UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Análisis y Desarrollo de Algoritmos de Altas Prestaciones para Reconstrucción de Imagen Médica TAC 3D basados en la Reducción de Dosis

TESIS DOCTORAL

Autor:
Mónica Chillarón Pérez

Directores:
Prof. Vicente E. Vidal Gimeno
Prof. Gumersindo J. Verdú Martín

Septiembre 2021

A mi familia,
mis logros también son vuestros.

Agradecimientos

Al final de este camino me gustaría agradecer a todas las personas que han formado parte de esta etapa y han hecho posible con su apoyo que hoy pueda estar dedicándoles estas palabras.

En primer lugar, quiero agradecer a mis directores, los Profesores Vicent Vidal y Gumersindo Verdú, por todo su apoyo y dedicación. Gracias a ellos me adentré en el campo de imagen médica, e inicié mi carrera como investigadora con todos los medios que pusieron a mi disposición. Sus ideas, orientación y la ayuda que siempre me han brindado cuando he necesitado ha sido indispensable para el desarrollo de esta tesis. Además, me gustaría destacar no solo su apoyo profesional, sino también personal, lo cual agradezco enormemente ya que me ha hecho posible sobrellevar los malos momentos. Asimismo, quiero agradecer a Vicent por todo este tiempo de trabajo juntos, por hacer todo lo que estuviera en su mano para que siempre pudiera avanzar, y por animarme a adentrarme en la docencia universitaria. Gracias a ambos por todo lo que me habéis brindado.

Por otra parte, quiero expresar mi enorme agradecimiento al Profesor Gregorio Quintana-Ortí, de la Universitat Jaume I. Sus extensos conocimientos, entusiasmo, y gran dedicación han sido factores vitales para el desarrollo de gran parte de mi investigación. Su trabajo en el campo de computación Out-Of-Core ha sido esencial para el desarrollo de un método de reconstrucción eficiente. Gracias por tu disposición en todo momento, y por la gran calidad de tu trabajo, que sin duda es siempre un ejemplo a seguir.

Además, me gustaría agradecer a los Profesores Damián Segrelles, Ignacio Blanquer y Josep Arnal por su colaboración y ayuda, así como a Rafael Miró por su disposición para ofrecernos las plataformas de cómputo que se adapten a nuestras necesidades. Gracias también a Sandra Oliver y Vicent Giménez, espero que la colaboración que inicia pueda perdurar en el tiempo y aportar a nuestras investigaciones.

No puedo dejar de expresar mi gratitud hacia la Universitat Politècnica de València, así como hacia el Departamento de Sistemas Informáticos y Computación (DSIC), donde he llevado a cabo mi labor durante estos años, y el Instituto Universitario de Seguridad Industrial, Radiofísica y Medioambiental (ISIRYM), donde la llevo a cabo actualmente. Gracias a todo el personal que ha facilitado mi día a día.

Gracias a Sheila. No tengo palabras que puedan llegar a expresar lo que has hecho por mí. Gracias por hacer posible que hoy esté aquí.

Para terminar, quiero expresar mi más sincero agradecimiento a mi familia, mi apoyo incondicional, que siempre están ahí para empujarme hacia adelante. A mis padres, Santiago y Pilar, por los valores que me han inculcado y que me han hecho llegar a lo que soy hoy en día. En especial quiero agradecer a mi madre por ser el mayor ejemplo de fuerza y superación que podría tener en la vida. Tus ganas de vivir son más fuertes que cualquier cosa que venga. A mi hermana Raquel, por ser mi ejemplo en todos los ámbitos. Gracias por no dejarme caer nunca. A mis sobrinos, Gael y Valeria, que son la alegría de mis días. A mi cuñado Carlos, no puedo decir otra cosa que gracias por ser mi familia. No sé qué haría sin vosotros. Por último, a Gyset, gracias por ser mi roca, por hacerme sonreír en los peores momentos, y formar parte de todos los mejores. Eres mi mejor suerte.

Resumen

La prueba médica de Tomografía Computarizada (TC) es esencial actualmente en la práctica clínica para el diagnóstico y seguimiento de múltiples enfermedades y lesiones, siendo una de las pruebas de imagen médica más importante por la gran cantidad de información que es capaz de aportar. Sin embargo, a diferencia de otros métodos de diagnóstico por imagen que son inocuos, la prueba de TC utiliza rayos X, que son ionizantes, por lo que suponen un riesgo para los pacientes.

Es por ello que es necesario desarrollar métodos que permitan reducir la dosis de radiación a la que se expone a los pacientes que se realizan un estudio, sin comprometer la calidad de imagen puesto que sino se estaría sometiendo a un riesgo a estas personas sin que el beneficio (un diagnóstico de calidad) esté garantizado.

Durante el desarrollo de esta tesis se han investigado métodos de reconstrucción de imagen TC que se basan en reducir el número de proyecciones usadas, con el objetivo de reducir el tiempo de exposición a los rayos X. Esta estrategia de reducción de dosis está en fase de investigación, a diferencia de otras que están implantadas en la práctica clínica y ya han sido desarrolladas por los propios fabricantes de los escáneres.

Por tanto, nos hemos centrado en los llamados métodos algebraicos de reconstrucción, que son los más apropiados para este tipo de adquisición de proyecciones puesto que son capaces de trabajar con menos información que los métodos clásicos conservando una buena calidad de imagen. En concreto,

se ha estudiado a fondo el comportamiento del método LSQR para la resolución de este problema, combinado con una técnica de filtrado llamada *Soft Thresholding Filter* y una técnica de aceleración llamada FISTA. Además, se ha introducido un filtro de imagen denominado filtro Bilateral que es capaz de mejorar la calidad de las imágenes cuando se combina con los métodos anteriores.

El estudio multiparamétrico realizado se ha llevado a cabo en un entorno de computación distribuida Grid, para analizar cómo los distintos parámetros que intervienen en el proceso de reconstrucción pueden influir sobre la imagen resultado. Dicho estudio se ha diseñado para hacer uso de la potencia de cómputo de la plataforma distribuida aunque el software que se necesita no esté disponible. La instalación de dicho software se puede realizar en el tiempo de ejecución de los trabajos, o bien se puede empaquetar en una imagen que estará instalada en un contenedor Docker, lo que es una opción muy interesante para sistemas donde no tengamos privilegios. El esquema seguido para la creación y lanzamiento de los trabajos es fácilmente reproducible para estudios multiparamétricos de este tipo.

Por otra parte, se han planteado dos métodos algebraicos directos para la reconstrucción de TC basados en la factorización de la matriz que modela el sistema. El primero es el método SVD, que se ha probado mediante la librería SLEPc, obteniendo mayores tasas de uso de memoria principal, por lo que ha sido descartado en favor del método QR. La primera aproximación a la resolución se ha hecho mediante la librería SuiteSparseQR, desarrollando después un método propio siguiendo la técnica Out-Of-Core que permite almacenar las matrices en el propio disco duro en lugar de cargarlas en memoria, por lo que el tamaño del problema puede aumentar sin que el coste del hardware sea muy alto. Dicho método obtiene reconstrucciones de alta calidad cuando el rango de la matriz factorizada es completo. En los resultados se muestra como para una resolución alta, garantizar el rango completo todavía supone una reducción del número de proyecciones con respecto a métodos tradicionales.

Por tanto, en esta tesis se ha llevado a cabo la investigación y el posterior desarrollo mediante librerías y técnicas de computación de Altas Prestaciones de varios métodos algebraicos de reconstrucción de TC basados en la reducción de proyecciones que permiten mantener una buena calidad de imagen. Dichos métodos han sido optimizados para lograr los menores tiempos de reconstrucción posibles, con el fin de hacerlos competitivos y que algún día puedan ser instaurados en la práctica clínica.

Abstract

The Computerized Tomography (CT) medical test is currently essential in clinical practice for the diagnosis and monitoring of multiple diseases and injuries, being one of the most important medical imaging tests due to the large amount of information it is capable of providing. However, unlike other safe imaging methods, the CT test uses X-rays, which are ionizing, posing a risk to patients.

That is why it is necessary to develop methods that allow reducing the radiation dose to which patients undergoing a study are exposed, without compromising image quality since otherwise they would be subjecting these people to a risk without the benefit of a high-quality diagnosis being guaranteed.

During the development of this thesis, several CT image reconstruction methods that are based on reducing the number of projections used have been investigated, with the aim of reducing the time of exposure to X-rays. This dose reduction strategy is in research phase, unlike others that are implemented in clinical practice and have already been developed by the scanner manufacturers themselves.

Therefore, we have focused on the algebraic reconstruction methods, which are the most appropriate for this type of projection acquisition since they are capable of working with less information than the classical methods while maintaining good image quality. Specifically, the behavior of the LSQR method to solve this problem has been thoroughly studied, combined with a filtering technique called Soft Thresholding Filter and an acceleration technique called FISTA. In addition, the so-called Bilateral filter has been introduced, which

is capable of improving the quality of images when combined with the above methods.

The multiparametric LSQR study was carried out in a Grid distributed computing environment, to analyze how the different parameters involved in the reconstruction process can influence the resulting image. This study has been designed to make use of the computing power of the distributed platform even if the software required is not available. The installation of said software can be done at the time of execution of the jobs, or it can be packaged in an image that will be installed in a Docker container, which is a very interesting option for systems where we do not have privileges. The scheme followed for the creation and launch of the jobs is easily reproducible for multiparametric studies of this type.

On the other hand, two direct algebraic methods have been proposed for CT reconstruction based on the factorization of the matrix that models the system. The first is the SVD method, which has been tested using the SLEPc library, obtaining higher rates of main memory usage, which is why it has been discarded in favor of the QR method. The first approximation to the resolution has been made through the SuiteSparseQR library, later developing our own implementation using the Out-Of-Core technique that allows the matrices to be stored on the hard drive itself instead of loading them in memory, so the size of the problem can increase without the cost of the hardware being very high. This method obtains high-quality reconstructions when the rank of the factored matrix is complete. In the results it is shown that for a high resolution, guaranteeing the full rank still means a reduction in the number of projections compared to traditional methods.

Therefore, in this thesis, research and subsequent development of several algebraic CT reconstruction methods has been carried out using libraries and High Performance Computing techniques. These methods based on the reduction of projections, which allows maintaining good image quality, and have been optimized to achieve the shortest possible reconstruction times, in order to make them competitive so that one day they can be implemented in clinical practice.

Resum

Actualment, la prova mèdica de tomografia computeritzada (TC) és essencial en la pràctica clínica per al diagnòstic i el seguiment de múltiples malalties i lesions, sent una de les proves d'imatge mèdica més importants a causa de la gran quantitat d'informació que és capaç d'oferir. Tanmateix, a diferència d'altres mètodes d'imatge médica, la prova CT utilitza raigs X, que són ionitzants i suposen un risc per als pacients.

Per això, és necessari desenvolupar mètodes que permetin reduir la dosi de radiació a la qual estan exposats els pacients sotmesos a un estudi, sense comprometre la qualitat de la imatge, ja que en cas contrari estarien sotmetent a aquestes persones a un risc sense que es garantís l'avantatge d'un diagnòstic d'alta qualitat.

Durant el desenvolupament d'aquesta tesi, s'han investigat diversos mètodes de reconstrucció d'imatges CT basats en la reducció del nombre de projeccions utilitzades, amb l'objectiu de reduir el temps d'exposició als raigs X. Aquesta estratègia de reducció de dosis es troba en fase d'investigació, a diferència d'altres que s'implementen a la pràctica clínica i que ja han estat desenvolupades pels propis fabricants d'escàners.

Per tant, ens hem centrat en els anomenats mètodes de reconstrucció algebraica, que són els més adequats per a aquest tipus d'adquisició de projecció, ja que són capaços de treballar amb menys informació que els mètodes clàssics mantenint una bona qualitat d'imatge. Concretament, s'ha estudiat a fons el comportament del mètode LSQR per resoldre aquest problema, combinat

amb una tècnica de filtratge anomenada Soft Thresholding Filter i una tècnica d'acceleració anomenada FISTA. A més, s'ha introduït un filtre d'imatges anomenat filtre bilateral, que és capaç de millorar la qualitat de les imatges quan es combina amb els mètodes anteriors.

L'estudi multiparamètric de LSQR es va dur a terme en un entorn informàtic distribuït Grid, per analitzar com els diferents paràmetres implicats en el procés de reconstrucció poden influir en la imatge resultant. Aquest estudi ha estat dissenyat per fer ús de la potència de càlcul de la plataforma distribuïda encara que el programari requerit no estigui disponible. La instal·lació d'aquest programari es pot fer en el moment d'executar els treballs o es pot empaquetar en una imatge que s'instal·larà en un contenidor Docker, que és una opció molt interessant per a sistemes on no tenim privilegis. L'esquema seguit per a la creació i el llançament dels treballs es pot reproduir fàcilment per a estudis multiparamètrics d'aquest tipus.

D'altra banda, s'han proposat dos mètodes algebraics directes per a la reconstrucció CT basats en la factorització de la matriu que modela el sistema. El primer és el mètode SVD, que s'ha provat mitjançant la biblioteca SLEPc, obtenint taxes d'ús més alt de memòria principal, motiu pel qual s'ha descartat a favor del mètode QR. La primera aproximació a la resolució s'ha fet a través de la biblioteca SuiteSparseQR, desenvolupant posteriorment la nostra pròpia implementació mitjançant la tècnica Out-Of-Core que permet emmagatzemar les matrius al disc dur en lloc de carregar-les a la memòria, de manera que la mida de el problema pot augmentar sense que el cost del maquinari sigui molt alt. Aquest mètode obté reconstruccions d'alta qualitat quan el rang de la matriu factoritzada és complet. En els resultats es demostra que per a una alta resolució, garantir el rang complet encara significa una reducció del nombre de projeccions en comparació amb els mètodes tradicionals.

Per tant, en aquesta tesi s'ha dut a terme la investigació i el desenvolupament posterior de diversos mètodes de reconstrucció algebraica de CT mitjançant biblioteques i tècniques de computació d'altes prestacions. Aquests mètodes basats en la reducció de projeccions, que permeten mantenir una bona qualitat d'imatge, s'han optimitzat per aconseguir els temps de reconstrucció més breus possibles, per tal de fer-los competitius perquè algun dia puguin implementar-se a la pràctica clínica.

Índice general

Índice de figuras

Índice de tablas

Siglas y Abreviaciones

AIDR Adaptive Iterative Dose Reduction.

API Application Programming Interface.

ART Algebraic Reconstruction Technique.

ASIR Adaptive Statistical Iterative Reconstruction.

ATCM Automatic Tube Current Modulation.

BiCGStab Biconjugate Gradient Stabilized Method.

BLAS Basic Lineal Algebra Subprograms.

CGLS Gradient method for Least Squares.

EGi European Grid Infraestructure.

FBP Filtered Back-projection.

FLAME Formal Linear Algebra Method Environment.

GANs Generative Adversarial Networks.

GEMM General Matrix-Matrix Multiplication.

GMRES Generalized Minimal Residual.

GPU Graphic Processing Unit.

HDD Hard Drive Disk.

HPC High-Performance Computing.

HU Hounsfield Units.

IDE Integrated Development Environment.

IR Iterative Reconstruction.

IRIS Iterative Reconstruction in Image Space.

kVp Kilovoltios pico.

LAPACK Linear Algebra PACKage.

LSQR Least Squares QR.

mAs Miliamperios por segundo.

MBIR Model-Based Iterative Reconstruction.

MIMD Multiple Instructions Multiple Data.

MISD Multiple Instructions Single Data.

MKL Intel's Math Kernel Library.

MPI Message Passing Interface.

MRI Magnetic Resonance Imaging.

mSv Milisieverts.

NUMA Non-Uniform Memory Access.

OS-SART Ordered Subsets Simultaneous Algebraic Reconstruction Technique.

PET Tomografía por Emisión de Positrones.

PETSc Portable, Extensible Toolkit for Scientific Computation.

SAFIRE Sinogram Affirmed Iterative Reconstruction.

SART Simultaneous Algebraic Reconstruction Technique.

SE Storage Element.

SIMD Single Instruction Multiple Data.

SIRT Simultaneous Iterative Reconstruction Technique.

SISD Single Instruction Single Data.

SLEPc Scalable Library for Eigenvalue Problem Computations.

SPECT Tomografía Computarizada por Emisión de Fotones Simples.

SSD Solid State Drive.

SVD Singular Value Decomposition.

TC Tomografía Computarizada.

UMA Uniform Memory Access.

VO Virtual Organization.

Capítulo 1

Introducción

1.1 Motivación

Los avances tecnológicos desarrollados durante el último siglo han supuesto un antes y un después en la medicina. Es innegable la importancia de la introducción de la tecnología en la práctica clínica a la hora de obtener atención médica personalizada y de calidad.

Un claro ejemplo de ello es el desarrollo de nuevas pruebas de imagen para el diagnóstico más allá de las radiografías basadas en diferentes técnicas, como pueden ser: las ecografías, basadas en ultrasonidos, la prueba de Resonancia Magnética (MRI por sus siglas en inglés *Magnetic Resonance Imaging*), que hace uso de los campos magnéticos, la Tomografía Computarizada (TC), que utiliza rayos X, o bien pruebas de medicina nuclear que hacen uso de radiofármacos y gamma-cámaras como la Tomografía por Emisión de Positrones (PET) o Tomografía Computarizada por Emisión de Fotones Simples (SPECT).

Todas estas pruebas están orientadas a obtener imágenes que permitan estudiar las estructuras internas o procesos biológicos del paciente, con el fin de facilitar el proceso de diagnóstico, decisión de tratamiento, o seguimiento de una patología a los profesionales médicos. Sin embargo, pese a que todas ellas son

no invasivas, no todas ellas son inocuas para el paciente. En concreto, la TC, PET y SPECT emiten radiación ionizante que puede llegar a ser perjudicial.

