span, los Michelogramas y la MRD.



Gráfica 24: Michelograma

4.3. SSRB.

El método SSRB (Single Slice ReBinning) consiste simplemente en asignar a cada par de cristales (Z_1, Z_2), una coordenada nueva $Z = Z_1 + Z_2 - 1$. De este modo los datos quedarán rebinados en un total de $(2\Delta n^{\circ} \text{cristales} - 1)$ rodajas en la coordenada z . A la hora de construir el SSRB hay que tener en cuenta tanto la MRD como el span para saber los datos de qué pares de cristales asignar a cada rodaja. En nuestro código hemos tomado tanto una MRD como un span máximos de forma que permitimos la combinación de todas las líneas con todas en (Z_1, Z_2).



Gráfica 25: SSRB

Este método es muy sencillo y fácil de implementar. Además para fuentes en el centro del escáner funciona muy bien. No obstante cuando tenemos imágenes más complejas o fuentes descentradas, el método no se comporta tan bien como nos gustaría dado que no tenemos ningún tipo de información de si la cuenta ha venido de un punto más cercano a Z_1 o a Z_2 .

Existe otro método llamado FORE (FOurier REbinning) que se basa en transformadas de Fourier y da más información que SSRB al respecto del rebinado de los datos. En las referencias puede encontrarse bibliografía al respecto pero dado que no implementamos este método durante el trabajo no le dedicaremos más espacio.

En este trabajo, proponemos realizar el rebineado a partir de la información que suministra la matriz de respuesta del sistema y su pseudoinversa. Así resolviendo el sistema:

$A_z \cdot \text{Datos}_{\text{Rebineados}}(X1, Y1, X2, Y2, Z) = \text{Datos}(X1, Y1, Z1, X2, Y2, Z2)$ por medio de la pseudo inversa A_z^+ , se puede obtener $\text{Datos}_{\text{Rebineados}} = A_z^+ \cdot \text{Datos}$.

4.4. Bibliografía.

Referencias

- [1] FREDERIC H. FAHEY, DSC PET CENTER, WAKE FOREST UNIVERSITY SCHOOL OF MEDICINE, WINSTON-SALEM, NORTH CAROLINA. *Data Acquisition in PET Imaging* .
<http://www.guillemet.org/irene/coursem/DataorganizationinPET.pdf>
- [2] MICHEL DEFRISE AND PAUL KINAHAN. *Data Acquisition and Image Reconstruction for 3D PET*.
- [3] MICHEL DEFRISE, PAUL E KINAHAN AND CHRISTIAN J MICHEL. *Image Reconstruction Algorithms in PET*.
<http://eknygos.lsmuni.lt/springer/370/63-91.pdf>

5. Reconstrucción de datos PET basado en la pseudoinversa.

5.1. Objetivo de la investigación.

Este proyecto ha estado orientado al desarrollo de un software original que permita la rápida reconstrucción de imágenes médicas de escáneres PET (generalizable a otras modalidades) mediante el uso de la pseudoinversa. Al ser imágenes en 3D, también se ha desarrollado un código rebineador cuyo resultado esperado es mucho mejor que el usual SSRB e incluso otros más sofisticados como FORE.

La forma de obtener las matrices pseudo-inversas ha sido mediante la función `pinv` (pseudo-inversa) que lleva incorporado el programa Matlab.

El primer paso a la hora de poder trabajar con la pseudoinversa es el de escribir la matriz de respuesta del sistema, la primera parte del código desarrollado está precisamente orientado a ello. Una vez construida esta, mejor dicho, estas, pues es necesario escribir una matriz para el rebineador y otra para el reconstructor de rodajas, podemos pasar a pseudoinvertirlas para finalmente usar estas pseudoinversas en la reconstrucción.

5.2. Matriz de respuesta del sistema. Geometría del escáner Superargus.

Como punto de partida, hemos tomado el escáner SuperArgus construido para PET preclínico en pequeños animales de la empresa SEDECAL. Este escáner está compuesto por cuatro anillos de cristales centelleadores, cada anillo a su vez formado por 13 filas de cristales (rows en el código) . Entre cada anillo existe un gap de un tamaño aproximado de 4 cristales.



Gráfica 26: Escáner SuperArgus

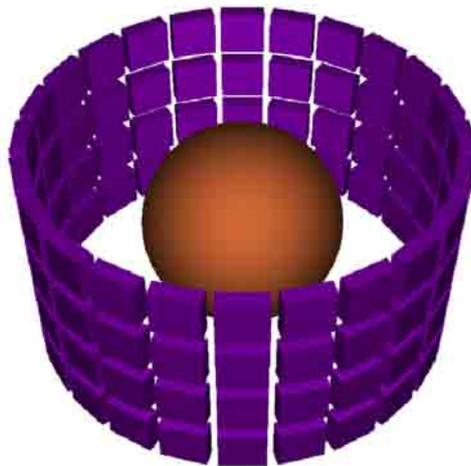
Cada anillo del escáner, está formado a su vez por 24 bloques con 13 columnas de cristales

cada uno. Cada uno de los cristales del aparato está formado por dos capas de diferente material, lo que conocemos como phoswich.

Este phoswich está pensado para mejorar la imagen del escáner en base a poder determinar correctamente la profundidad de interacción del fotón con el cristal, como se comentó en la sección 1.5 y así mejorar la información al respecto de la inclinación de la LOR correspondiente.

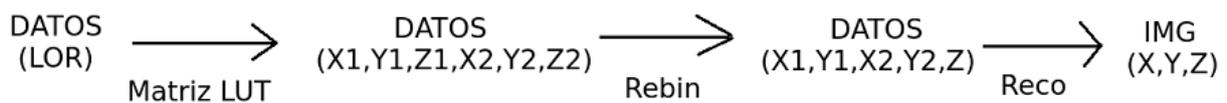
Por simplicidad al desarrollar el código de momento hemos supuesto una sola capa de phoswich, aunque el programa está escrito de tal forma que añadir la segunda capa resulta muy sencillo.

Las matrices de respuesta del sistema se han escrito en base a esta geometría de manera que describan de la manera más correcta posible cada una de las probabilidades de que cada LOR atraviese una serie de vóxeles. A la hora de construir estas probabilidades se han tenido en cuenta efectos como la no colinearidad de los fotones y el rango de los positrones. Esto hace que las líneas de respuesta en realidad tengan un cierto ancho formando algo así como tubos de respuesta.



Gráfica 27: Simulación escáner SuperArgus

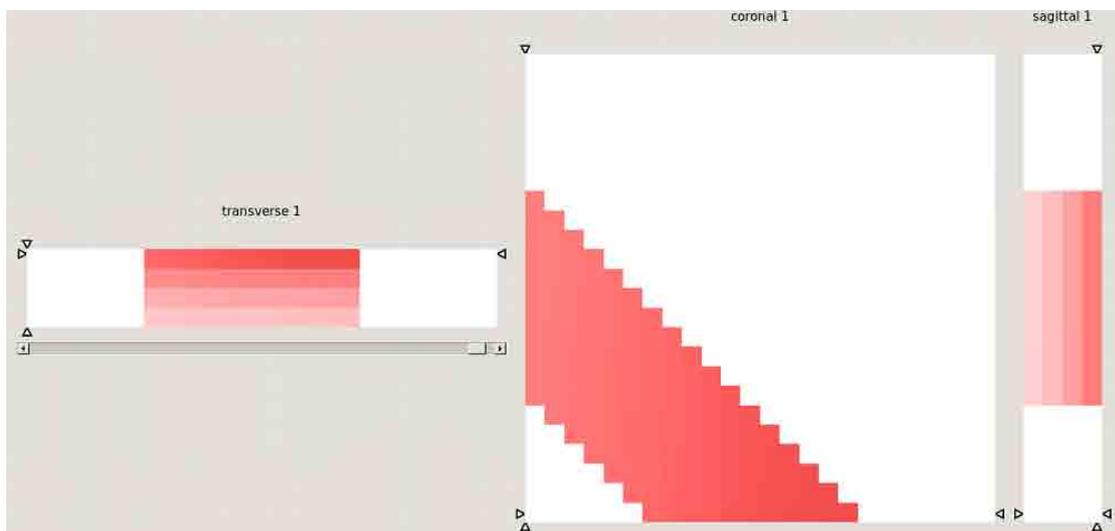
Para poder llegar a la reconstrucción final de una imagen tridimensional debemos realizar una serie de pasos: ordenar los datos primero, rebinearlos después y finalmente reconstruir cada rodaja del rebineado:



5.3. Código LUT. (LookUp Table).

Cuando tomamos una serie de datos para reconstruirlos, estos no suelen estar en el orden deseado, que en este caso sería primero los datos de las LORs correspondientes a las z 's para rebinear y después los de cada rodaja para reconstruir. Como no es el caso, tenemos que buscar otra herramienta para reordenar nuestros datos antes de reconstruirlos.

En nuestro caso hemos utilizado datos sacados del simulador PENELOPET. Trabajaremos con los datos dentro del histograma de LOR's completo. Es importante tener la información del orden en el que se han grabado los datos utilizados. Recorriendo el bucle del orden de cada LOR, asignamos a cada par de cristales ($X1, Y1, Z1, X2, Y2, Z2$) un contador que será la posición de la línea de respuesta.



Gráfica 28: Matriz LUT

En Figure [9] se puede observar la matriz de ordenamiento de las líneas de respuesta. En el eje x se observan las coordenadas correspondientes a x e y del primer cristal ($XY1$), en el eje y la coordenada z del primer cristal ($Z1$) mientras que en el eje z (vista coronal) se observan las coordenadas x e y para el cristal enfrenteado ($XY2$) y en las distintas gates (ventanas de amide) se colocan las coordenadas z del segundo cristal ($Z2$). En la imagen se observan todas las gates para tener una idea intuitiva de la forma que tiene la matriz.