No obstante, pese a la posibilidad de que la radiación ionizante tenga algún efecto nocivo sobre el paciente, sigue siendo necesario su uso hasta que se descubran nuevas técnicas que sean completamente equivalentes y totalmente inocuas. El claro ejemplo de esto es la prueba de TC. Pese a que el tipo de imagen y el objetivo de esta prueba es muy similar al de la resonancia magnética (estudiar las estructuras internas del paciente centrándose en la zona de interés), no son del todo equivalentes. La principal diferencia es que el TC funciona bien en zonas con elementos de diferentes densidades, como puede ser hueso con tejidos blandos. Sin embargo, la MRI es especialmente útil para distinguir en detalle entre tejidos blandos, por ejemplo en el cerebro, o para diagnosticar problemas de ligamentos. Además, puesto que la MRI es una prueba más larga y compleja, lo cual puede generar ansiedad o claustrofobia, y está contraindicada para personas con marcapasos o cualquier implante metálico, no es adecuada para ciertos paciente. Por tanto, en estos momentos se consideran pruebas complementarias que deben coexistir.

Con todo ello, debido a que el TC sigue siendo necesario y muy usado a pesar de no ser inocuo, es indispensable aplicar técnicas que sean capaces de conservar la calidad de imagen a la vez que disminuyen la dosis de radiación para que su posible nocividad se vea reducida. Actualmente hay diferentes estrategias que se pueden tomar para reducir la dosis, como puede ser tomar menos proyecciones en cada vuelta del escáner (por lo tanto menos disparos de rayos X), o bien tomar las proyecciones con menos intensidad y/o voltaje.

Dentro de los avances tecnológicos, las ciencias computacionales tienen un papel clave para lograr que los procesos sean lo más eficientes y precisos posible. En cualquiera de las pruebas nombradas es necesaria la obtención de la imagen digital a partir de los datos captados por los dispositivos. Es por ello que nuevas técnicas que hacen uso de la Computación de Altas Prestaciones (HPC) y Computación Paralela están en constante evolución y desarrollo, ya que el propósito principal es agilizar el proceso de obtención de las imágenes, tanto para acortar el tiempo que el paciente se encuentra realizando la prueba, como para que se reduzcan los tiempos de visita así como el de diagnóstico en casos urgentes.

Además, como se verá durante el desarrollo de esta tesis, los métodos utilizados para reducir la dosis en TC basados en la reducción de proyecciones son más complejos a nivel computacional que los métodos tradicionales al ser problemas

algebraicos, por lo que es todavía más importante optimizarlos para lograr reducir el tiempo necesario para obtener las imágenes.

1.2 Objetivos

Los métodos desarrollados en reconstrucción de imagen de TC deben ser competitivos a la vez que robustos para que se puedan llegar a utilizar en la práctica clínica. Los basados en la reducción de dosis tienen un potencial impacto muy positivo en la calidad de vida de los pacientes de alto riesgo, como son niños y mujeres embarazadas, ya que la radiación es especialmente nociva para ellos. Además, los pacientes con alguna enfermedad crónica también se podrán ver beneficiados, puesto que requieren realizar un seguimiento de la evolución de su patología mediante pruebas TC de manera regular, lo que conlleva que generalmente sobrepasen con creces la dosis máxima recomendada de radiación.

Por lo tanto, el objetivo general de este trabajo es el desarrollo de métodos de reconstrucción de imagen TC centrados en disminuir la dosis de rayos X inducida al paciente a través de la reducción del número de proyecciones tomadas, haciendo uso de computación paralela y técnicas HPC para lograr que las implementaciones sean eficientes y puedan competir con las técnicas de reconstrucción tradicionales.

Para lograr este objetivo ha sido necesario el cumplimiento de los siguientes objetivos específicos:

1. Estudio de los métodos de reconstrucción más extendidos así como de nuevos enfoques para reducción de dosis.
2. Estudio y análisis del efecto de la reducción de vistas tanto sobre el sinograma, como en la matriz del sistema. Determinar cómo influye en el proceso de reconstrucción mediante el método algebraico iterativo LSQR.
3. Análisis de la viabilidad de otros métodos de reconstrucción algebraicos mediante el uso de librerías.
4. Desarrollo de filtros que permitan prevenir artefactos en la imagen y reducir ruido.
5. Estudio y análisis sobre la simulación de los datos para tener una base de trabajo.

6. Análisis de todos los parámetros que influyen en el proceso de reconstrucción de la imagen para lograr su parametrización.
7. Implementación de los métodos de reconstrucción escogidos, haciendo uso de programación *multicore*.
8. Análisis de prestaciones de los métodos desarrollados.
9. Estudio de la calidad de imagen obtenida y las limitaciones de cada uno de los métodos.

1.3 Estado del Arte

1.3.1 Reducción de dosis

La creciente utilización del TC desde su invención en los años 70 ha originado un gran interés en cuanto a la investigación de nuevos métodos y técnicas de reconstrucción. Pese a estar tan instaurado, es un campo en constante evolución y desarrollo, en el cual se combinan avances de hardware que permiten que los escáneres sean más rápidos, exactos y utilicen menos radiación, como los avances en software y métodos matemáticos que permitan optimizar los procesos de reconstrucción, maximizando la calidad de imagen y reduciendo la cantidad de dosis a la que se tienen que exponer los pacientes.

Hoy en día, aproximadamente el 50% de la población en Estados Unidos que está expuesta radiación es debido a pruebas de imagen médica. Dentro de estas pruebas, el TC supone el 63% de la dosis total inducida mediante técnicas nucleares según el *National Council on Radiation Protection and Measurements* (Mettler y col. 2019), siendo la principal fuente de exposición no natural a radiación ionizante, lo cual la convierte en el foco principal en cuanto a protección radiológica se refiere.

Varios estudios han demostrado el aumento generalizado de la incidencia de cáncer en todo el mundo, y también la proyección de muertes provocadas por cáncer para 2020 y 2030 (Rahib y col. 2014). Por ejemplo, el riesgo de tener cualquier tipo de cáncer es de alrededor de un 30% hoy en día en la Unión Europea, según la Agencia Internacional para la Investigación del Cáncer ("IARC"). En todo el mundo, se estima que habrán 28,4 millones nuevos casos de cáncer en 2040, un aumento del 47% de los correspondientes 19,3 millones de casos en 2020 según las proyecciones (Sung y col. 2021). El cáncer está entre la primera y la tercera causa de muerte en personas menores de 70 años según el país.

Los riesgos por la exposición continuada de un paciente a rayos X siguen siendo difíciles de probar. Esto es debido a que estos estudios en adultos son limitados, ya que los efectos se observan a largo plazo, dado que la latencia de un cáncer provocado por exposición a radiación ionizante es de 10 a 20 años, por lo que muchas personas mayores no llegarán a mostrar síntomas. El estudio más importante para estimar el riesgo de cáncer por exposición a la radiación es la investigación de seguimiento continua a largo plazo de los sobrevivientes japoneses de las bombas atómicas lanzadas sobre Hiroshima y Nagasaki en 1945. Este estudio se conoce como *"Life Span Study"* o LSS.

De los resultados periódicos de este estudio, se concluye que existe una relación lineal entre la dosis de radiación a la que una persona ha sido expuesta y el riesgo de padecer algún tipo de cáncer (estómago, colon, hígado, pulmones, próstata, mama, entre muchos otros) incluso hasta 60 años después de la exposición (Grant y col. 2017). Por tanto, según esta relación, no existe un nivel seguro de exposición.

Sin embargo, hay estudios como (Schultz y col. 2020) que establecen que utilizando metodología de cohortes se puede determinar que sí existe un umbral de 100 milisieverts (mSv), por debajo del cual no se produce un aumento del riesgo de cáncer provocado por radiación ionizante. Si esto es así, este umbral equivaldría a poder realizar una media de 10 estudios de TC de abdomen para un adulto sin aumentar el riesgo.

Aunque este parece un límite razonable, un análisis de datos obtenidos de 324 hospitales (Rehani y col. 2020) demuestra que aproximadamente el 1,3% de pacientes que requieren TCs recurrentes durante un periodo de entre 1 y 5 años para estudiar la evolución de sus enfermedades supera la dosis umbral establecida. De hecho, según este análisis, el tiempo mínimo para acumular 100 mSv fue un solo día, lo que supone una problemática evidente.

En cuanto a pacientes pediátricos, hay muchos más estudios que evidencian los riesgos de la exposición a radiación ionizante. Esto se debe a que los niños tienen más años por delante para que se vean los efectos malignos de la radiación, y también a que ésta tiene mayor efecto en células más proliferativas (Ichimaru, Ishimaru y Belsky 1978). Se ha probado que la exposición a radiación ionizante producida por el TC de estos pacientes aumenta el riesgo de padecer leucemia o cáncer cerebral principalmente, y otros tipos de cáncer como mama, tiroides o faringe (Meulepas y col. 2019; Hong y col. 2019; Krille y col. 2015; Pearce y col. 2012).

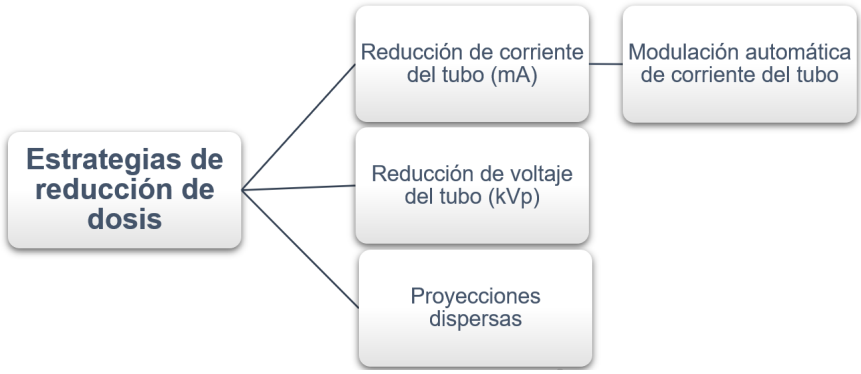


Figura 1.1: Clasificación de las distintas estrategias de reducción de dosis TC.

Por todo lo expuesto anteriormente, es clara la necesidad de centrar los esfuerzos en reducir la dosis empleada por los escáneres de TC. Para conseguirlo existen diferentes estrategias que se pueden adoptar. En la Figura 1.1 se puede apreciar una clasificación de estas estrategias, que se detallarán a continuación.

La técnica más usada y la más sencilla de aplicar es la reducción de la corriente o intensidad de la fuente empleada para realizar la adquisición. Esta intensidad de mide en miliamperios, aunque puede ser expresada multiplicando por el tiempo de exposición en miliamperios-segundo (mAs). El flujo de fotones generado por el tubo de rayos X es proporcional a la corriente del tubo, por lo que una la dosis se reducirá en el mismo porcentaje que se reduzca la corriente. Sin embargo, la reducción es limitada. Por una parte, el número de fotones que penetran en el paciente tiene que ser lo suficientemente alto como para tener la información necesaria en la imagen reconstruida.

Por otra parte, el ruido que aparece en las reconstrucciones al reducir la intensidad de la fuente se debe considerar. El ruido dominante es el ruido cuántico, que obedece a una distribución de Poisson. El ruido cuántico es proporcional a $\sqrt{N}$ y el ruido de imagen correspondiente es aproximadamente proporcional a $1/\sqrt{N}$, donde N es el número de fotones que han contribuido a la imagen reconstruida. El valor de N depende en gran medida del potencial del tubo, y además es proporcional al ancho de la sección, la corriente del tubo y la cantidad de tiempo necesaria para adquirir todos los datos de proyección necesarios para la reconstrucción. En el modo secuencial, este tiempo es igual al tiempo que la fuente de rayos X está activa por rotación, por lo que el ruido de la imagen es aproximadamente proporcional a $1/\sqrt{mAs}$. Se estima que con una reducción al 50%, el ruido en la imagen se incrementa en un 41% (Maldjian

y Goldman 2013). Esto puede provocar mala calidad de imagen, y principalmente una pérdida de la calidad del diagnóstico, lo que supondría exponer en vano al paciente a radiación ionizante. En la Figura 1.2 se puede apreciar el efecto de la reducción de intensidad, observando claramente el aumento de ruido en las imágenes con una intensidad de 50 mAs.

Dependiendo de la zona a examinar y diagnosticar, la reducción de la intensidad podrá ser mayor o menor para conservar la capacidad de diagnóstico aunque el ruido en la imagen se vea incrementado. Además, hay que tener en cuenta las características del paciente, puesto que pueden influir en las imágenes. Por ejemplo, en un escáner abdominal a un paciente obeso el nivel de ruido permitido puede ser más alto ya que el tejido adiposo proporciona un mejor contraste natural entre los órganos (Qurashi y col. 2018).

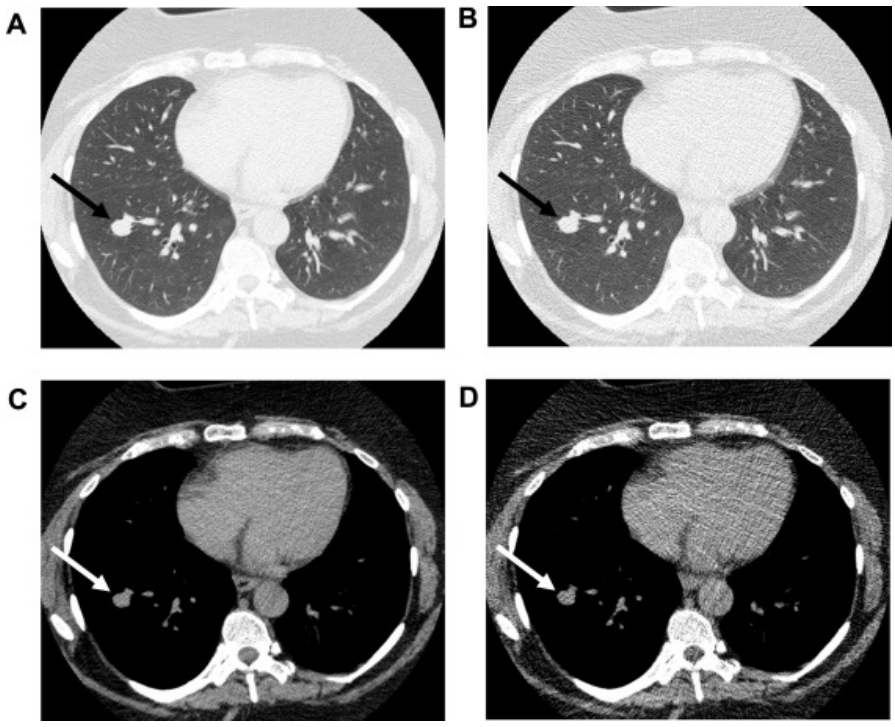


Figura 1.2: Comparativa imagen obtenida a 100 mAs (A y C) e imagen obtenida a 50 mAs (B y D) (Kubo y col. 2016).

En trabajos como (Lee y col. 2017; Kubo y col. 2016; Hetmaniak y col. 2003; Tack y col. 2005) se muestra que es factible reducir la intensidad de la fuente y por tanto la dosis de radiación conservando la calidad de diagnóstico en campos como detección de nódulos pulmonares, embolias pulmonares, detección de tumores, entre otros.

Basado en la técnica anterior surge una mejora: la modulación automática de la corriente del tubo (ATCM), también conocida como control automático de exposición. Esta técnica se basa en la idea de que de la misma manera que la atenuación de los fotones varía entre las diferentes regiones del cuerpo también lo hace el ruido de píxeles de la imagen reconstruida si el resto de parámetros son fijos. Por tanto, el objetivo es variar la corriente del tubo en diferentes proyecciones y regiones del cuerpo para lograr niveles de ruido de imagen más uniformes, manteniendo el mismo nivel de calidad de imagen durante un barrido.

La modulación de la corriente se puede aplicar tanto longitudinalmente (eje z) como de manera angular (en el plano xy). Cuando se aplica de manera longitudinal la corriente debe ir variando a medida que el paciente se desplaza, y se utiliza una radiografía para determinar la atenuación de fotones a lo largo del eje z del paciente. Ciertas partes del cuerpo humano atenúan más fotones que otras. Por ejemplo, el tórax contiene principalmente aire y, por lo tanto, atenúa poco los fotones. Por el contrario, la pelvis contiene huesos y tejidos blandos y es un fuerte atenuador de fotones. Utilizando los valores de atenuación obtenidos del explorador, se ajusta la corriente del tubo durante la adquisición para lograr un ruido de píxeles constante en todas las regiones del cuerpo exploradas.

Esto se puede combinar con la modulación angular, que consiste en variar la corriente del tubo mientras éste rota alrededor del paciente para conseguir una calidad de imagen uniforme. Esto se puede llevar a cabo debido a que el cuerpo humano es altamente asimétrico. Por ejemplo, en un escáner de torso, los hombros van a generar una mayor atenuación cuando se tomen las proyecciones laterales que cuando se tomen las frontales. Por tanto, se necesitará más intensidad en las laterales para mantener el nivel de ruido constante con respecto a las demás posiciones.

En (Leswick y col. 2008) se realiza un estudio de tiroides con un fantoma físico en el que se demuestra como el uso de ATCM longitudinal puede reducir hasta en un 60% la dosis de radiación manteniendo el ruido en niveles muy bajos. En un estudio similar realizado con fantomas pediátricos (Papadakis y Damilakis 2019) se concluye que se puede lograr una reducción de entre un 8% a un 24%

de la dosis en la cabeza, entre un 16% y un 39% en el tórax y entre 25% y 41% en el abdomen/pelvis. Esto ha sido verificado en estudios con pacientes reales, como el estudio retrospectivo realizado en (Yurt, Özsoykal y Obuz 2019) que concluyó una reducción de entre un 21% y un 31% usando este protocolo en TC abdominal, según la fase arterial.

La siguiente técnica más usada es la reducción del voltaje de la fuente. Sin embargo, este técnica tiene más limitaciones. La principal es que no es tan configurable como la anterior, puesto que los escáneres tienen predeterminados unas configuraciones de voltaje a escoger, típicamente 80, 100, 120, o 140 kVp. En estudios como el que se presenta en (Khan y col. 2013) se demuestra cómo reduciendo el voltaje de 120 a 100 kVp se puede lograr un 30% de reducción de dosis de radiación.