El objetivo es que a través de la ordenación que tiene la matriz LUT podamos ordenar los datos del histograma asignándoles el orden correspondiente en el bucle.

5.4. Código rebineador

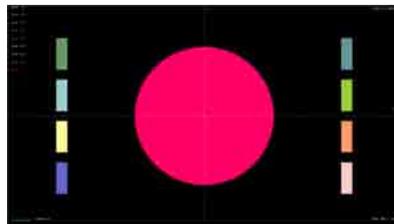
El objetivo primero del código es el de reducir el problema de 3D a uno de 2D habitual. Para ello en vez de utilizar SSRB que da unos resultados bastante pobres o un FORE que al basarse en la transformada de Fourier se comporta mal en presencia de gaps utilizaremos la propia pseudoinversa como rebineador. Muy parecido a este método podemos encontrar métodos iterativos cuyos resultados son equiparables a utilizar la pseudoinversa, no obstante tenemos los inconvenientes

de todo método iterativo, sobre todo el coste en tiempo que requiere para converger a la solución.

Para ello, teniendo nuestro escáner un total de 52 filas (4 anillos con 13 cristales cada anillo), escribiremos una matriz en base a esta geometría. Presenta ventajas interesantes frente a otros métodos como es el hecho que podemos tener en cuenta los huecos entre anillos (gaps) sin que eso estropee la imagen.

En nuestro código hemos supuesto un MRD máximo, esto es, tenemos en cuenta las coincidencias de todos los cristales con todos los posibles enfrentados.

Podemos ver el escáner como un cilindro de una cierta altura, ordenando los cristales del 1 al 52 empezando por el anillo más “abajo” en el cilindro. Para verlo gráficamente lo mejor sería observar la imagen de un corte longitudinal:



Gráfica 29: gview2D

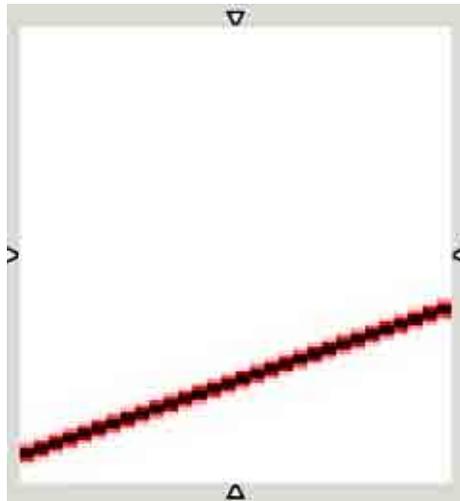
La esfera en el medio del escáner forma parte de una simulación en PENELOPE y sería el objeto definido para enmarcar el FOV de la simulación. A los lados pueden verse los cuatro anillos separados por gaps, enfrentados a otros cuatro. Cada uno de estos sabemos que a su vez está compuesto por 13 filas de cristales.

A cada par de cristales, el programa les asignará unas coordenadas a las que llamaremos $Z1$ y $Z2$ (vertical) y $T1$ y $T2$ (horizontal) y que tendrán su origen en el centro geométrico del escáner. Así, los cristales que entendemos por encima del origen tendrán su coordenada z positiva y los que estén por debajo negativa al igual que la coordenada t se entenderá positiva para la columna de cristales a la derecha y negativa para los de la izquierda. Hay que entender que ésto sólo es una simple herramienta para poder después de manera sencilla calcular la distancia de cada LOR a cada vóxel. Una vez tenemos bien ordenados los cristales podemos determinar las líneas de respuesta de cada par de cristales, y para cada una de las líneas de respuesta la distancia a todos los posibles vóxeles de la imagen.

En nuestro caso el número de vóxeles que elijamos para z determinará el número total de rodajas que tendrá la imagen final, mientras que el que elijamos para t contribuirá a una mejor colocación de la información de la probabilidad de que cada LOR pase por un cierto vóxel. Para que la matriz no sea excesivamente grande de momento elegimos 127 vóxeles en z y 30 en t , si bien el número de vóxeles en los que dividamos la imagen depende de nuestro criterio y puede ser modificado en cualquier momento.

Una subrutina se encarga, una vez está determinada la línea de respuesta, de ir recorriendo todos los vóxeles de la imagen asignándoles una probabilidad de pertenecer a la línea en cuestión. Dicha probabilidad dependerá de la distancia a la que se encuentren línea y vóxel pero por motivos de emborronamiento por rango de positrón y no colinearidad dijimos que estas líneas debían tener una cierta anchura por lo que le asignamos un ancho que en principio tomaremos gaussiano, aunque en el futuro podemos modificar para ser más realista con la situación física.