Sin embargo, no es una técnica libre de problemas. En el estudio citado, la reducción del voltaje incrementó el ruido de la imagen en un 14%, lo que puede llegar a dificultar el diagnóstico. Además, para compensar el incremento de ruido en la imagen al bajar la energía, se suele incrementar la intensidad de la fuente, lo que a su vez incrementa de nuevo la dosis de radiación, perdiendo así gran parte del beneficio en cuanto a dosis se refiere.

En pacientes obesos se genera un problema adicional, ya que la grasa absorbe más los fotones de baja energía, perdiéndose así información valiosa para la imagen que tiene que ser reconstruida. Por ello, esta técnica es más adecuada para pacientes con un bajo índice de grasa corporal o pacientes pediátricos.

A pesar de los problemas anteriores, sigue siendo una técnica útil, sobretodo en casos donde se pueda o tenga que usar contraste que contenga yodo para realizar la prueba, debido a su borde de absorción de capa K. El borde de absorción de capa K describe un aumento repentino en el coeficiente de atenuación de los fotones que se produce a una energía de fotón justo por encima de la energía de enlace del electrón de la capa K de los átomos que interactúan con los fotones.

Los elementos más abundantes en el tejido humano (hidrógeno, carbono, oxígeno, nitrógeno) tienen bordes de absorción K que son demasiado bajos para ser detectables. Pero elementos como el yodo lo tiene a 33,2 kVp, lo cual se sitúa cerca de la energía media de cualquier escáner. En particular, el yodo tiene su máxima absorción con una energía de 80 kVp.

En estudios de angiografía como (Cho y col. 2012; Szucs-Farkas y col. 2008) se muestra como utilizando un contraste de yodo se puede reducir la dosis

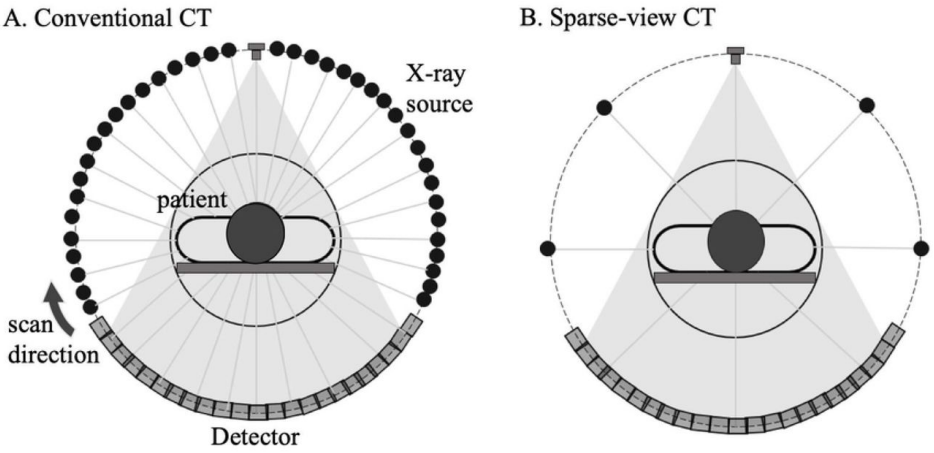


Figura 1.3: Escáner tradicional continuo vs Escáner disperso (Dong y col. 2019).

en grandes porcentajes logrando mejor realce de las arterias en las imágenes empleando 80 kVp que cuando se emplean 100 o 120 kVp, por lo que no sólo se reduce la radiación, sino que se mejora la imagen.

Los métodos de reducción de dosis expuestos son bien conocidos y están establecidos en la práctica clínica. Los fabricantes de escáneres TC los han estudiado e implementado de manera óptima en sus máquinas, por lo que tienen poco margen de mejora y nadie mejor que los fabricantes para optimizarlos puesto que conocen al detalle el hardware y todos los procesos que se realizan en la adquisición.

En los últimos años se ha propuesto un nuevo método de adquisición para reducción de dosis: el *"Sparse Sampling"* o adquisición de proyecciones dispersas. También es referido en ocasiones como TC de pocas vistas. Este método, a diferencia de todos los anteriores se basa en reducir el tiempo de exposición en lugar de reducir la intensidad o el voltaje de la fuente, lo cual genera ruido en la imagen debido a que los fotones no penetran de igual manera en el paciente y se pierde información. Sin embargo, con la adquisición dispersa, lo que se pretende es que la fuente de rayos X solo esté activa en el momento que se toma la proyección, y que las proyecciones no sean continuas, sino que se reduzcan en número y realicen adquisiciones en ciertos ángulos de rotación seleccionados. El funcionamiento de la adquisición se ilustra en la Figura 1.3.

De esta manera, el ruido de la imagen no aumentaría ya que se mantiene la misma intensidad y voltaje. En cambio, puesto que el tiempo de exposición es

más bajo, el paciente absorberá menos dosis de radiación. Aún así, el reducir el número de adquisiciones produce otro tipo de artefactos, mayoritariamente artefactos tipo *"Streak"* o de rayas, que habrá que tratar. Son muchos los trabajos que han explorado la viabilidad de utilizar este tipo de muestreo durante el escaneado TC, y actualmente se presenta como una alternativa muy factible a los métodos de reducción de dosis más tradicionales.

A pesar de ello, todavía no es una técnica que se haya aplicado de forma real, ya que no existe ningún escáner físico que tenga las características requeridas para realizar este tipo de muestreo. Esto se debe a que en los escáneres clínicos existentes no es posible generar los pulsos de rayos X a la velocidad requerida para la adquisición. Por tanto, esta aproximación requiere del diseño y fabricación de un nuevo tipo de escáner, lo cual ningún gran fabricante ha llevado a cabo hasta el momento.

En la actualidad, hay un prototipo de escáner basado en adquisición dispersa, que es el definido en (Muckley y col. 2019). En este prototipo de escáner con múltiples filas de detectores lo que se plantea no es encender y apagar la fuente, lo cual es complicado para la velocidad requerida, sino bloquear los rayos para ciertas filas de detectores, reduciendo así el número de proyecciones y por tanto la radiación inducida al paciente. Los rayos se bloquean con una placa de tungsteno con ranuras espaciadas periódicamente, de modo que los datos de proyección están submuestreados. Este prototipo, aunque no se adapta completamente al planteamiento de reducción de proyecciones como tal, podría ser el punto de inicio para introducir la adquisición dispersa en la práctica clínica, y de esta forma, una nueva estrategia de reducción de dosis.

Siguiendo esta estrategia, se podrían lograr desarrollar escáneres de muy baja dosis. En un estudio preliminar sobre detección de embolias pulmonares (Sauter y col. 2019) se ha conseguido una reducción del 87.6% de la dosis manteniendo una calidad alta de imagen y una validez para el diagnóstico con sensibilidad de casi un 100%, evaluado por radiólogos.

En otro ámbito como es la detección de lesiones vertebrales, también se ha estudiado la viabilidad de este tipo de adquisición (Sollmann y col. 2019), concluyendo que es posible lograr una reducción de dosis de hasta un 50% sin que se pierda capacidad de detectar lesiones, y de hasta un 75% para determinar el tiempo que tiene una lesión. Comparado con la reducción de intensidad de la fuente se demuestra un mejor rendimiento con proyecciones dispersas.

La estrategia de adquisición dispersa o con pocas vistas ha sido la adoptada para el desarrollo de esta tesis, puesto que al estar en fase de investigación y no establecida en la práctica clínica tiene más margen de mejora, además del incentivo extra que es la gran reducción de dosis que se lograría al utilizarse en casos reales.

1.3.2 Métodos de reconstrucción

Todo lo expuesto en el punto anterior es la base del proceso de obtención de una imagen de TC. El primer paso es la adquisición de los datos del paciente, lo cual se realiza mediante un escáner validado para su uso clínico, y es llevado a cabo por profesionales médicos. Sin embargo, es a partir de esos datos cuando las ciencias computacionales ganan protagonismo.

Una vez se han obtenido las proyecciones es necesario transformarlas en imágenes anatómicas para poder diagnosticar enfermedades o detectar anomalías. Para ello, es necesario aplicar métodos de reconstrucción para pasar los datos al dominio de la imagen.

En este campo se podrían clasificar los métodos de reconstrucción en tres grupos: los analíticos, los algebraicos y los basados en técnicas de *Deep Learning*. En la Figura 1.4 se puede visualizar la clasificación y algún ejemplo concreto de cada tipo de método, que se explicarán a continuación.

Pese a que los algebraicos se plantearon desde la misma invención del escáner TC, fue difícil ponerlos en práctica por la baja potencia computacional existente en aquellos años, la cual era requerida por este tipo de métodos que se basan en resolver un sistema de ecuaciones lineales. Por ello, los métodos analíticos basados en la transformada de Radon tomaron todo el protagonismo ya que tienen un bajo coste computacional y por tanto son muy rápidos a la hora de obtener las imágenes.

El método base para todos los analíticos es el método de retroproyección filtrada (FBP por sus siglas en inglés). A su vez, se basa en el método de retroproyección simple, el cual emplea la transformada inversa de Radon para pasar al dominio de la imagen, al que se le añade un proceso de filtrado. El filtrado se añade para eliminar las componentes de baja frecuencia que emborronan la imagen. Este proceso será detallado en capítulos posteriores de manera más teórica.

El FBP sigue siendo el más usado en estudios generales, sin reducción de dosis de ningún tipo y para pacientes no obesos. Cuando se da alguna de esas

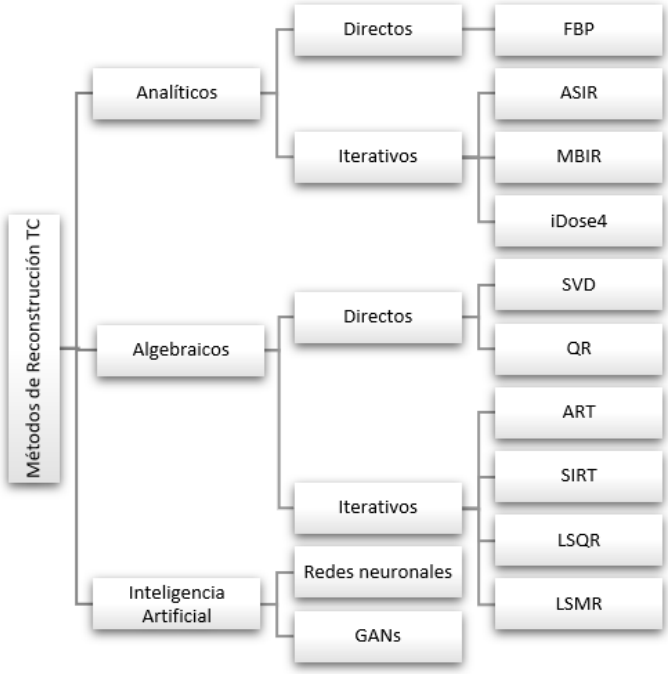


Figura 1.4: Clasificación de los distintos tipos de métodos de reconstrucción TC.

circunstancias, la calidad de las imágenes empeora, aumentando el ruido y los artefactos. Por tanto, deja de ser un método adecuado ya que la validez de las imágenes puede estar comprometida.

Es por ello que dentro de esta categoría surgen posteriormente los métodos analíticos-iterativos, referidos en la literatura como IR. Los métodos iterativos analíticos buscan paliar los efectos negativos de la reducción de dosis (ya sea por reducción de intensidad o voltaje), obteniendo imágenes de calidad similar al FBP. Se basan en ciclos iterativos en los que la imagen se va corrigiendo, eliminando ruido mediante procesamiento no lineal a partir de información estadística hasta que se obtiene la calidad óptima con un nivel de ruido mínimo. Estas correcciones se pueden hacer basándose en el sinograma, basándose solo en la imagen o en ambos. En la Figura 1.5 se puede apreciar el funcionamiento de este tipo de algoritmo.

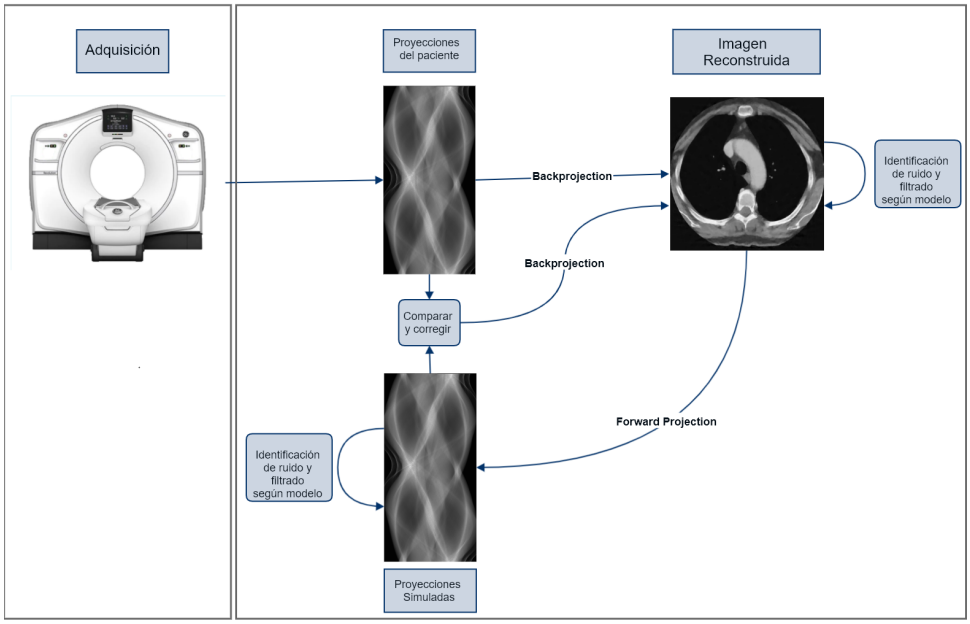


Figura 1.5: Esquema de funcionamiento de los algoritmos de reconstrucción iterativos (IR).

Sin embargo, esta estrategia también aumenta el coste temporal, puesto que iterar y corregir la imagen aumenta la complejidad. El primer método de este tipo usado a nivel clínico fue el *Adaptive Statistical Iterative Reconstruction (ASIR)*, de *General Electrics Healthcare*. Este algoritmo iterativo utiliza la reconstrucción obtenida mediante FBP como imagen inicial, que se vuelve a proyectar con el método *Forward Projection* para obtener un sinograma simulado, el cual se compara con un sinograma ideal obtenido a partir de un modelo de ruido creado teniendo en cuenta estadísticas de los fotones y el ruido electrónico que afectan al ruido de la imagen. Se comparan ambos sinogramas midiendo la diferencia, y se vuelve a realizar el paso de retroproyección para obtener la imagen diferencia con la que se corrige la reconstrucción. Este proceso se repite hasta que la imagen converja. Además, la imagen final es combinada con la reconstrucción obtenida con FBP en el porcentaje que se elija.

Una evolución del método anterior es el llamado *Model-Based Iterative Reconstruction (MBIR)* (conocido como VEO comercialmente), del mismo fabricante. En este caso, la imagen no se combina con la obtenida por el FBP. La mejora que aporta esta nueva técnica es que el modelo estadístico es mucho más

avanzado, ya que tiene en cuenta el haz de rayos X en el punto focal, la forma del haz cuando sale del ánodo, la interacción del haz cuando pasa a través del paciente y la interacción entre el haz y el detector de rayos X en el otro lado del el paciente. Esta técnica es puramente iterativa y más compleja computacionalmente.

Existen otros métodos comerciales como iDose^4 de *Philips Healthcare*, *Sinogram Affirmed Iterative Reconstruction (SAFIRE)* de *Siemens Medical Solutions* o *Adaptive Iterative Dose Reduction (AIDR)* 3D de *Toshiba Medical Systems*, que iteran y corrigen tanto el sinograma como la imagen en cada pasada, y otros como *Iterative Reconstruction in Image Space (IRIS)* de *Siemens Medical Solutions* que únicamente identifican y corrigen el ruido sobre la imagen reconstruida.

Todos estos métodos están operativos en escáneres clínicos actualmente, y se ha mostrado su efectividad a la hora de trabajar con proyecciones adquiridas con estrategias de reducción de dosis que no se basen en la reducción de vistas. Al eliminar el ruido de las imágenes de forma más efectiva, es posible trabajar con proyecciones adquiridas con muy baja dosis. Los métodos que logran más calidad son iDose^4 y MBIR, con reducciones de dosis de hasta un 80% y 75% respectivamente según los fabricantes, seguidos de IRIS con un 60% (May y col. 2011), AIDR 3D con una reducción del 52% (Gervaise y col. 2012), SAFIRE del 50% (Moscariello y col. 2011), y ASIR, que obtiene entre un 23% y un 76% de reducción de dosis dependiendo de en qué porcentaje se combine con FBP (Sagara y col. 2010; Yanagawa y col. 2012)).

En cuanto a eficiencia, se han conseguido optimizar para que puedan competir con el método FBP y funcionar en tiempo real, a excepción del método MBIR. Como ejemplo, en (Korn y col. 2012) concluyen que el método IRIS reconstruye un estudio completo en una media de 68 segundos contra 25 segundos del FBP. Con SAFIRE, el número de cortes por segundo es de 20 mientras que con FPB es 40 según (Moscariello y col. 2011). En otro estudio se consigue reconstruir 16 cortes por segundo con iDose^4 comparadas con 22 con FPB (Funama y col. 2011). Mediante ASIR, se pueden lograr velocidades un 40% menores que con FPB.

Por tanto, todos los tiempos son lo suficientemente competitivos como para emplearse en tiempo real (obteniendo las reconstrucciones in-situ en el momento que se realiza el estudio). Por contra, el método MBIR, que obtiene mayor calidad de imagen, solo puede ser empleado de forma *offline*, generalmente en máquinas externas con mayor capacidad computacional que la que tienen las integradas en los escáneres. Esto es debido a que los tiempos de

reconstrucción son mucho más elevados. Por ejemplo, según (Notohamiprodjo y col. 2015), un estudio de TC craneal se reconstruye en una media de 48 segundos mediante ASIR, y en 1920 segundos (32 minutos) mediante MBIR, lo que supone un tiempo casi 40 veces mayor.