Estas probabilidades que hemos ido construyendo son las que determinarán las coordenadas de la matriz de rebineado $A(iV, iLOR)$. Nótese que tal y como estamos utilizando la notación, esta matriz en realidad se trata de la traspuesta de la matriz del sistema $AX = Y$ puesto que X tiene dimensiones de iV (vóxeles de la imagen) e Y dimensiones de $iLOR$ (esto es, datos o líneas de respuesta). A, por tanto, debería tener dimensiones $A(iLOR, iV) \cdot X(iV) = Y(iLOR)$. Por comodidad y para que sea más intuitivo a la hora de dibujarlo con AMIDE escribimos $A(iV, iLOR)$ en vez de $A(iLOR, iV)$.



Gráfica 30: Línea de respuesta del rebineador z,t .

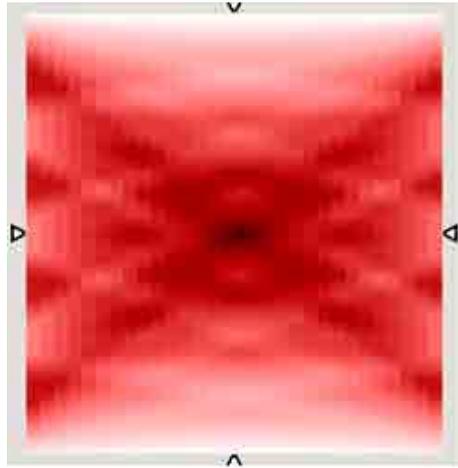
En la gráfica 30 podemos ver una línea de respuesta generada por la matriz del sistema. Podemos representarlas todas juntas para ver el aspecto que tiene la distribución de LORs:

Aquí podemos observar las 52×52 (esto es 2704) LORs de rebineador.

Una vez la matriz está construida, podemos pseudoinvertirla. Ahora si tenemos el sistema:

$AX = Y$, entonces $X = A^+Y$. Donde A^+ es la pseudoinversa de la matriz de respuesta. En este caso Y serán los datos de coincidencias histogramados con $(X1, Y1, Z1, X2, Y2, Z2)$ con los cristales 1 y 2 enfrentados. Ahora X son los datos rebineados, lo cual quiere decir que para cada par $(Z1, Z2)$ le asigna un par (z, t) , de manera que quedan los datos $(X1, Y1, X2, Y2, z, t)$

La coordenada t , aparte de permitirnos determinar con precisión a que lado de la imagen le corresponden los datos nos permitiría en un futuro hacer reconstrucción teniendo en cuenta una especie de TOF (Pseudo Time Of Flight). De momento para el presente trabajo sólo utilizare-



Gráfica 31: Distribución de líneas de respuesta del rebineador.

mos la coordenada t como más información para repartir mejor la información en las rodajas correspondientes. Si eligiéramos $t = 1$ sería equivalente a hacer SSRB. Para agilizar el proceso posterior, una vez pseudoinvertida la matriz (ya utilizada la información que da t), sumaremos todas las t dejando los datos rebineados en $(X1, Y1, X2, Y2, Z)$ ya sin la coordenada t . Hacer esto nos permite reducir el tiempo posterior de cálculo en un orden del número de vóxeles en t veces menos.

La coordenada z determinará el número de rodajas que tendrá nuestra imagen final.

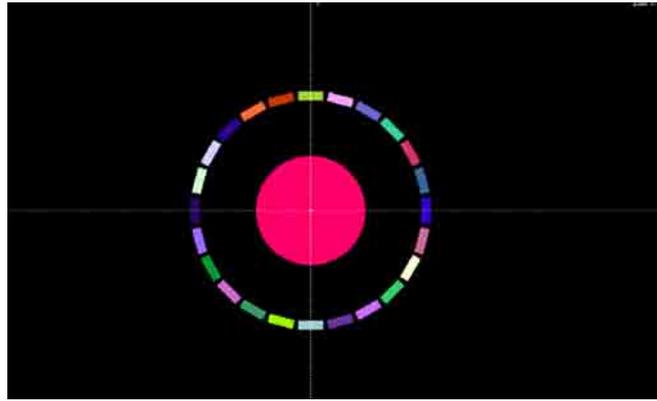
5.5. Código reconstructor de rodajas.

Una vez tenemos los datos rebineados y repartida la información en las rodajas correspondientes, podemos pasar a reconstruir cada una de estas rodajas usando otra matriz pseudoinversa. Para ello es necesario escribir ahora una matriz de respuesta del sistema para las coordenadas (x, y) de los cristales.