Todos los métodos iterativos descritos están formulados para trabajar con proyecciones completas adquiridas a baja dosis. Sin embargo, cuando se trabaja con proyecciones dispersas, dejan de ser óptimos, puesto que la información que se pierde al reducir los puntos de proyección genera artefactos utilizando métodos de retroproyección.

En este supuesto es cuando es más conveniente aplicar los métodos de reconstrucción algebraicos. Este tipo de reconstrucciones se plantearon con la propia invención del TC, siendo *Algebraic Reconstruction Technique (ART)* la primera técnica de reconstrucción propuesta. Esta estrategia se basa en modelar el problema de reconstrucción como un sistema de ecuaciones lineales, en el que interviene una matriz del sistema o matriz de pesos que se modela según los parámetros físicos del escáner (número de detectores, distancias, ángulo de apertura, etc) y el sinograma o proyecciones tomadas en la adquisición, y se obtiene la imagen resultado correspondiente a la reconstrucción. El sistema tiene tantas incógnitas como píxeles tenga la imagen.

El problema que presenta esta aproximación es la gran dimensionalidad que puede tomar el sistema de ecuaciones, lo que no era abordable en la época, tanto por disponibilidad de memoria para cargar los datos, como en tiempo de cómputo por la falta de potencia computacional. De este modo, la reconstrucción fue orientada hacia métodos analíticos, más simples y rápidos.

Con la evolución del hardware durante las últimas décadas la resolución de este tipo de problemas que antes era de un coste inasumible empezó a ser relativamente asequible, por lo que los métodos algebraicos volvieron a tomar protagonismo. Para resoluciones de imagen no muy altas, este sistema de ecuaciones incluso se puede resolver de manera directa mediante la factorización de la matriz del sistema, siempre que la reducción de proyecciones no sea tanta como para que la matriz del sistema sea de rango deficiente.

En cuanto a estos métodos directos hay poca literatura, ya que aunque se ha planteado su uso con anterioridad, consumen más recursos por lo que son más limitados. En (Gullberg, Hsieh y Zeng 1996) proponen el uso de la descomposición en valores singulares SVD destacando el extenso tiempo requerido para realizar las operaciones, mientras que en (Rodríguez-Alvarez y col. 2018) utilizan la QR para factorizar la matriz del sistema y después obtener la imagen

reconstruida a partir de datos reales, concluyendo que la calidad es comparable a los métodos analíticos pero planteando el problema del tiempo de resolución y la dimensionalidad. Ambos métodos han sido probados durante el desarrollo de esta tesis, y se ha desarrollado un método basado en la QR altamente eficiente y escalable que no requiere de gran cantidad de memoria principal.

Por otra parte, también puede aproximarse la solución al sistema de ecuaciones mediante algoritmos iterativos. Entre estos, se encuentra el ART (Gordon, Bender y Herman 1970), y sus evoluciones *Simultaneous Iterative Reconstruction Technique (SIRT)* (Gregor y Benson 2008) y *Simultaneous Algebraic Reconstruction Technique (SART)* (Jiang y Wang 2003). Estos métodos calculan iterativamente una aproximación a la imagen solución, actualizando en cada iteración proyectando la solución aproximada con la matriz del sistema y comparando con el sinograma original. De esta manera se calculan las diferencias y las correcciones a aplicar en la solución en la siguiente iteración.

Si el sistema es de rango completo y las proyecciones fueran ideales (sin ningún tipo de ruido), estas técnicas encontrarían la solución exacta. Sin embargo, puesto que hay ruido involucrado y además se suelen utilizar para proyecciones en las que se han reducido las vistas, el sistema está infradeterminado, por lo que la solución es una aproximación.

La diferencia de ART con SIRT y SART es que el primero actualiza el vector solución utilizando una sola fila de la matriz del sistema en cada iteración, por lo que la actualización es secuencial. Sin embargo los otros dos actualizan de forma simultánea utilizando la información de todas las filas sin tener que iterar sobre todas las filas de la matriz secuencialmente. Existen más optimizaciones de estos métodos, como *Ordered Subsets Simultaneous Algebraic Reconstruction Technique (OS-SART)* (Wang y Jiang 2004), que realiza la actualización simultánea a bloques, lo que permite acelerar el tiempo de reconstrucción.

Por otro lado están los métodos iterativos basados en los subespacios de Krylov, que se pueden usar para resolver el sistema de ecuaciones con una convergencia más rápida que los anteriores, que además están indicados especialmente cuando la matriz del sistema es de grandes dimensiones y dispersa, lo cual se adapta perfectamente a este tipo de problema.

Son diversos los métodos de Krylov aplicados al campo de reconstrucción de imagen de TC, como puede ser Gradient method for Least Squares (CGLS) (Marquez y col. 2020), Biconjugate Gradient Stabilized Method (BiCGStab) (Cools y col. 2015), Generalized Minimal Residual (GMRES) (Coban y Lionheart 2014) o Least Squares QR (LSQR) (Parcero y col. 2017), todos ellos con buen

rendimiento en cuanto a calidad de imagen reconstruida, además de tener una mejor tasa de convergencia que los iterativos basados en ART.

Por otra parte, estos métodos suelen ser combinados con técnicas de regularización, puesto que el sistema no es perfecto, ya que las proyecciones suelen contener ruido y además suele estar infradeterminado. Las técnicas de regularización más usadas son la de Tikhonov (Golub, Hansen y O'Leary 1999; Wysoczański, Mroczka y Polak 2013), y la minimización de la Variación Total (Yu y Zeng 2014).

En los últimos años, los esfuerzos en la investigación en este campo se están dirigiendo hacia el uso de inteligencia artificial mediante redes neuronales profundas (*Deep Learning*). Con ello, se pueden solventar los problemas que aparecen en las reconstrucciones al reducir la dosis: eliminando el ruido generado al reducir voltaje/intensidad de la fuente (Wolterink y col. 2017; Yang y col. 2018), y eliminando los artefactos de tipo *Streak* o raya generados por la reducción de proyecciones (Han y Ye 2018; Xie y col. 2018).

La aplicación de las redes neuronales se puede hacer tanto a nivel de imagen (filtrado), como a nivel de sinograma (filtrado e interpolado para rellenar los datos que faltan), como una combinación de las dos con Redes Generativas Antagónicas (RGAs o GANs). Además, se ha demostrado que es posible desarrollar métodos de reconstrucción mediante una red neuronal entrenada para pasar de un sinograma a una imagen directamente sin tener que aplicar ningún otro método de reconstrucción (Li y col. 2019; Ge y col. 2020), y también que es posible extraer característica de las imágenes sin necesidad de reconstruir, directamente desde los sinogramas (De Man y col. 2019).

El principal inconveniente a la hora de investigar en este campo es la escasez de datos. Es sabido que para aplicar técnicas de inteligencia artificial se necesita una gran cantidad de datos que conformen una muestra lo suficientemente grande para poder realizar un entrenamiento sobre ella. Pese a que es común encontrar bancos de datos de imágenes de TC, no ocurre lo mismo con los sinogramas. Por lo tanto, sigue siendo complicado desarrollar métodos de calidad si no es de la mano de los propios fabricantes.

Como se ha expuesto con anterioridad, la eficiencia de los métodos de reconstrucción es un aspecto muy importante de cara a poder usarlos clínicamente. Puesto que los métodos iterativos, ya sean analíticos-estadísticos o algebraicos conllevan un gran coste computacional es vital optimizarlos para lograr un alto rendimiento y que de esta manera los tiempos de reconstrucción se reduzcan hasta poder ser comparables a los métodos analíticos utilizados clásicamente.

Existen multitud de implementaciones paralelas de métodos de reconstrucción TC, destacando el *toolbox* ASTRA (Aarle y col. 2016), que contiene implementaciones tanto CPU como GPU de diversos métodos (algebraicos e iterativos), orientado a investigadores no comerciales. Dicho *toolbox* obtiene tiempos de reconstrucción muy competitivos en GPU para métodos algebraicos, por ejemplo reconstruyendo un volumen de 400^3 voxels mediante 100 iteraciones de SIRT en alrededor de 40 segundos, mientras que la aproximación implementada en CPU mediante OpenMP obtendría un tiempo similar para reconstrucciones en 2D de 400^2 píxeles.

En otra implementación paralela con OpenMP por bloques de métodos iterativos basados en ART y SIRT presentada en (Sørensen y Hansen 2014) se consiguen *SpeedUps* de un factor entre 5 y 10 en un sistema de 20 *cores*, logrando tiempos por iteración de poco más de 3 segundos para un problema de 256^3 voxels.

Otro ejemplo es la implementación cuART que proponen en (Yu y col. 2019), donde la paralelización de métodos basados en ART mediante GPU y la librería CUDA puede obtener *SpeedUps* de hasta 25 para imágenes de 1024^2 píxeles y de 18 para resoluciones de 512^2 píxeles comparados con la implementación en CPU para un único hilo.

En otros tipos de tomografía, como la tomografía de electrones, se han empleado aproximaciones de paralelismo mediante vectorización de instrucciones para acelerar el proceso de reconstrucción, obteniendo optimizaciones de métodos iterativos como SIRT mediante instrucciones SIMD (Agulleiro y col. 2010).

Es evidente que el tiempo es un factor decisivo en este ámbito, y que el desarrollo de nuevos métodos pasa necesariamente por una optimización del software para explotar al máximo los recursos hardware que existen actualmente como sistemas *multicore* o GPUs, con el fin de reducir los tiempos computacionales típicamente elevados de los métodos algebraicos.

1.4 Estructura del documento

El documento de tesis se ha desarrollado mediante compendio de artículos, por lo que la estructura se ha adaptado consecuentemente, contando con el presente capítulo y los capítulos 2 y 3 a modo introductorio, seguido por un capítulo por cada artículo publicado, donde se realiza un resumen del contenido, seguido del propio artículo adaptado al formato del documento. Los artículos han sido

incluidos en su versión original sin traducir. El último capítulo está dedicado a la conclusión general y los trabajos futuros.

En el capítulo 2 se realiza una introducción histórica sobre la Tomografía Computarizada, así como a los conceptos básicos de la obtención de los datos y los escáneres utilizados en este prueba. Además, se introducen conceptos relacionados con la medida de la radiación y se muestran la dosis que conllevan algunas pruebas de TC.

Por otra parte, se define el problema de reconstrucción de la imagen, y se introducen los métodos que van a ser empleados para resolverlo, así como las métricas de calidad empleadas para evaluar las reconstrucciones obtenidas.

En el capítulo 3 se realiza una introducción a la Computación de Altas Prestaciones, definiendo las diferentes arquitecturas disponibles, así como los diferentes tipos de paralelismo. Además, se enumeran y definen tanto las herramientas hardware que se han empleado durante el desarrollo de la tesis, como las herramientas software.

Los capítulos 4-9 corresponden a los artículos publicados. En el capítulo 4 se presenta un estudio multiparamétrico sobre el método LSQR. Dicho estudio se ha desarrollado en una plataforma Grid para emplear paralelismo de grano grueso, realizando un barrido de todos las combinaciones posibles de los parámetros que intervienen en esta técnica de reconstrucción y pueden influir sobre la calidad final de la imagen. Debido a la necesidad de instalación de software así como de disponer de los datos de entrada, se analiza el uso de los elementos de almacenamiento de la plataforma Grid para la descarga de los datos, y de emplear contenedores Docker que ya tengan instalado el software, realizando una comparativa de la eficiencia de cada aproximación.

El estudio que se presenta en el capítulo 5 está basado también en el método LSQR. Se trata de una evaluación de distintos filtros de imagen para su posible incorporación al proceso de reconstrucción. En él, se han simulado proyecciones con ruido para observar el efecto de los filtros, tanto sobre el sinograma como sobre las propias imágenes. Finalmente, se ha seleccionado el filtro bilateral por aportar los mejores resultados, y se ha analizado su comportamiento en combinación con los otros pasos de la reconstrucción.

El capítulo 6 contiene un estudio preliminar sobre el uso de métodos algebraicos directos para resolver el problema de reconstrucción de TC. En él se realiza una explicación de las técnicas SVD y QR y cómo pueden ser utilizadas para resolver el problema. Además, se analizan los resultados obtenidos para distintas resoluciones de imagen (y por tanto, distintos tamaños de proble-

ma), analizando el comportamiento y el tiempo de cada método, así como los requerimientos de memoria, y comparando las imágenes obtenidas.

De los resultados obtenidos surge el estudio que se presenta en el capítulo 7, en el cual se ha analizado más en profundidad el método QR, así como las prestaciones que se pueden obtener mediante la implementación multihilo que emplea la librería utilizada. Además, se compara la técnica tradicional que emplea la matriz Q formada explícitamente, con el uso de las reflexiones de Householder, que puede permitir ahorrar tiempo y memoria, lo cual es crucial como se verá más adelante.

En el capítulo 8 se realiza una comparación de los métodos LSQR y QR para reconstruir imágenes de TC. Para ello, se ha hecho uso de un *dataset* de imágenes reales que contienen lesiones de distinto tipo debidamente etiquetadas. De este modo, se ha podido comparar la calidad obtenida por ambos métodos, tanto mediante las métricas de imagen, como con la detección visual de las lesiones. Además, se ha analizado las fortalezas y debilidades de cada uno de ellos, planteando los escenarios en los que se pueden emplear según el objetivo a lograr.

Una implementación de método QR con técnicas Out-Of-Core se presenta en el capítulo 9, En él, se realiza una introducción a estas técnicas, que permiten aumentar el tamaño del problema a resolver basándose en el uso de disco y no de memoria RAM. Esto es crucial debido a que el problema de reconstrucción aumenta de tamaño para resoluciones altas, y la memoria insuficiente es el factor que en los estudios anteriores no ha permitido llegar a la máxima resolución. En el estudio se detallan cuatro variantes del método: utilizando discos HDD o SSD, y empleando o no solapamiento de lectura/escritura con cálculos para lograr mayor eficiencia.

Finalmente, en el capítulo 10 mostramos las conclusiones generales de todos los estudios realizados, así como un planteamiento de posibles trabajos futuros para continuar con esta línea de investigación.

Capítulo 2

Tomografía Computarizada

2.1 Descubrimiento de los Rayos X

Con el descubrimiento de los rayos X en el año 1895 de mano de Wilhelm Röntgen nació una nueva disciplina dentro de la medicina: el diagnóstico por imagen. Este científico alemán hacía pruebas con un aparato que generaba rayos catódicos (tubos de Crookes) para investigar la fluorescencia violeta. Mientras experimentaba con los rayos catódicos, Röntgen percibió que un cartón situado a unos metros del tubo emitía un resplandor. Tras sellar el tubo con un cartón negro opaco, y determinar que el campo de acción de los rayos catódicos era más corto que la distancia a dicho cartón resplandeciente, concluyó que había algún tipo de rayo desconocido e invisible que estaba viajando a través de la sala y penetrando en él, que estaba pintado con una capa de platino-cianuro de bario.

Después de este hallazgo, el físico realizó varias pruebas empleando placas fotográficas para comprobar el alcance y la potencia de penetración de estos rayos, a los que denominó rayos X por ser una incógnita. Durante semanas estudió sus propiedades, para presentar un informe en el año 1896 en el artículo "Sobre una nueva clase de rayos" (Röntgen 1896). Las propiedades que descubrió Röntgen son prácticamente todas las que se conocen hoy en día.

Figura 2.1: Mano con anillos. Primera radiografía médica tomada por Wilhelm Röntgen. 22 de diciembre de 1895.

Entre los resultados, destaca el estudio sobre el poder de penetración de los rayos X sobre distintos materiales como aluminio, madera, platino, plomo, papel, o el propio cuerpo humano. Determinó que penetran más en materiales con baja densidad, como pueden ser la carne, madera, o papel, que en materiales densos como hueso o plomo. Para ilustrar esto, mostró la primera radiografía humana de la historia: la mano de su esposa, tal y como se aprecia en la Figura 2.1, con la alianza en uno de sus dedos. Como se observa, los rayos X penetran completamente la carne. Sin embargo, tanto el hueso como la alianza, de material más denso, son más difíciles de penetrar y por ello dejan una sombra en la placa fotográfica.

Todo este estudio le permitió ser galardonado con el premio Nobel de Física en el año 1901, y aunque el propio Röntgen no siguió sus investigaciones en este campo, su descubrimiento fue el punto de inicio para las investigaciones de muchos otros. Y pese a que tarda unos años en tener aplicación real en la medicina, el uso de radiografías se empieza a implantar en los hospitales a

principios el siglo XX después de que la comunidad científica ve su utilidad para realizar diagnósticos.

Sin embargo, pronto se puso en foco la nocividad de los rayos X para los seres vivos. Desde el propio descubrimiento, hubo científicos que advirtieron sobre los efectos adversos, como Thomas Edison, y en 1986 Wolfram C. Fuchs desarrolló una guías para reducir el tiempo de exposición y la distancia al foco, recogidas en (Clarke y Valentin 2009) con el fin de reducir la radiación a la que estaba expuesto el paciente. Desde el año 1901, el científico y dentista W. Rollings advertía de los peligros, verificando sus teorías sobre el daño producido a fetos mediante experimentación animal. Rollings introdujo términos como colimación, filtración, efectos tardíos y llegó a desarrollar una serie de recomendaciones en cuanto a protección de partes del cuerpo a través del uso de plomo.

Sus estudios no fueron popularmente aceptados hasta años más tarde, cuando se hizo más que evidente la relación de casos de carcinomas y sarcomas con la exposición a rayos X. En el año 1928 se pone en marcha el "Comité Internacional de Protección para los Rayos X y el Radio", el cual abría una nueva disciplina en el campo de la física: la protección radiológica". Científicos de renombre como la asistente de Edison, Clarence Dally, W. C. Fuchs, o la propia Marie Curie, que se dedicaron a investigar en dicho campo sufrieron en sus carnes los efectos de la radiación, falleciendo los dos primeros por cáncer y la tercera por leucemia.