Ahora estamos tratando con rodajas circulares del escáner lo que dificulta ligeramente la asignación de coordenadas. El procedimiento utilizado es el siguiente: primeramente asignamos unas coordenadas (x, y) a un cristal concreto de un bloque concreto que nos servirá de referencia. Para todo el bloque vamos dando las coordenadas a cada cristal en función de las de referencia. Para determinar las coordenadas del resto de bloques simplemente rotamos las que ya tenemos y vamos asignando las de los demás. El ángulo que tendremos que girar para pasar de un bloque a otro será 360 grados entre el número de bloques.

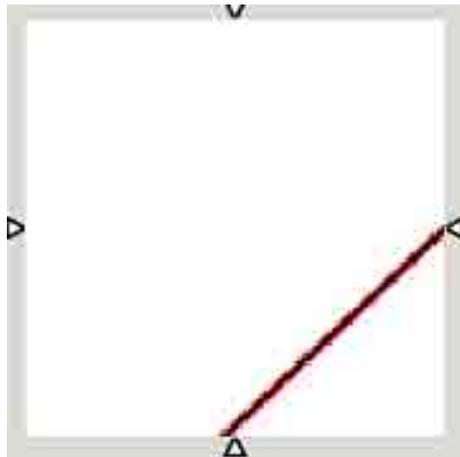
Al final lo que se pretende caracterizar será un corte transversal del escáner. Para hacernos una idea:

En la imagen podemos ver los 24 bloques de cristales que forman el escáner. Por el momento, y por simplicidad, los huecos entre cada bloque no se han tenido en cuenta. Por ello la matriz quedará ligeramente distorsionada.



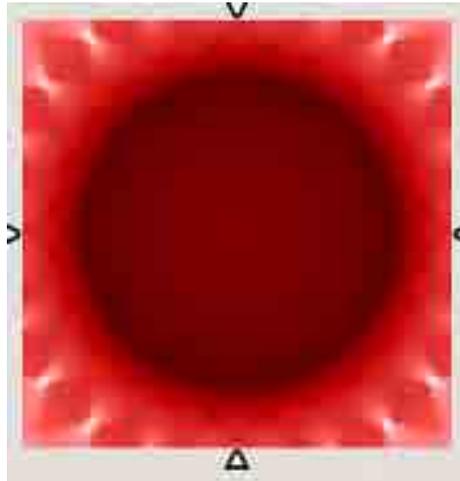
Gráfica 32: Corte transversal escáner

Esta matriz, al igual que la del código rebineador, habrá asignado unas probabilidades de que cada LOR pertenezca a un vóxel determinado. En la imagen de abajo podemos observar una línea de respuesta de la matriz.



Gráfica 33: LOR de la matriz xy

También podríamos ver todas las LORs resultando lo siguiente:



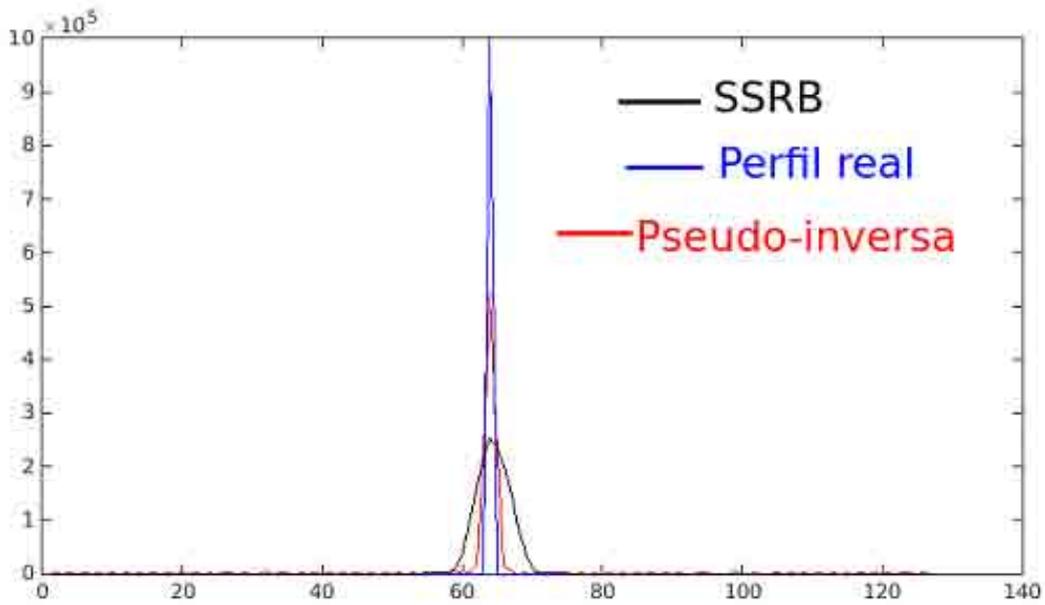
Gráfica 34: LORs de la matriz xy

Este es el resultado final de la matriz para un caso de 100 vóxeles en x y otros 100 en y .

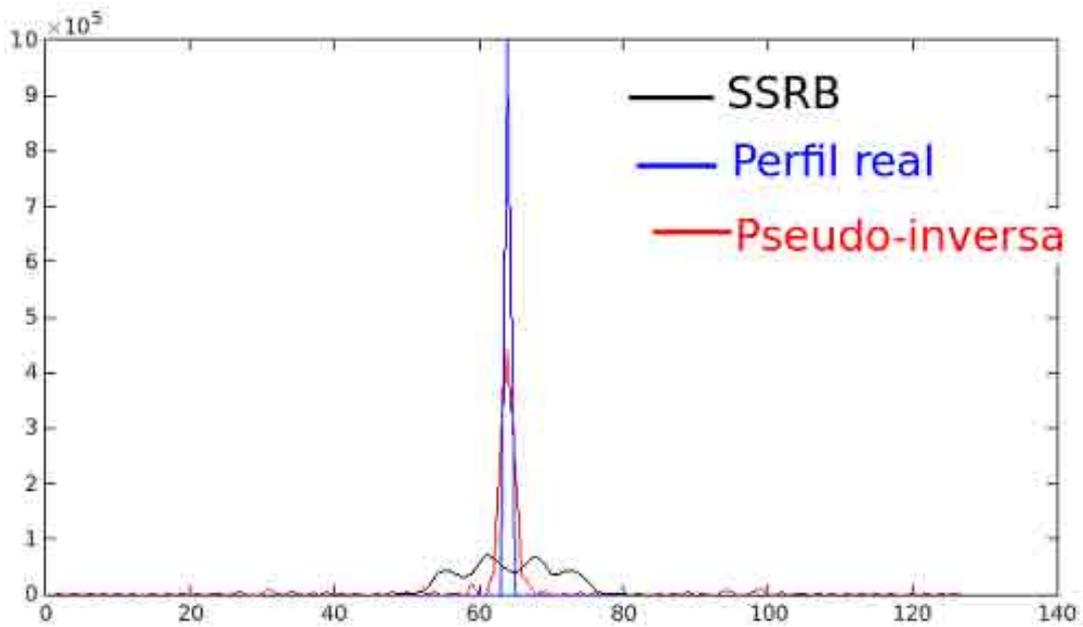
Ahora nuevamente quedará pseudoinvertirla, y multiplicando esa matriz $A_{xy}^+(iV, iLORs)$ donde $iLORs$ es el número de líneas de respuesta para cada rodaja, que en este caso y con la distribución elegida será de 22308 (12x11 bloques en coincidencia $x13x13$ cristales enfrentados para cada par de bloques) e iV será el número de vóxeles que elijamos para la imagen final. Vamos a intentar conseguir que cada vóxel cubra un tamaño real de $0,5mm$, y también intentaremos que el campo de visión esté lo más amplio posible. Lo óptimo será al menos construir imágenes de $175x175$ pero el coste computacional para hacer la pseudo inversa aumenta mucho con el tamaño de la matriz por lo que para la realización del trabajo hemos estado usando algunas más pequeñas. De cara a hacer imagen real sería conveniente hacerlas al menos de ese tamaño.

5.6. Resultados y discusión.

Empezaremos rebineando los datos generados por una fuente puntual centrada utilizando nuestro método. Compararemos con SSRB que resulta muy fácil de implementar. Para el caso de una fuente puntual centrada, SSRB se comporta muy bien, pues no hay información que perder acerca de la distancia al centro de coordenadas. Para otras fuentes, SSRB no se comportará tan bien. Podemos ver en las siguientes gráficas la diferencia del perfil rebineado para fuentes situadas en distintos puntos de la imagen:



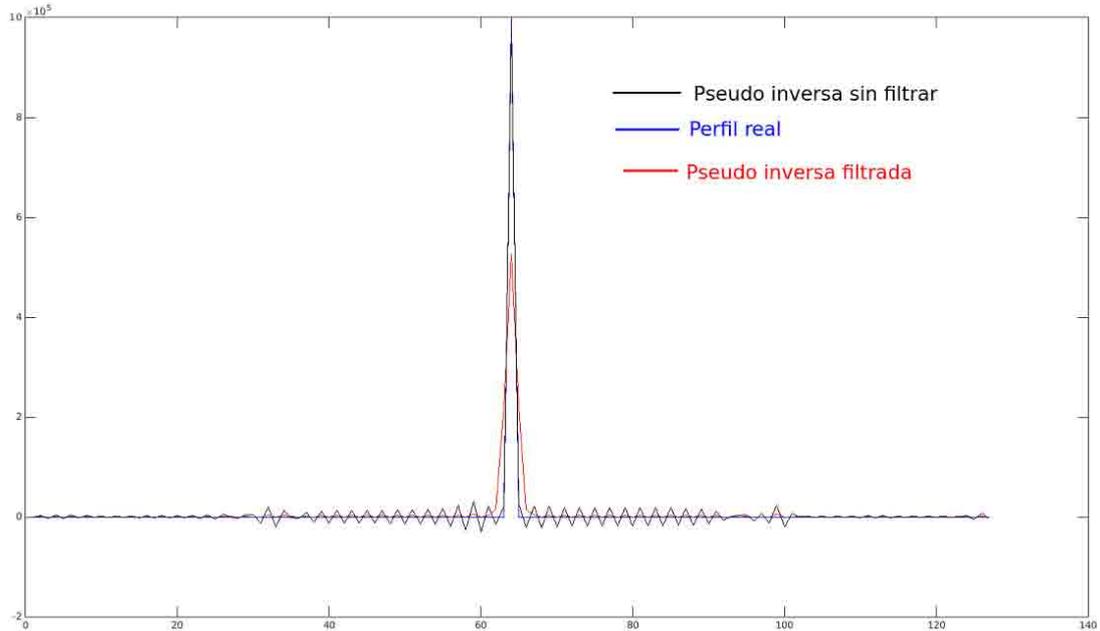
Gráfica 35: Perfil de rebineado de una fuente puntual centrada en el escáner. En el negro el perfil de SSRB, al ser centrada se ajusta bien, pero no tan bien en ningún caso como el rebineado con pseudo inversa (rojo) al perfil real (azul)