2.2 De los Rayos X a la Tomografía Computarizada

Desde el inicio de la aplicación de los rayos X en el campo de la medicina se advirtió el principal problema de las radiografías planas. Al ser proyecciones bidimensionales de objetos tridimensionales, la zona de interés se ve oscurecida por las sombras de las zonas subyacentes en el plano anterior y posterior. Por lo tanto, el diagnóstico se complica al no ser posible centrarse en un plano escogido. La primera aproximación para resolver este problema fue la Tomografía clásica. El concepto básico era mover la fuente de rayos X alrededor del paciente, y la placa fotográfica en dirección opuesta, para obtener imágenes en las que el objeto en el centro de rotación (punto de interés) se viera nítido y el resto de estructuras se vieran borrosas. La ilustración de este efecto se observa en la Figura 2.2. En ella se aprecia cómo si se desplaza la fuente en sentido opuesto a la placa, los objetos que no estén en el plano de interés (verde y naranja), se van a proyectar desplazados por el movimiento, y se verán borrosos.

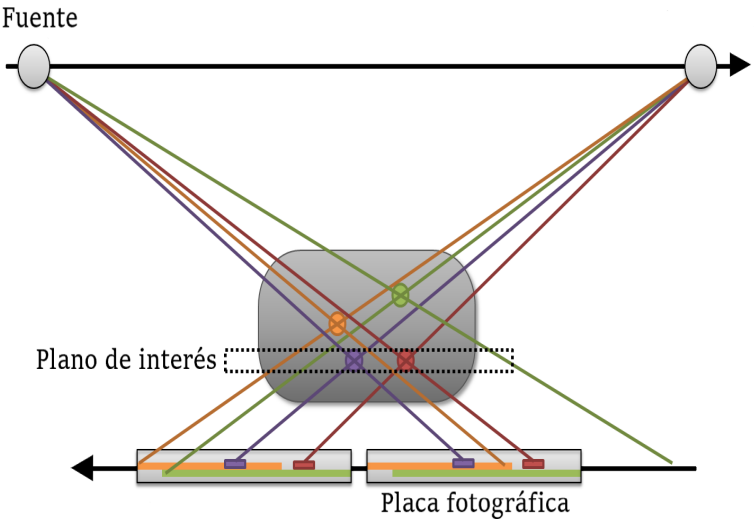


Figura 2.2: Principio de la Tomografía clásica.

Sin embargo, los que están situados en el plano de interés (morado y rojo), van a verse siempre en la misma posición, por lo que aparecerán nítidos en la radiografía.

Esta idea se sugirió por primera vez en el año 1914, pero pasaron años hasta que se puso en práctica. Pese a que la primera patente completa y viable fue presentada por el médico parisino André Bocag en el año 1921, no fue hasta 1930 cuando se construye una máquina para tomografía y se obtiene la primera imagen con este método. Esto se hace de la mano de Allesandro Vallebona, que se convirtió en uno de los pioneros en este campo. A partir de ahí se desarrollan varias máquinas de tomografía usando diferentes técnicas. La primera tomografía realizada a un paciente se lleva a cabo en 1950 por el radiólogo británico William Watson. En (Buzzi y Suárez 2013) se puede encontrar una explicación detallada de la evolución de esta técnica.

A pesar de los avances que se iban realizando en cuanto a esta técnica, el problema de inicio seguía ocurriendo, aunque en menor medida. En la tomografía clásica era complicado distinguir estructuras adyacentes, sobretodo entre tejidos blandos como puede ser tejido sano y un tumor. En el año 1963, el físico Allan Cormack propone dejar de utilizar placas fotográficas para captar los rayos X y en su lugar medir la atenuación que han sufrido los rayos al atra-

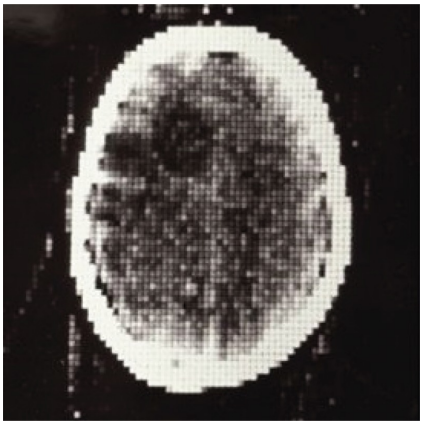


Figura 2.3: Primera imagen clínica de CT, 1971.

vesar un objeto, para poder de esa manera determinar la densidad de dicho objeto. Para lograr esto habría que captar y procesar los datos de forma digital, combinando las diferentes proyecciones tomadas para obtener una imagen. Posteriormente se determinó que muchas de las ideas que proponía Cormack ya habían sido introducidas en 1917 por Johan Radon.

Esta idea supuso el punto de partida para el ingeniero Godfrey Hounsfield, que propuso a la compañía EMI el desarrollo de un escáner con estas características. Con los fondos de la compañía, Hounsfield inició el desarrollo, y obtuvo el primer prototipo en 1967. La resolución espacial de las primeras imágenes era muy pequeña (80×80 píxeles), pero suficiente para poder distinguir estructuras. Esto quedó probado en el año 1971, cuando se presentó el primer escáner clínico cerebral llamado EMIMARK1 (descrito en (Hounsfield 1973)), con el cual se realizó un estudio en el Hospital Atkinson Morleyen Londres a una paciente que tenía un tumor en el lóbulo frontal, mostrado en la Figura 2.3, consiguiendo así intervenirla de manera exitosa con ayuda de las imágenes.

Desde ese momento empieza el auge del TC y la decadencia de la tomografía clásica, extendiéndose su uso en los hospitales de todo el mundo. Tanto así que en el año 1979 Hounsfield y Cormack son galardonados conjuntamente con el premio nobel de medicina.

2.3 Generaciones de Escáneres

Desde el primer escáner desarrollado (EMIMARK1), la tecnología fue evolucionando, y con ello la forma de obtener las proyecciones. En la Figura 2.4 se muestran las principales generaciones y su funcionamiento (Flohr 2013).

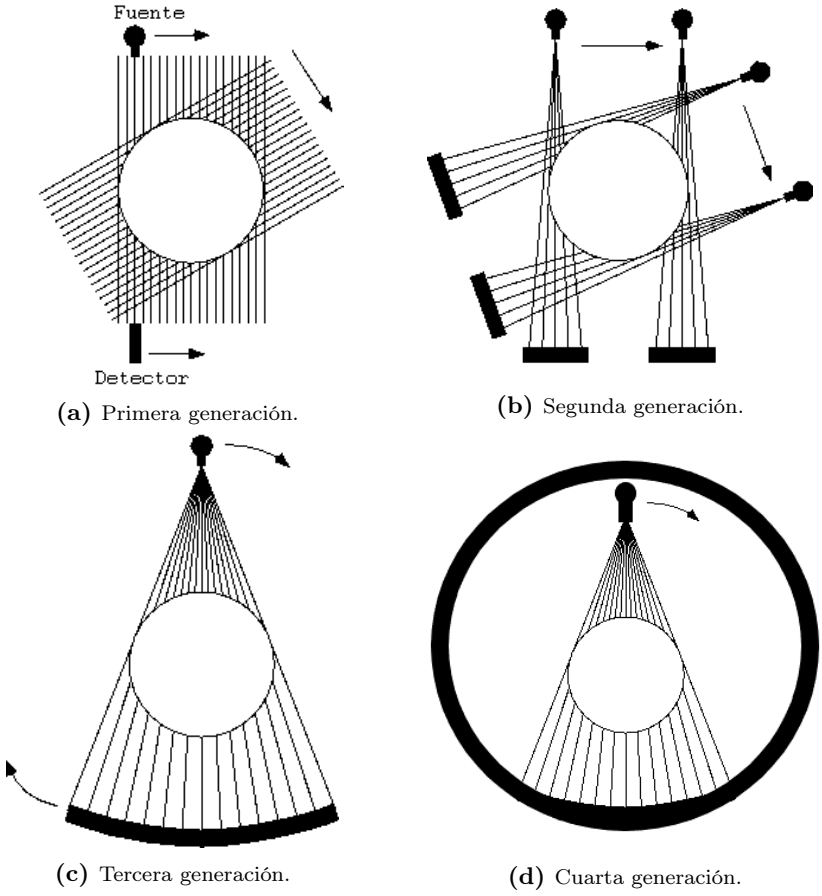
En la primera generación, el haz de rayos era paralelo. Había un único detector, que junto con la fuente debía realizar un movimiento de traslación (para tomar todos los datos de una proyección a lo largo del objeto), y de rotación (para tomar proyecciones en distintos ángulos. Con este tipo de escáner sólo se tomaban proyecciones en los primeros 180^{o} alrededor del paciente.

Esta generación estaba restringida a algunas partes del cuerpo, sobretodo el cerebro, ya que tardaba 4.5 minutos en realizar el estudio, y el paciente no podía mover la zona en todo ese tiempo. Una mejora posterior en esta generación fue utilizar dos detectores para reducir el tiempo a la mitad. Aún así, la resolución lograda era muy baja.

En la segunda generación se introduce el concepto de *fanbeam*, o haz de rayos en abanico. De esta manera, en cada disparo se podía obtener más. Todavía es necesario tanto movimiento de traslación como de rotación, pero el tiempo de escaneo se puede reducir dependiendo del número de detectores. En los primeros modelos, que contaban con 3 detectores, se tomaban proyecciones cada 3^{o} en lugar de 1^{o}, cubriendo los 180 grados con sólo 60 disparos en lugar de 180. Por lo tanto, el tiempo requerido se dividía por 3. Posteriormente se fueron aumentando el número de detectores hasta 53, logrando ya tiempos de decenas de segundos, lo que permitió realizar estudios de tórax mientas el paciente aguantaba la respiración para evitar el movimiento.

La tercera generación supuso un avance muy notable. Con ella, se aumenta el ángulo del haz de rayos para que cubra al paciente y así eliminar el movimiento de traslación. Además, se aumenta el número de detectores (desde 200 hasta 700 aproximadamente) y pueden estar colocados en forma de arco. Tanto la fuente como el array de detectores rotan, en sentidos opuestos, hasta cubrir los 360^{o} alrededor del paciente. Ahora el tiempo de un estudio se reduce a varios segundos (5 segundos de media), lo que supone una gran mejora.

Con el aumento del número de detectores también se incrementa la resolución de las imágenes. Dentro de esta generación se incluyen varias mejoras, como el array de detectores con múltiples filas (permitiendo así captar diferentes cortes al mismo tiempo), el escáner de doble fuente, o el escáner helicoidal, en el que se combina el movimiento de la fuente y detectores con un movimiento



(a) Primera generación.

(b) Segunda generación.

(c) Tercera generación.

(d) Cuarta generación.

Figura 2.4: Generaciones principales de escáner TC.

de traslación de la camilla para así no tener que realizar el estudio corte a corte.

Con la cuarta generación se aumenta el número de detectores (hasta 4000) y se colocan en forma de anillo estacionario. Por lo tanto, ahora sólo rota la fuente, y los detectores están fijos. Se reduce así la complejidad de los movimientos, pero en este caso se desaprovecha el aumento de los detectores. No por tener más detectores se tiene mayor resolución que en la generación anterior, ya que el haz en cada disparo va a llegar aproximadamente al mismo número de detectores que antes. Por tanto, aún siendo más caro, no aporta

muchas ventajas, ya que el tiempo se reduce pero no lo suficiente como para que suponga una gran diferencia.

Por todo ello, el escáner de tercera generación ha sido y sigue siendo el más usado, tanto en su versión corte a corte como helicoidal. Actualmente la mayoría utilizan array de detectores con múltiples filas y doble fuente.

El modelo de escáner empleado durante el desarrollo de esta tesis es de tercera generación, con haz de abanico o *fanbeam*, y una sola fila de detectores. De esta forma, la reconstrucción de volúmenes en 3 dimensiones se realizará corte a corte, que podrán resolverse de forma simultánea con los métodos propuestos, como se verá en posteriores capítulos.

2.4 Dosis de Radiación

Como ya se ha explicado con anterioridad, los rayos X que se utilizan en una prueba de TC son un tipo de radiación ionizante, por lo que tienen un efecto en la materia sobre la que se deposita. A continuación se introducen lo diferentes términos utilizados en el cálculo de la dosimetría.

Dosis absorbida

La dosis absorbida por un cuerpo es equivalente a la energía depositada por unidad de masa del cuerpo expuesto. Esta medida cuantifica la absorción de la radiación en la sustancia sin tener en cuenta los efectos biológicos. La unidad usada es el Gray (Gy), que representa la absorción de un julio de energía por kilogramo de materia ($1Gy = 1J/kg$).

La dosis absorbida por un órgano o tejido T por la incidencia de un tipo de radiación R se expresa como $D_{T,R}$ según la ecuación (2.1). Generalmente en cualquier campo la radiación está por debajo de un Gray, por lo que se suele expresar en miliGrays (mGy). Dado que esta medida no contempla el efecto biológico que la radiación puede tener en la materia se planteó la medida de la dosis equivalente.

$$D_{T,R} = \frac{E}{m} \tag{2.1}$$

Dosis equivalente

A menudo es interesante saber cómo cierta tipo de radiación afecta a un cuerpo para cuantificar el posible efecto adverso. La dosis equivalente mide justamente esto, el efecto de la radiación sobre tejidos vivos. La dosis equivalente H sobre un órgano o tejido para un tipo de radiación R se calcula según la ecuación (2.2), donde w_R es un factor de ponderación y $D_{T,R}$ la dosis absorbida. La unidad del Sistema Internacional para medir la dosis equivalente es el Sievert (Sv).

$$H_{T,R} = w_R * D_{T,R} \tag{2.2}$$

El factor de ponderación depende del tipo de radiación, y para rayos X es igual a 1, por lo que $1Sv = 1Gy$.

Dosis efectiva

Con el estudio de los efectos de la radiación en el cuerpo se ha observado que el daño sufrido no sólo depende del tipo radiación sino que el tejido influye, ya que unos son más sensibles que otros. Por tanto, en ciertos órganos la misma cantidad de dosis equivalente provocará menos daño que en otros. Para cuantificar este efecto se utiliza la dosis efectiva, que también se mide en Sieverts y es el resultado de multiplicar la dosis equivalente $H_{T,R}$ por un factor de ponderación según el tipo de tejido tal como se muestra en la ecuación (2.3). Si más de un órgano es expuesto, entonces la dosis efectiva total es la suma de la dosis efectiva de cada órgano.

$$E_{T,R} = w_T * H_{T,R} \tag{2.3}$$

En la Tabla 2.1 se pueden observar los factores para cada tipo de tejido. Como se aprecia, hay órganos como el pulmón o el estómago que son mucho más sensibles a la radiación que otros como el cerebro o el hígado.

Tejido	Factor de ponderación por tejido	Suma de los factores de ponderación
Médula ósea, colon, pulmón, estómago, mama, tejidos restantes(*)	0.12	0.72
Gónada	0.08	0.08
Vejiga, esófago, hígado, tiroide	0.04	0.16
Superficie ósea, cerebro, glándulas salivales, piel	0.01	0.04
	Total	1.00

(*) Tejidos restantes: Suprarrenales, región extratorácica, vesícula biliar, corazón, riñones, ganglios linfáticos, músculo, mucosa oral, páncreas, próstata, intestino delgado, bazo, timo, útero/cérvix

Tabla 2.1: Factor de ponderación según el tipo de tejido.

Procedimiento	Dosis Efectiva Aproximada	Tiempo de radiación natural de fondo
TC tórax	8.8 mSv	3 años
TC columna	8.8 mSv	3 años
Angiografía coronaria por TC	8.7 mSv	3 años
TC cerebro	1.6 mSv	6 meses
TC dental	0.18 mSv	22 días
Rayos X de la columna lumbar	1.4 mSv	6 meses
Rayos X de las extremidades (mano, pie, etc.)	0.001 mSv	3 horas

Tabla 2.2: Equivalencia de dosis efectiva con radiación natural de fondo.

Radiación natural de fondo

Además de la exposición a radiación por fuentes artificiales, también nos vemos expuestos a radiación ionizante generada por fuentes naturales. Hay isótopos radiactivos en el cuerpo, el agua, el aire, etc. Todos también estamos expuestos a la radiación del espacio cósmico, pero la mayor fuente de exposición natural es el gas Radón. La exposición a cada una de esas fuentes depende principalmente del lugar de residencia ya que habrá elementos radiactivos que estén más presentes en unos lugares que en otros.

Se estima que la radiación natural de fondo anual por persona es de entre 2 y 3 mSv dependiendo de la zona, aunque puede llegar a ser mucho más elevada en algunos lugares como India o Brasil. En la Tabla 2.2 se recogen una serie de equivalencias de estudios que usan rayos X con el tiempo que se tardaría en estar expuesto a la misma dosis de manera natural. Como se puede observar, una fuente artificial como el TC incrementa en gran cantidad la exposición que el cuerpo recibe de fondo.

2.5 Principios Básicos

EL objetivo de la prueba de TC es medir la atenuación que sufren los rayos X al atravesar el cuerpo de un paciente. Para obtener las imágenes es necesario tomar diferentes proyecciones (disparos de rayos X) alrededor del paciente, rotando tanto la fuente como los detectores que están al otro extremo para captar lo que llega. El array de detectores está formado por entre 600 y 1000 detectores individuales y puede tener múltiples filas según la generación del escáner, como se ha explicado anteriormente.

Para reconstruir la imagen deseada es necesario saber el llamado coeficiente de atenuación lineal de cada material (μ). Este coeficiente representa la capacidad de un material para detener fotones. Es directamente proporcional al número atómico del material y su densidad, e inversamente proporcional a la energía del haz. Por lo tanto, un material más denso tendrá un coeficiente de atenuación lineal más alto, y bajará conforme aumente la energía. Los materiales con coeficiente de atenuación alto se verán más blancos en la imagen, y los bajos se verán más negros.