Gráfica 36: Fuente descentrada. En este caso el perfil de rebineado de SSRB (negro) no funciona adecuadamente. La pseudoinversa (rojo) por el contrario sigue siendo un buen método y se ajusta bastante bien al perfil real (azul).

Para poder llegar al perfil de la pseudo inversa que podemos ver en Figure [9] y Figure [10]

fue necesario suavizar el perfil directo de la pseudo inversa. En Figure [11] podemos ver el perfil tal y como sería antes de ajustarlo:

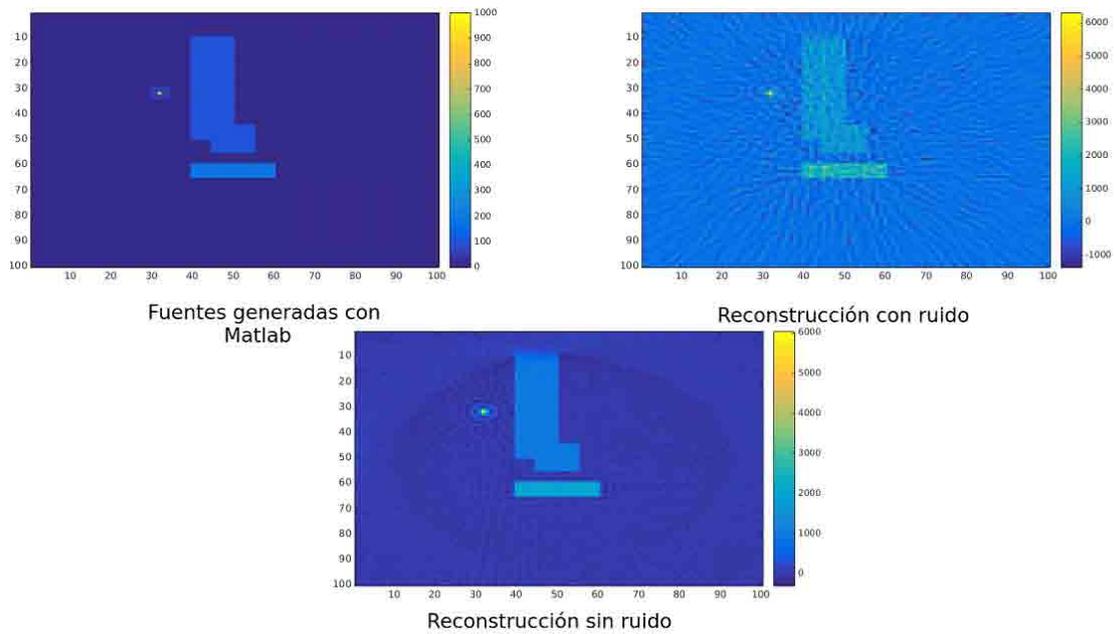


Gráfica 37: En negro pseudo inversa sin filtrar donde se pueden apreciar fluctuaciones que desaparecen al aplicar un filtro gaussiano (rojo). En azul el perfil real.

Vemos que aplicar directamente la pseudoinversa, por cuestiones estadísticas a la hora de intentar recuperar frecuencias donde no hay datos, devuelve esos artefactos que vemos en la imagen. Aplicando un filtro, por ejemplo gaussiano que promedie los valores de alrededor se consigue un resultado mucho mejor. De todas formas deberá existir un valor de la tolerancia para el cual no aparecerían estos artefactos, pero al no poder elegir siempre la tolerancia que nos gustaría por cuestiones de ruido tenemos que recurrir a estos filtros que restauren una solución correcta para el rebineado.