De este modo, cada material tendrá un coeficiente de atenuación distinto, dependiendo también de la energía del haz. Además, depende del espesor del cuerpo que atraviesan los rayos. Por tanto, la fórmula utilizada para calcular la intensidad de un haz que se ha adentrado x centímetros en un objeto queda definida según la ecuación (2.4), siendo I_0 la intensidad inicial del haz y μ el coeficiente de atenuación del material.

Si se quiere calcular la intensidad de un rayo en todo el trayecto desde la fuente al detector se hará mediante la fórmula presentada en la ecuación (2.5), donde s es la trayectoria seguida por el rayo. Desde el punto de vista de la reconstrucción será necesario discretizar esta ecuación.

$$I = I_0 e^{-\mu(x)} \tag{2.4}$$

$$I = I_0 e^{-\int_s \mu(x)dx} \tag{2.5}$$

Rotando la fuente alrededor del objeto, obteniendo proyecciones en un rango $\theta = [0, \pi]$ según la geometría que se muestra en la Figura 2.5 y colocando las proyecciones $p(\theta, x)$ obtenidas en una matriz se conforma el sinograma, que tiene la forma que se presenta en la Figura 2.6. Si la fuente es de rayos en abanico y no paralela, habría que obtener proyecciones en el rango $\theta = [0, 2\pi]$.

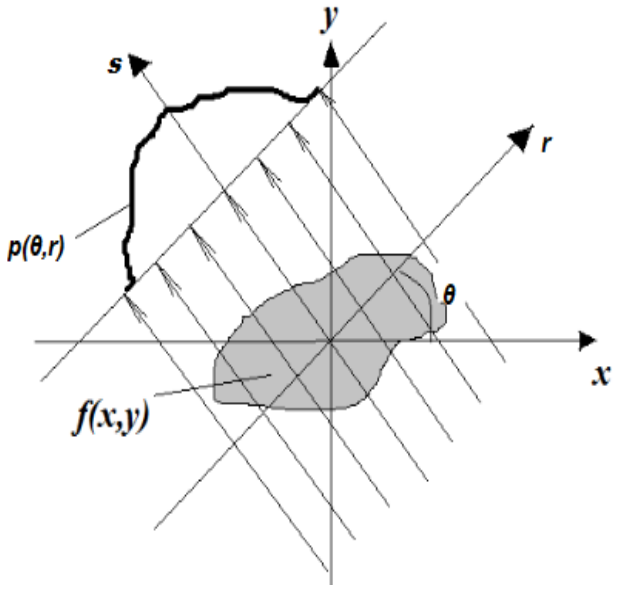


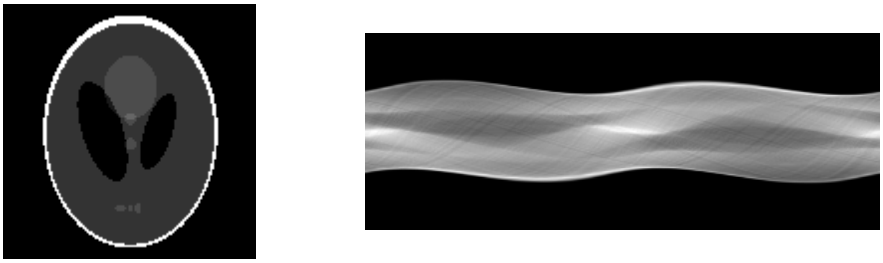
Figura 2.5: Geometría paralela de las proyecciones.

El cálculo de estas proyecciones siguiendo un esquema de geometría paralela como el que se presenta en la Figura 2.5 se calcula mediante la llamada Transformada de Radon. Rotando la fuente alrededor del objeto se forma el sistema de coordenadas (r, s), cuya relación con el sistema de coordenadas (x, y) es la que se muestra en la ecuación (2.6).

Para obtener las distintas proyecciones se realizarán disparos en diferentes ángulos θ y se generará el llamado sinograma. Para ello, se calcula el conjunto de valores para cada detector de la proyección lanzada en el ángulo θ, según la ecuación definida en (2.7), donde (x, y) son las coordenadas de cada punto en el plano de dos dimensiones definido que atraviesa el rayo, pudiendo calcular $I(\theta, r)$ según la ecuación (2.8).

$$\begin{bmatrix} r \\ s \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.6}$$

$$p(\theta, r) = -ln(\frac{I(\theta, r)}{I_0}) = \int_s \mu(rcos\theta - ssin\theta, rsin\theta + scos\theta)ds \tag{2.7}$$



(a) Corte central del fantoma Shepp-Logan **(b)** Sinograma con 137 detectores y 360 proyecciones

Figura 2.6: Fantoma Shepp-Logan y su sinograma tras proyectarlo.

$$I(\theta, r) = I_0 e^{-\int_s \mu(rcos\theta - ssin\theta, rsin\theta + scos\theta)ds} \tag{2.8}$$

Las imágenes reconstruidas se representan en tonos de gris mediante las llamadas Unidades Hounsfield HU, o número CT. La relación entre los coeficientes de atenuación y las unidades Hounsfield de un material se presenta en la ecuación (2.9). Tal y como se puede observar, se representa en relación al coeficiente de atenuación del agua. El HU del agua destilada en condiciones de presión y temperatura estándar es 0, y el coeficiente de atenuación lineal del aire también es 0 puesto que no opone resistencia a los rayos.

$$HU_{mat} = \frac{\mu_{mat} - \mu_{agua}}{\mu_{agua} - \mu_{aire}} \times 1000 \tag{2.9}$$

En estas unidades, el hueso tiene un número CT de entre 400 y 1000 dependiendo de su densidad, el aire de -1000, y entre estos dos valores se encuentran el resto de elementos del cuerpo, como puede ser la grasa (entre -100 y -50), tejidos blandos (entre 10 y 60), etc. La lista completa de las unidades Hounsfield de cada material se puede observar en la Figura 2.7.

Estas unidades se traducen a valores de gris. Por ello, las imágenes deben estar representadas en una profundidad mínima de 12 bits, lo que hace que se puedan representar valores de -1024 a 3001, cubriendo así la escala completa de unidades Hounsfield en el cuerpo humano, Cabe mencionar que hay materiales con un número CT por encima de 3000, como algunos metales que se pueden utilizar en implantes o suturas (acero o titanio), por lo que la representación de las imágenes se puede extender a 14 bits (15359 HU) para incluir estos materiales.

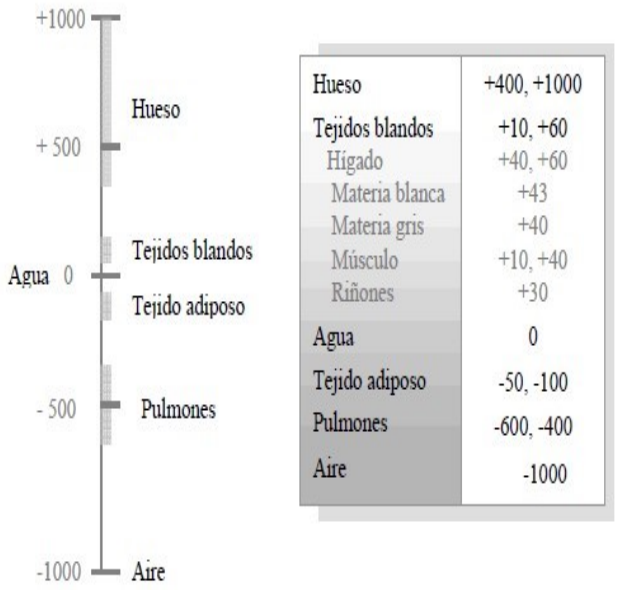


Figura 2.7: Unidades Hounsfield por material (Ramírez Giraldo, Arboleda Clavijo y McCollough 2008).

A la hora de visualizar las imágenes reconstruidas, es importante la ventana elegida. El nivel de la ventana y el ancho de la ventana permiten ajustar el contraste y centrarse en el tejido deseado. Por tanto, estos parámetros variarán según lo que se quiera estudiar. Por ejemplo, el nivel y ancho de ventana no será el mismo si se quiere estudiar el tejido de los pulmones (entre -600 y -400 HU), que el del hígado (entre 40 y 60 HU).

2.6 Modelado del problema para resolución algebraica

El problema de reconstrucción de imágenes de TC mediante métodos algebraicos se puede modelar como un sistema de ecuaciones lineales dependiente de las características físicas del escáner (ángulo de apertura de haz, distancias, posicionamiento del paciente, etc.). Dicho sistema está definido en la ecuación (2.10). Para resolverlo, necesitamos dos conjuntos de datos de entrada. Primero, la matriz del sistema A, que se muestra en la ecuación (2.11), se calcula una vez para cada resolución deseada de la imagen reconstruida si las características del escáner no cambian. Por otro lado se necesita el vector de

proyecciones b (ecuación 2.12). Dicho vector es el sinograma obtenido de un paciente u objeto, que será diferente para cada resolución y para cada sección proyectada de un objeto. Ambos conjuntos de datos deben generarse para un modelo de escáner CT con las mismas características y configuración.

La matriz A es de tamaño $M \times N$, donde M es el número de rayos trazados y N es el tamaño en píxeles de la imagen a reconstruir. Esta matriz se obtiene discretizando el espacio de escaneo en píxeles y midiendo la influencia de cada haz que se traza en cada píxel, según lo definido por la ecuación (2.11), donde $a_{i,j}$ representa la contribución de píxel j al rayo i. Para calcular los pesos de cada rayo se ha aplicado el método de proyección hacia adelante o *Forward Projection* propuesto por Joseph en (Joseph 1982).

El vector de proyecciones es el cálculo de la atenuación que experimentan los rayos que atraviesan el objeto de estudio. El tamaño del vector para cada ángulo de proyección es $1 \times n$, siendo n el número de detectores que forman el escáner. Al concatenarlos todos obtenemos un vector unidimensional que representa el objeto completo de tamaño M. Este vector de proyecciones es el propio sinograma representado en una dimensión, ya que en lugar de colocar las proyecciones formando una matriz que se puede visualizar como una imagen (similar a la presentada en la Figura 2.6b), se colocan por orden en un vector unidimensional. Por ejemplo, un sinograma que se haya obtenido en un escáner de 700 detectores y 180 ángulos de proyección tendría dimensión 700×180, y su vector de proyecciones equivalente $1 \times (700 \times 180) = 1 \times 126000$.

Por último, la solución al sistema es el vector representado en la ecuación (2.13), de tamaño N. Dicho vector, de forma análoga al vector de proyecciones, es la representación unidimensional de la imagen reconstruida. Es decir, si se representa como una matriz cuadrada de $\sqrt{N}$ filas y columnas se visualizaría la imagen solución del problema.

Por tanto, se observa que el tamaño del problema viene definido por el número de rayos trazados, que será dependiente del número de proyecciones tomadas y el número de detectores, y la resolución de la imagen a reconstruir. La matriz resultante de la discretización del espacio físico suele ser rectangular, dispersa y de dimensiones elevadas si se quieren obtener imágenes de buena resolución.

En la Figura 2.8 se puede observar la estructura típica de la matriz del sistema. En este caso en concreto la imagen se corresponde a la submatriz para las 20 primeras proyecciones obtenidas con 700 detectores (14000 filas), y los primeros 15000 píxeles de la imagen (correspondiente a una resolución de imagen de $512 \times 512 = 262144$ píxeles). Como se indica en la imagen, el número de

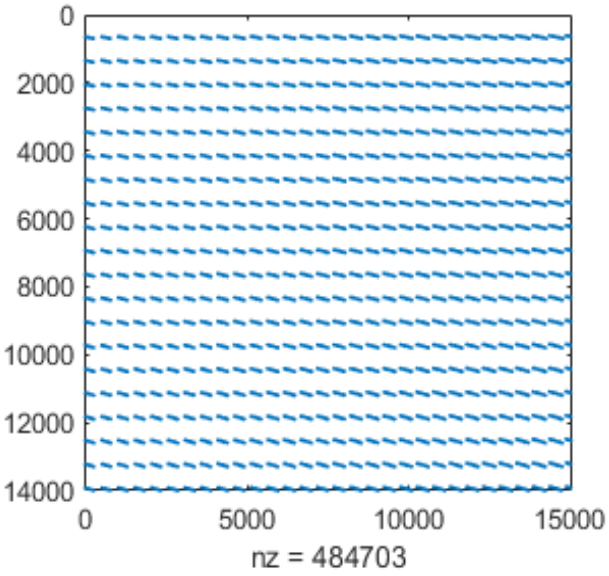


Figura 2.8: Estructura de la matriz del sistema.

elementos no cero es 484703, lo que supone un 0.23% del número de elementos de dicha submatriz.

$$A * x = b \tag{2.10}$$

$$A = a_{i,j} \in R^{MxN} \tag{2.11}$$

$$b = [b_1, b_2, ..., b_M]^T \in R^M \tag{2.12}$$

$$x = [x_1, x_2, ..., x_N]^T \in R^N \tag{2.13}$$

2.7 Resolución mediante Least Squares QR

El método algebraico iterativo que más se ha estudiado durante el desarrollo de esta tesis ha sido el Least Squares QR (LSQR), una implementación del gradiente conjugado, propuesto en (Paige y Saunders 1982) como un algoritmo iterativo con un comportamiento muy adecuado para resolver sistemas de ecuaciones lineales cuya matriz asociada es dispersa y rectangular, por lo que se adapta muy bien al problema a resolver. Además, los propios autores afirman que el método LSQR funciona especialmente bien para sistemas mal condicionados, lo que suele ocurrir cuando se reduce el número de proyecciones tomadas.

Este algoritmo ya ha sido empleado para resolución de imagen TC en trabajos previos del grupo de investigación, como (Flores, Vidal y Verdú 2015) y (Parcero y col. 2017), habiendo determinado en éstos la buena convergencia así como la capacidad de obtener imágenes de buena calidad incluso cuando el número de vistas tomadas es muy reducido.

El método LSQR resuelve el sistema de ecuaciones definido en la ecuación (2.10) minimizando el error mediante la expresión (2.14) dentro de una secuencia de subespacios de Krylov. Para obtener la solución se va generando una secuencia de aproximaciones x_k tal que la 2-norma del residuo en la etapa k va disminuyendo monotónicamente. El residuo r viene definido por la ecuación (2.15).

$$\min_{x \in \mathbb{R}^N} ||b - Ax||_2, A \in \mathbb{R}^{MxN}, b \in \mathbb{R}^M \tag{2.14}$$

$$r_k = b - Ax_k \tag{2.15}$$

Esta técnica se basa en el método de bidiagonalización propuesto por Golub y Kahan en (Golub y Kahan 1965), y su objetivo es hallar la solución al sistema descrito en la ecuación (2.16), donde r es el vector residuo (2.15) y λ un escalar real arbitrario, minimizando la expresión (2.17).

$$\begin{pmatrix} I & A \\ A^T & -\lambda^2 I \end{pmatrix} \begin{pmatrix} r \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix} \tag{2.16}$$

$$\left\| \begin{pmatrix} A \\ \lambda I \end{pmatrix} x - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2 \tag{2.17}$$

El pseudocódigo correspondiente a este método se muestra en el Algoritmo 1, y es el propuesto originalmente en (Paige y Saunders 1982). Las operaciones que marcarán el coste computacional son las multiplicaciones matriz-vector (Av y $A^T u$).

Como se observa, el método realiza un máximo de iteraciones dadas, o bien itera hasta que alcanza convergencia. La convergencia se da cuando la 2-norma del residuo es menor o igual a una tolerancia dada, típicamente 10^{-6}. Por otra parte, la solución inicial x_0 podría ser un parámetro de entrada del algoritmo.

2.7.1 STF

Como paso extra en la reconstrucción algebraica iterativa se puede introducir el filtrado STF (Yu y Wang 2010; Yu y Zeng 2014). Este filtro con umbral recalcula la imagen a partir de la imagen de la etapa anterior según la ecuación (2.18). Como se observa, realiza una ponderación entre los gradientes horizontales-verticales y los diagonales, por lo que el filtro elimina ruido de la imagen conservando los bordes. La variable x es la solución previamente obtenida mediante el método LSQR. El parámetro α es seleccionado por el usuario y suele variar entre 0 y 2.

Pese a ser un filtro, esta técnica actúa como un proceso regularización, ya que sirve para completar información que se pierde al reducir las proyecciones tomadas, ayudando de esta manera a eliminar el ruido que esto provoca en la imagen aproximada mediante LSQR.

$$\begin{array}{c} x_{i,j}^{n+1} = \frac{1}{4+4\alpha}(q(w, x_{i,j}, x_{i+1,j}) + q(w, x_{i,j}, x_{i,j+1}) + \\ q(w, x_{i,j}, x_{i-1,j}) + q(w, x_{i,j}, x_{i,j-1}) + \alpha(q(w, x_{i,j}, x_{i+1,j+1}) + \\ q(w, x_{i,j}, x_{i+1,j-1}) + q(w, x_{i,j}, x_{i-1,j-1}) + q(w, x_{i,j}, x_{i+1,j-1}))) \end{array} \tag{2.18}$$

donde

$$q(w, y, z) = \begin{cases} (y+z)/2 \;\; if \;\; \mid y - z \mid < w \\ y - w/2 \;\; if \;\; y - z \geq w \\ y + w/2 \;\; if \;\; y - z \leq -w \end{cases}$$

y el umbral $w = \max \mid r_i \mid$, siendo r el vector residuo calculado como $r = A^T - (b - Ax)$ y r_i la componente i-ésima del vector residuo.

Algoritmo 1 LSQR

INPUT: Matriz A, Vector proyecciones b, Tolerancia tol
OUTPUT: Vector solución x

1) Inicialización

$\beta_1 u_1 = b$
$\alpha_1 v_1 = A^T u_1$
$w_1 = v_1$
$x_0 = 0$
$\overline{\phi}_1 = \beta_1$
$\overline{\rho}_1 = \alpha_1$

for i=1:maxIter **do**
2) Bidiagonalización

$\beta_{i+1} u_{i+1} = Av_i - \alpha_i u_i$
$\alpha_{i+1} v_{i+1} = A^T u_{i+1} - \beta_{i+1} v_i$

3) Construir y aplicar la siguiente transformación ortogonal

$\rho_i = (\overline{\rho}_i^2 + \beta_{i+1}^2)^{1/2}$
$c_i = \overline{\rho}_i / \rho_i$
$s_i = \beta_{i+1} / \rho_i$
$\theta_{i+1} = s_i \alpha_{i+1}$
$\overline{\rho}_{i+1} = -c_i \alpha_{i+1}$
$\phi_i = c_i \overline{\phi}_i$
$\overline{\phi}_{i+1} = s_i \overline{\phi}_i$

4) Actualizar los vectores x, w

$x_i = x_{i-1} + (\phi_i / \rho_i) w_i$
$w_{i+1} = v_{i+1} - (\theta_{i+1} / \rho_i) w_i$

5) Aplicar test de convergencia

if $\|b - Ax\|_2 \leq$ tol **then**

Salir

end if
end for

2.7.2 FISTA

Adicionalmente, se puede utilizar la técnica FISTA en combinación con los pasos anteriores. Mediante esta técnica de aceleración propuesta en (Beck y Teboulle 2009) se introducen nuevas direcciones de búsqueda de cara a la siguiente iteración de LSQR. Esto provoca una convergencia más rápida y por tanto una disminución de las iteraciones necesarias así como del tiempo de reconstrucción total.

Esta técnica se aplica mediante las fórmulas de la ecuación (2.19) y como se puede observar tiene un coste computacional muy bajo. En la primera etapa t está inicializado a 1 y x_{n-1} a 0.

$$\begin{aligned} t_{n+1} &= \frac{1+\sqrt{1+4t_n^2}}{2} \\ x_{n+1} &= x_n + (\tfrac{t-1}{t_{n+1}})(x_n - x_{n-1}) \end{aligned} \tag{2.19}$$

2.8 Resolución mediante métodos directos: QR y SVD

Como ya se ha mencionado anteriormente, otra estrategia para resolver el sistema de ecuaciones lineales que conforma el problema de resolución de imagen TC mediante métodos algebraicos puede ser el uso de factorizaciones y descomposiciones matriciales. A estos métodos los llamaremos directos puesto que no necesitan iterar para aproximar la solución, sino que la calculan mediante operaciones matriciales.

Para resolver el sistema de la ecuación (2.10) de forma totalmente directa sería necesario poder invertir la matriz A. Esto no es posible puesto que las matrices generalmente no son cuadradas. Si aún así se quisiera invertir la matriz A se podría trabajar con su pseudoinversa, calculada según la ecuación (2.20). Sin embargo, calcular explícitamente la matriz pseudoinversa también puede resultar prohibitivo para matrices grandes, como es el caso en este problema al trabajar con resoluciones grandes.

$$A^+ = (A^T A)^{-1} A^T \tag{2.20}$$

Es por ello que la idea aplicada en esta tesis es emplear las factorizaciones matriciales para simular la matriz pseudoinversa sin calcularla de manera explícita, y emplearla para resolver el sistema de ecuaciones lineales. Para ello,

se han empleado dos métodos: la descomposición en valores singulares, y la factorización QR.

Mediante la descomposición SVD, que se muestra en la ecuación (2.21), donde U y V son matrices ortogonales y Σ una matriz diagonal que contiene los valores singulares de la matiz A, se puede simular la pseudoinversa de la matriz conforme a la ecuación (2.22), y resolver el sistema tal como se muestra en la ecuación (2.23). Obtener la inversa de Σ es trivial puesto que consiste en calcular los valores recíprocos que conforman la diagonal.

$$A = U\Sigma V^T \tag{2.21}$$

$$A^+ = V\Sigma^{-1}U^T \tag{2.22}$$

$$x = A^+ b \tag{2.23}$$

De forma similar, calculando la factorización QR, que factoriza una matriz con columnas linealmente independientes en el producto de una matriz ortogonal (Q) y una matriz triangular superior R (2.24), se puede simular la pseudoinversa como (2.25), y resolver el sistema mediante (2.23) de manera sencilla ya que R es triangular e invertible, y puede no calcularse explícitamente mediante *backward substitution*.

$$A = QR \tag{2.24}$$

$$A^+ = R^{-1}Q^T \tag{2.25}$$

La explicación detallada de ambos métodos se presenta en el Capítulo 6.

2.9 Métricas de calidad para la evaluación de las imágenes

Con el fin de evaluar la calidad de las imágenes TC reconstruidas mediante los métodos desarrollados se han empleado distintas métricas de calidad de imagen que cuantifican el error, el nivel de ruido y el nivel de conservación de las estructuras internas, pudiendo así determinar la calidad guiados por varios criterios. Las métricas empleadas se presentan a continuación.

2.9.1 Error absoluto medio (MAE)

La métrica MAE mide el error absoluto medio entre la imagen referencia y la imagen reconstruida, y sirve para cuantificar la precisión. Es una métrica que depende de la escala de la imagen por lo cual no debe usarse para imágenes en diferentes escalas.

El MAE se calcula según la ecuación (2.26), donde n es el número total de píxeles de la imagen, y es la imagen reconstruida y x la imagen de referencia.

$$MAE = (\frac{1}{n}) \sum_{i=1}^{n} |y_i - x_i| \tag{2.26}$$

2.9.2 Error cuadrático medio (MSE)

El MSE es el error cuadrático medio entre la imagen reconstruida y la imagen original. MSE es una función de riesgo que corresponde al valor esperado del error al cuadrado. El MSE es el segundo momento de error y, por lo tanto, incorpora tanto la varianza de la estimación como su sesgo. El MSE viene dado por la ecuación (2.27), donde M y N son el ancho y la altura de las imágenes, x la imagen reconstruida y I_0 la imagen original.

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} ||I_0(i,j) - x(i,j)||^2 \tag{2.27}$$

2.9.3 Relación Señal a Ruido de Pico (PSNR)

La métrica PSNR mide el ratio entre la señal de la imagen y el ruido que contiene. Para calcularlo se utiliza el MSE, presentado anteriormente. El PSNR según la ecuación (2.28), en la cual MAX representa el máximo valor que puede tomar un píxel. El resultado es mejor cuanto más alto y usualmente se expresa en decibelios.

$$PSNR = 10 * log_{10} \frac{MAX_{I_0}^2}{MSE} \tag{2.28}$$

2.9.4 Índice de Similitud Estructural (SSIM)

A diferencia de las tres métricas anteriores, el Índice de Similitud Estructural mide la diferencia entre la imagen reconstruida y la imagen de referencia teniendo en cuenta su estructura interna, y no los valores de los píxeles. Es por ello que no mide diferencias de intensidad ni ruido, sino que compara y cuantifica la conservación de las diferentes estructuras de la imagen según lo bien que se haya conservado su forma.

El SSIM se aplica mediante ventanas de tamaño fijo, y se calcula la diferencia entre dos ventanas x e y correspondientes a las dos imágenes a comparar, mediante la ecuación (2.29). En dicha ecuación, μ_x y μ_y denotan el valor medio de la ventana x e y, σ_x^2 y σ_y^2 la varianza, σ_{xy} la covarianza entre las ventanas, y c_1 y c_2 son dos variables estabilizadoras dependientes del rango dinámico de la imagen.

Los valores de esta métrica están entre 0 (estructuras no coincidentes) y 1 (imágenes completamente iguales a nivel estructural).

$$SSIM = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{x,y} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{2.29}$$

Capítulo 3

Herramientas de Computación de Altas Prestaciones

La Computación de Altas Prestaciones (HPC por sus siglas en inglés) es un campo de las ciencias computacionales centrado en la resolución de problemas de alto coste computacional o alto volumen de datos. Este tipo de problemas existe en la mayoría de campos de la ciencia, desde simulaciones tipo Montecarlo, a análisis de datos mediante técnicas de *Big Data*, pasando por resolución de problemas matriciales asociados a un problema real discretizado. En cualquiera de estos casos, es imprescindible hacer uso de máquinas que proporcionen los suficientes recursos para poder resolver el problema planteado, así como lograr reducir el tiempo de cómputo para aumentar las posibilidades que ofrece la computación. Para ello, será necesario emplear eficientemente los recursos hardware de alto rendimiento disponibles mediante software especializado que explote los recursos de la máquina.

Dentro de la Computación de Altas Prestaciones, hay dos tendencias según el tipo de problema a resolver. Por una parte, la computación de alta productividad, basada en conseguir ejecutar el máximo de tareas independientes por unidad de tiempo. Esta técnica es útil para métodos que necesitan hacer muchos cálculos sin dependencias entre ellos, como pueden ser las simulaciones de Montecarlo, exploración de parámetros o algoritmos genéticos. Por otra parte,

la computación de alto rendimiento, cuyo objetivo es reducir el tiempo de ejecución una única aplicación, por lo que se pretende aumentar el rendimiento, o el número de operaciones de coma flotante realizadas por unidad de tiempo. Con esta aproximación habrá que realizar los cálculos de manera paralela entre los distintos *cores*, procesadores o máquinas, teniendo en cuenta las dependencias de datos. Al haber dependencias, será necesario que haya comunicación entre los procesos, ya sea mediante bus de datos o redes de interconexión.

A continuación se describen las diferentes arquitecturas de computación existentes, así como los distintos modelos de programación que se pueden emplear según la arquitectura. Además, se especifican las herramientas hardware utilizadas para el desarrollo de esta tesis, así como los recursos software empleados.

3.1 Arquitecturas Paralelas

La clasificación de las diferentes arquitecturas se suele realizar mediante la llamada taxonomía de Flynn, propuesta en (Flynn 1972). Dicha taxonomía clasifica los computadores según el número de datos y número de instrucciones que se pueden manejar simultáneamente. En la Figura 3.1 se observa la clasificación realizada por Flynn, así como los sistemas que emplean cada arquitectura. Tal como muestra la figura, existen cuatro arquitecturas paralelas:

- *Single Instruction Single Data (SISD)*: Arquitectura basada en el concepto de ejecutar una sola instrucción sobre un único dato, procesando los datos de uno en uno de manera secuencial. Esta arquitectura fue propuesta por John von Neuman en el año 1945, por lo que también se conoce como arquitectura von Neuman. Las máquinas que emplean este concepto cuentan con una única unidad aritmético-lógica, con sus correspondientes registros, y una unidad de control que gestiona el acceso a memoria. Esta arquitectura es secuencial, debido a que no puede procesar más de un dato y una instrucción simultáneamente en el mismo ciclo de reloj.

  Es la arquitectura más antigua, y algunos computadores conocidos que emplearon esta arquitectura son el Mark-1 o el ENIAC, entre otros.

- *Single Instruction Multiple Data (SIMD)*: Aplicación de una instrucción sobre múltiples datos diferentes en el mismo ciclo de reloj. Esta arquitectura paralela describe máquinas con varias unidades de procesamiento que son capaces de aplicar todas ellas la misma instrucción sobre datos diferentes, explotando así el paralelismo a nivel de datos.

Esta filosofía fue la base para el desarrollo de los computadores vectoriales a principio de los 70, que fueron especialmente populares en esos años, siendo el más conocido el computador Cray. Las unidades de procesamiento gráfico GPU siguen este esquema, ya que cuentan con numerosas unidades de procesamiento que ejecutan la misma instrucción generando tantas salidas como datos de entrada. Sin embargo, las nuevas generaciones Turing y Ampere de la marca de GPUs Nvidia ya permite tener un flujo diferente de instrucciones para diferentes grupos de *cores* con sus *cores* de trazado de rayos (RT), por lo que también permiten el esquema MIMD.

- *Multiple Instructions Single Data (MISD)*: Esta arquitectura teórica se basa en la aplicación de diferentes instrucciones por parte de múltiples unidades de procesamiento sobre el mismo dato. En la práctica no existe ninguna arquitectura que se adapte totalmente a este modelo, aunque se podría decir que las basadas en sistemas de detección de fallos se aproxima, ya que se ejecutan distintas instrucciones sobre los mismos datos para comprobar que el estado no cambia.

- *Multiple Instructions Multiple Data (MIMD)*: Los sistemas MIMD se basan en que un conjunto de unidades de procesamiento o núcleos independientes apliquen diferentes instrucciones sobre múltiples datos. Estos sistemas son los más utilizados en la actualidad y son los principales sistemas paralelos.

  Dentro de esta clasificación se podría hacer la distinción entre multiprocesadores y multicomputadores, o lo que es lo mismo, máquinas de memoria compartida o máquinas de memoria distribuida, cuya diferencia se explica a continuación.

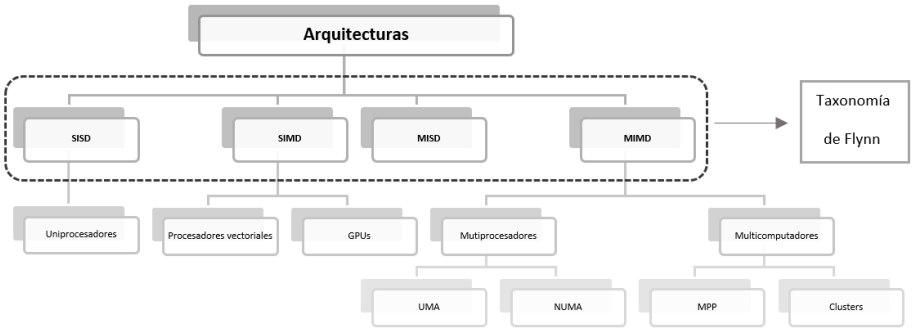


Figura 3.1: Clasificación de las diferentes arquitecturas de computadores según la taxonomía de Flynn.

3.1.1 Computación Paralela en Memoria Compartida

Los sistemas paralelos de memoria compartida pueden ser de tipo SIMD (como las GPUs), o de tipo MIMD, categoría en la cual entran la mayoría de computadores multiprocesador o *multicore* actuales. En la Figura 3.2 se puede apreciar la estructura de este tipo de sistemas. Como se observa, son sistemas de uno o varios procesadores *multicore* que tienen espacio de memoria compartido.

El tipo de acceso a memoria puede variar, teniendo por una parte la variante UMA (Uniform Memory Access), o acceso uniforme a memoria, en la cual

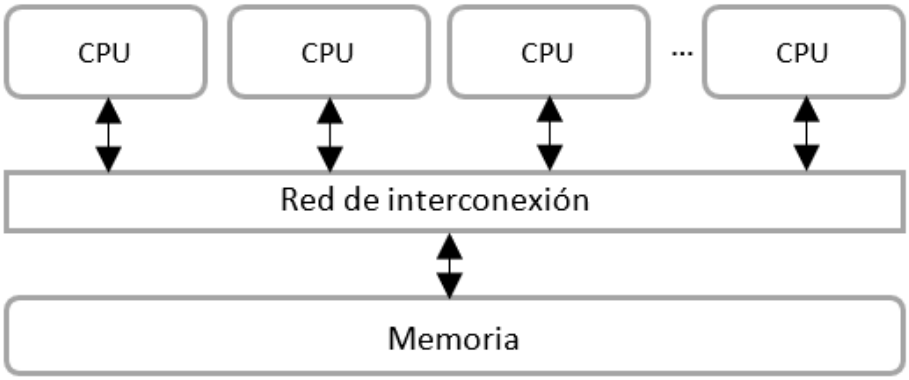


Figura 3.2: Esquema de sistema paralelo con memoria compartida.

todos los procesadores son idénticos. Este tipo de acceso es más sencillo de cara al programador, ya que el espacio de memoria de todos los procesadores es compartido y coherente (los cambios de cualquiera de los procesadores en la memoria serán visibles para el resto).

Por otra parte está la variante NUMA (Non-Uniform Memory Access) o acceso a memoria no uniforme, en la cual cada conjunto de procesadores que se encuentre en el mismo chip tiene su propia memoria local, a la que se accede más rápido que en la variante NUMA. Los distintos chips están interconectados para que cualquier procesador pueda acceder a la memoria local de los otros. En este modelo, la gestión de las direcciones de memoria es más complejo, pero compensa por su menor latencia.

Hoy en día, la mayoría de computadores cuentan con múltiples procesadores que a su vez cuentan con múltiples *cores* o núcleos. Al haber múltiples unidades de procesamiento, los procesos se pueden repartir entre ellas, logrando así realizarlas de forma paralela y por tanto aumentar la eficiencia lograda y reducir el tiempo de cómputo.

La programación paralela en memoria compartida se basa en disponer de diferentes hilos que realicen tareas diferentes trabajando sobre los datos en memoria a los cuales tienen acceso todos ellos. Por tanto, es indispensable la sincronización y control de acceso a las variables para garantizar el correcto funcionamiento.

El modelo de programación más extendido para estos sistemas es OpenMP. Esta API se basa en la utilización de directivas de compilador para indicar de qué manera se debe paralelizar una sección de código, sin tener que manejar explícitamente los hilos. Este modelo sigue un sistema *fork-join*, en el cual existe un hilo principal que ejecuta las secciones de código secuenciales, y un grupo de hilos paralelos que se activarán al entrar en una región paralela y se desactivarán cuando ésta finalice, momento en el que se realiza una sincronización, recopilando entonces el hilo principal los resultados obtenidos por el resto, de la forma que es especifique.

OpenMP tiene herramientas para gestionar el alcance de las variables, o lo que es lo mismo, qué hilos tienen acceso a dichas variables, ya que será esencial gestionarlo correctamente para que el programa paralelo sea correcto evitando condiciones de carrera.

Esta API, con implementaciones para C, C++ y Fortran proporciona al usuario un nivel de abstracción en el manejo de hilos, lo que hace más accesible la programación según el modelo de memoria compartida.