En cuanto al tiempo de cálculo de la matriz pseudo inversa para el rebineador de dimensiones realistas, como es el caso no lleva más de un minuto con Matlab. Además una vez se tiene la matriz pseudoinversa previamente calculada el cálculo del rebineado se reduce a muy pocos segundos.

En cuanto a la reconstrucción de rodajas en dos dimensiones, el cálculo de la matriz pseudo inversa es lo que más tiempo lleva, al ser más grandes las matrices tanto en vóxeles como en líneas de respuesta que en el caso del rebineado. Por ilustrarlo con un ejemplo, una matriz de 100x100 vóxeles en la imagen final tarda del orden de 6 minutos en ser pseudo invertida con Matlab. Si la imagen fuera de 325x325 como esperamos llegar a generar, el tiempo de cálculo se haría mucho más grande, así como los recursos que consumiría el ordenador. Una vez tenemos la matriz invertida, el proceso de reconstrucción de cada rodaja supone del orden de medio segundo, mucho menos que cualquier método iterativo, por lo que esperamos tener imágenes 3D realistas en tiempos de pocos minutos.



Gráfica 38: Fuentes generadas en Matlab reconstruidas arriba a la derecha añadiendo ruido y abajo sin ruido utilizando la matriz del sistema para el reconstructor de rodajas descrita en la sección 5.5 del trabajo

5.7. Aproximación a la solución directa del problema en 3D.

A priori, podría parecer lógica y más sencilla la resolución del problema mediante el uso de una sola matriz que recogiera la probabilidad de que todas y cada una de las posibles líneas de respuesta y las asociase a cada uno de los vóxeles de la imagen. Esto nos evitaría el paso del rebineado para reducir el problema $3D$ a uno $2D$ resolviendo directamente el problema en tres dimensiones.

A día de hoy, ese problema resulta impensable por el tamaño de la matriz que generaríamos. Supongamos el escáner que hemos descrito en los puntos anteriores. Para hacernos una idea vamos a tener en cuenta sólo las líneas de respuesta que hemos tomado en los puntos del rebineado y del reconstructor de rodajas, esto es, en el eje Z 52×52 líneas de respuesta y en cada plano XY un total de 11 anillos enfrentados lo que son $12 \times 13 \times 11 \times 13$ posibles combinaciones. Multiplicando ambas tendremos el total de líneas de respuesta que deberíamos tener en cuenta en este caso donde recordamos no estamos teniendo en cuenta el phoswhich lo que multiplicaría por 4 las combinaciones posibles de cristales. Con todo, obtenemos un total de 60320832 líneas de respuesta que de hecho son las que tienen los ficheros de datos que hemos usado. Ahora bien, si no usáramos el rebineado primero, tendríamos que asignarle a cada una de estas líneas la probabilidad de que atravesase cada LOR. Vamos a suponer las mismas condiciones que con el rebineador y con el reconstructor de rodajas, esto es, 127 vóxeles en z y cogiendo un tamaño de vóxel exagerado 100×100 vóxeles en xy . Esto nos dejaría un total de 1270000 vóxeles en la imagen. Multiplicando por las LORs, la dimensión total de la matriz de respuesta que se obtiene es del orden de $7,66 \cdot 10^{13}$ elementos. Si almacenamos esta matriz en formato binario y real*4 que sería lo habitual, el total de bytes de memoria real que ocuparía la matriz es de $3 \cdot 10^{14}$, o lo que es lo mismo una matriz de prácticamente 300 terabytes. Ahora quedaría el trabajo de

pseudoinvertir semejante matriz y utilizarla para sacar un resultado.

Si la computación sigue desarrollándose al ritmo actual quizás en no mucho tiempo podamos estar trabajando con matrices de ese calibre pero por el momento supera las capacidades de los ordenadores actuales.

6. Conclusiones y expectativas de futuro.

A lo largo de este Trabajo de Fin de Máster, hemos visto distintos métodos para afrontar la reconstrucción de imágenes tanto en 2D como en 3D, y hemos propuesto utilizar la pseudoinversa tanto para el rebineado como para la posterior reconstrucción de rodajas.

Hemos comprobado que la pseudoinversa presenta claras ventajas frente a SSRB a la hora de hacer el rebinning del problema y también lo hace frente a FORE por no tratarse de un método analítico y poder introducir información de la física del problema. Esperamos que nos permita en un futuro mejorar aún más el rebineado utilizando pseudo TOF con toda la información de la matriz y sin colapsar en la componente horizontal.

Se está desarrollando un código en FORTRAN que permitirá de manera sencilla reconstruir datos completos en 3 dimensiones. Cuando el programa esté optimizado esperamos que los tiempos de obtención de las imágenes sean al menos igual de rápidos que usando métodos analíticos y obtener resultados igual de buenos o mejores que utilizando métodos iterativos. El objetivo final sería poder hacer imagen en tiempo real utilizando este método.