3.1.2 Computación Paralela en Memoria Distribuida

Los sistemas con memoria distribuida se basan en combinar la potencia de cómputo de sistemas independientes conectados mediante una red. Los sistemas más populares que siguen esta estructura son los denominados *clusters*, conjuntos de computadores generalmente homogéneos, es decir, que tienen las mismas características, que a los ojos del usuario actúan como un único sistema. Cada uno de estos computadores se denominan nodos. En la Figura 3.3 se observa el esquema que siguen los sistemas de memoria distribuida.

Para que sea eficiente la programación paralela en estos sistemas es esencial contar con una red de interconexión de muy baja latencia. Ya que el sistema está compuesto por computadores independientes, no comparten espacio de memoria, por lo que si requiere comunicación entre procesos paralelos será necesario el envío y recepción de mensajes a través de la red. Dicha red puede ser de tipo Ethernet, pero en sistemas de alto rendimiento es preferible una red Infiniband, que tiene la latencia más baja actualmente. De este modo se evita en la medida de lo posible que la red se convierta en un cuello de botella debido a la comunicación necesaria, disminuyendo así la escalabilidad de los programas paralelos.

Cabe mencionar que actualmente la mayoría de computadores son multiprocesador y *multicore*, por lo que los *clusters* se podrían clasificar también co-

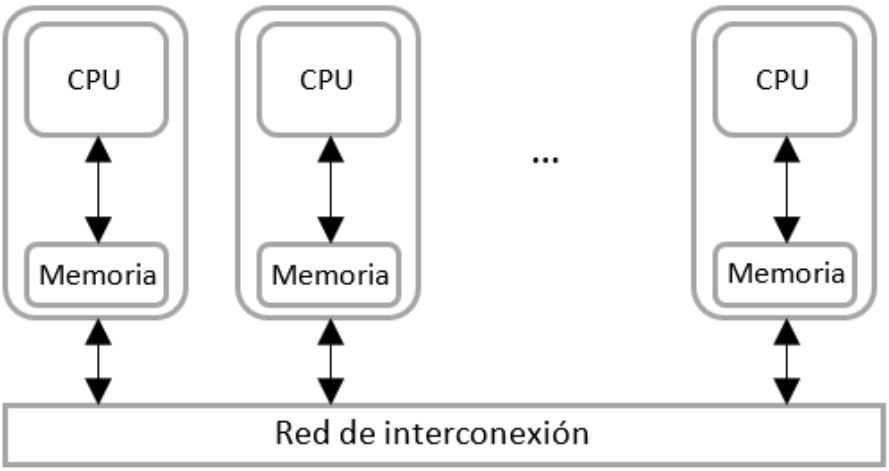


Figura 3.3: Esquema de sistema paralelo con memoria distribuida.

mo sistemas híbridos de memoria compartida en un único nodo y memoria distribuida en el conjunto. Este hecho se puede explotar para combinar el paralelismo a los dos niveles y aumentar la eficiencia.

La programación en estos sistemas se realiza mediante un modelo de paso de mensajes. Actualmente, MPI (*Message Passing Interface*) es el estándar más utilizado de este modelo. MPI proporciona un conjunto de funciones que extienden la funcionalidad de un lenguaje (C, C++ o Fortran) para gestionar el paralelismo y el paso de mensajes. En este modelo, cada proceso ejecuta una copia del mismo programa ejecutable, y será necesario gestionar la distribución de los datos que necesita cada uno de ellos ya que en el inicio únicamente tendrá acceso a los datos el proceso principal.

Esto se puede realizar mediante comunicaciones punto a punto (un proceso envía un mensaje a otro proceso que contiene los datos), o bien mediante comunicaciones colectivas, en las que se hace un reparto o difusión de los datos a todo el conjunto de procesos sin tener que especificar el envío a cada uno de ellos. Es imprescindible gestionar adecuadamente las comunicaciones para que el coste temporal del programa no aumente excesivamente debido a éstas. Para ello existen primitivas de comunicación no bloqueantes que permiten evitar tiempos de espera de un proceso que realiza un envío cuando otro proceso no puede realizar la recepción de un mensaje por estar ocupado con otra tarea.

La distribución de tareas en MPI la puede realizar el proceso principal mediante un esquema maestro-esclavo, en el cual se van asignando tareas a medida que un proceso termina las pendientes, o bien se puede hacer un reparto inicial, lo cual se adapta muy bien cuando se trata con matrices o vectores puesto que se pueden realizar los cálculos por bloques.

3.1.3 Computación Distribuida en Sistemas Heterogéneos

Fuera de los sistemas de computación paralela locales existen los sistemas de computación distribuida. A diferencia de los anteriores, son sistemas cuyos recursos no tienen que estar situados en el mismo lugar, sino que se pueden encontrar en cualquier parte del mundo y están conectados entre ellos mediante internet. Esta organización se denomina Grid, con una topología de interconexión en malla. Los sistemas Grid se basan en la integración y compartición de recursos como computadores de alto rendimiento, sistemas de almacenamiento, bases de datos y otros, que son administrados por entidades diferentes. Por tanto, son sistemas altamente heterogéneos, con características diferentes.

A través de un software *middleware* se consigue una abstracción de estas particularidades de cada sistema. A ojos del usuario, el Grid es un gran supercomputador en el cual se pueden solicitar los recursos a usar según necesidad. Estos sistemas son de gran utilidad para la comunidad científica, puesto que la potencia de cómputo es alta sin la necesidad de invertir en equipos propios. Además, debido a la heterogeneidad de los sistemas es altamente probable que sea sencillo cumplir las necesidades específicas de los usuarios.

Por otra parte, el inconveniente de la computación en sistemas distribuidos es la velocidad de la red. A diferencia de los *clusters*, donde se puede emplear un modelo de paso de mensajes con eficiencia, en los sistemas Grid el tiempo de latencia aumenta. Por ello, están pensados para programas paralelos que no necesiten comunicación durante la ejecución, sino que sean tareas independientes que se puedan ejecutar en recursos diferentes y combinar después sus resultados, es decir, paralelismo de grano grueso.

3.2 Herramientas Hardware

En esta sección se listan los entornos y herramientas hardware que se han empleado para la implementación y validación de los métodos desarrollados en esta tesis.

3.2.1 Entorno Grid

La infraestructura Grid utilizada para el estudio multiparamétrico del método LSQR ha sido European Grid Infraestructure Egi.eu 2021. Dicha infraestructura está financiada con fondos públicos para dar a la comunidad científica acceso a más de 1,000,000 *cores*, 500 PB de almacenamiento en disco y *online*, y 19 proveedores de *cloud* federados para impulsar la investigación y la innovación en Europa. Este Grid cuenta con centros de recursos ubicados por toda Europa, la región de Asia y el Pacífico, Canadá y América Latina.

Para hacer uso de EGI, se ha empleado la Organización Virtual VO Biomed, una VO de gran tamaño a escala global, centrada en los ámbitos de Biomedicina, Biotecnología, Medicina, Biología y Ciencias Computacionales, que son las áreas en las que encuadra perfectamente este proyecto.

En este entorno se ha desarrollado el estudio multiparamétrico del método LSQR que se presenta en el capítulo 4, mostrando la utilidad de este tipo de plataforma para exploración de parámetros de un método, así como las

dificultades que puede conllevar y planteando estrategias sobre cómo resolver las dependencias de software.

3.2.2 Cluster

El *cluster* que se ha empleado principalmente ha sido Rigel, perteneciente al centro de cálculo de la Universitat Politècnica de València. Este *cluster* fue seleccionado por ser el que más cantidad de memoria RAM por nodo ofrecía de entre todos los disponibles. Además de contar con un *cluster* paralelo general, y otro para el uso de GPUs, Rigel tiene un *cluster* de sistemas de memoria compartida para programas muy intensivos en consumo de memoria. Está compuesto por 4 servidores RX500S7 con cuatro procesadores Intel Xeon E5-4620 de 8 núcleos (32 núcleos por nodo). Las características de cada nodo son:

- Cuatro procesadores Intel Xeon E5-4620 8c/16T.
- Memoria RAM: 256GB DDR3 (ratio 8GB/*core*).
- 2 x interfaces 10GbE.
- 2 x Interfaces GbE.
- 2 x FC 8Gb/seg.

El procesador E5-4620 proporciona buen compromiso entre eficiencia y consumo, el *cluster* alcanza una potencia de 2.1TeraFlops según el test LINPACK.

Los trabajos en los que se ha empleado este *cluster* son los que se presentan en los capítulos 6, 7 y 8, mostrando como la cantidad de memoria RAM que proporciona sigue siendo insuficiente cuando aumenta el tamaño de problema.

3.2.3 Equipo multicore

Las pruebas realizadas para equipos *multicore* se han llevado a cabo en una estación de trabajo llamada Anka. Esta máquina cuenta con las siguientes características:

- Procesador Intel i7-7800X.
- 6 *cores* físicos y 12 hilos.
- 128 GiB de memoria RAM DDR4.

- Frecuencia de reloj: 3.5 GHz.
- Frecuencia de reloj turbo: 4.0 Ghz.

Pese a ser una estación de trabajo, se ha conseguido resolver en ella la reconstrucción mediante la factorización QR para alta resolución, con objetivo de demostrar que es un método que no necesita de hardware de última generación ya que no exige muchos recursos.

El principal trabajo desarrollado en este equipo ha sido el presentado en el capítulo 9, donde se muestra cómo es posible resolver un problema de grandes dimensiones en un equipo modesto como este.

3.2.4 Discos HDD y SSD

Para el método QR Out-Of-Core es esencial la velocidad del disco puesto que se escriben y leen bloques a medida que se necesitan. Por ello, se ha realizado una comparativa empleando dos discos diferentes, uno tradicional HDD, y otro de estado sólido SSD, que son los siguientes:

- Disco HDD: Toshiba DT01ACA200 (velocidad de lectura: 190 MB/s)
- Disco SSD: Samsung SSD 970 EVO 2TB con conector M.2 contectado a PCI mediante adaptador (velocidad de lectura: 2400 MB/s)

La principal diferencia entre ambos tipos es que los HDD emplean discos rígidos en los que un cabezal lee o escribe los datos, mientras que los SSD son más actuales y emplean memorias NAND flash interconectadas entre sí, con un controlador integrado que se encarga de gestionar la lectura y escritura así como la caché y el borrado de datos del disco.

El uso y comparativa de estos dos discos se ha llevado a cabo en el capítulo 9, analizando la eficiencia del método QR Out-Of-Core según el disco empleado.

3.3 Herramientas Software

A continuación se describen todas las herramientas software que han sido utilizadas directa o indirectamente durante el desarrollo de esta tesis, ya sean librerías, lenguajes o entornos de programación.

3.3.1 Matlab

Una plataforma muy utilizada para cálculo matricial es Matlab (The MathWorks, Inc. 2021a), cuyo nombre es una abreviación de *Matrix Laboratory*, lo cual indica claramente la idea principal de este entorno. Matlab es un lenguaje de programación de alto nivel orientado a la computación científica y de altas prestaciones, que proporciona una gran cantidad de funciones para manipulación de matrices, cálculo numérico y simbólico, análisis y visualización de datos y un largo etcétera, ampliando cada vez más su lista de utilidades a través de *toolboxes*.

Además de lenguaje de programación, Matlab proporciona un entorno de desarrollo integrado IDE que facilita el desarrollo de algoritmos, interfaces gráficas, y sobretodo visualización de los datos para que sea amigable y accesible a todo tipo de usuarios, desde principiantes hasta programadores más expertos. Por otra parte, pese a ser un lenguaje de alto nivel, está muy optimizado, utilizando rutinas de LAPACK para el álgebra lineal y la computación matricial, lo que hace que las rutinas sean rápidas y robustas.

Por último, a través de su *toolbox* de computación paralela proporciona herramientas para cómputo paralelo en procesadores *multicore*, GPUs o sistemas distribuidos como *clusters* y *clouds*. Con todo ello Matlab es una potente herramienta que es especialmente útil en el desarrollo y testeo de nuevos algoritmos como paso previo a una implementación más optimizada.

Matlab ha sido empleado en el capítulo 4 mediante su Runtime para realizar un estudio en una plataforma Grid. Además, en los capítulos 6, 7 y 8, ya que se emplea la versión para Matlab de la librería SuiteSparse, que se definirá a continuación.

3.3.2 Docker y udocker

Docker (Merkel 2014) es una API de alto nivel que proporciona herramientas para la creación y despliegue de contenedores de software. Mediante esta herramienta se pueden virtualizar aplicaciones sin tener que virtualizar al completo sistemas operativos, ya que a diferencia de las máquinas virtuales tradicionales, emplean los recursos del sistema operativo base de la máquina, ya que comparte el mismo Kernel.

De esta manera se pueden empaquetar y aislar aplicaciones de forma ligera para su uso en diferentes plataformas, lo que resulta de gran utilidad para poder desarrollar herramientas multiplataforma sin tener que modificar el código.

Las imágenes creadas que contienen todo lo necesario para la ejecución de una aplicación pueden ser registradas en *Docker Hub*, un repositorio de imágenes, para posteriormente descargarla de sus servidores y desplegarla en la máquina a instalar. Sin embargo, puede surgir un problema al emplear Docker en máquinas no propias, ya que su instalación requiere de privilegios de superusuario.

Esto supone una gran desventaja en el ámbito de la computación científica, donde se hace uso de múltiples *clusters*, supercomputadores, servidores y sistemas Grid en los que habitualmente no se tienen privilegios y no siempre es posible solicitar la instalación de nuevo software.

Como solución a dicho problema nace udocker (Gomes y col. 2018), una herramienta simple que permite la descarga y ejecución de contenedores Docker en espacio de usuario sin necesidad de instalación ni privilegios. Con el uso de udocker es posible la ejecución de contenedores en cualquier sistema aunque no sea propio, lo cual es muy útil para investigadores. A pesar de su gran utilidad, cuenta con la desventaja de no poder ejecutar ningún proceso que requiera de privilegios, lo que reduce la funcionalidad de los contenedores y puede ser un factor importante a la hora de escoger la plataforma a emplear para ciertos estudios.

Docker y udocker se han empleado para la realización del estudio multiparamétrico que se presenta en el capítulo 4, donde ha sido analizado su uso en una plataforma Grid.

3.3.3 BLAS

Basic Lineal Algebra Subprograms (BLAS) es una especificación que describe un conjunto de rutinas de bajo nivel para operaciones de álgebra lineal densa desarrollada por *netlib* (Blackford y col. 2002). Entre las operaciones proporcionadas están la suma y multiplicación de vectores, combinaciones lineales y productos de matrices, entre otras. BLAS se ha convertido en la especificación de de rutinas de álgebra lineal por excelencia, formando una base que permite personalización para lograr mejorar el rendimiento según el hardware utilizado.

De hecho, existen variedad de implementaciones de BLAS para distintas arquitecturas, centradas en explotar las características de éstas para lograr un mayor aprovechamiento. Ejemplo de ello es la librería de Intel MKL (Wang y col. 2014), desarrollada para arquitecturas x86 y x86-64 con especial optimización para procesadores Intel, o BLIS (Van Zee y Van De Geijn 2015), optimizada para AMD. Además, existen otras optimizaciones de BLAS más generales desarrolladas por investigadores como OpenBLAS (Xianyi, Qian y Saar 2021),

optimizada para las arquitecturas más usadas, o ATLAS (Whaley 2011), que se optimiza de manera automática al instalarla en cierta arquitectura.

Por todo ello, el uso de BLAS aporta por una parte eficiencia, ya que hay opciones optimizadas para cada hardware, y por otra robustez por todos sus años de desarrollo y amplio uso, además de portabilidad entre distintas arquitecturas.

La funcionalidad de BLAS se divide en tres niveles, según la complejidad de sus algoritmos, además de ser el orden en el que se incluyeron en la librería:

- Nivel 1: Operaciones sobre vectores, con coste de orden lineal, como producto escalar, producto vectorial, norma vectorial, o suma de vectores (Lawson y col. 1979).
- Nivel 2: Operaciones matriz-vector, con coste de orden cuadrático (Dongarra y col. 1988). La operación principal de este nivel es la multiplicación matriz-vector con matrices en diferentes formatos (triangulares, simétricas, banda, para tipos enteros, reales, complejos, entre otros).
- Nivel 3: Operaciones matriz-matriz, con coste de orden cúbico (Dongarra y col. 1990). La operación base de este nivel es la multiplicación entre matrices, pudiendo ser generales, Hermitianas, simétricas, etc. Esta operación sobre matrices generales es conocida como General Matrix-Matrix Multiplication (GEMM), y es la base de muchos algoritmos de cálculo numérico, por lo que optimizarla puede lograr un aumento de rendimiento considerable en los programas que implementen dichos algoritmos.

Las rutinas de BLAS3, por ratio entre cantidad de datos y operaciones a ejecutar son las que más eficiencia aportan en implementaciones paralelas. Las versiones multihilo de BLAS, como ATLAS o la mayoría de las desarrolladas por fabricantes, implementan paralelismo sobre operaciones escalares, o bien sobre operaciones por bloques.

Las rutinas de BLAS se han empleado de manera indirecta a través de las librerías Intel's MKL y SuiteSparse, que se explican a continuación. En los capítulos 6, 7, 8 y 9 se ha empleado una versión multihilo de BLAS para lograr paralelismo en ordenadores *multicore*.

3.3.4 LAPACK

Linear Algebra PACKage (LAPACK) es una librería de software para la resolución de problemas de álgebra lineal (Anderson y col. 1992). Nace originalmente como la combinación de la funcionalidad que proporcionaban EISPACK, dedicada a valores propios, y LINPACK, para sistemas de ecuaciones lineales, convirtiéndose en la librería estándar.

Incluye rutinas dedicadas a la resolución de sistemas de ecuaciones lineales, problemas de mínimos cuadrados, valores propios y valores singulares, además de distintas factorizaciones de matrices como pueden ser la LU, Cholesky o QR.

LAPACK utiliza llamadas a BLAS para hacer las operaciones elementales, siguiendo la misma filosofía y notación. Como BLAS dispone de implementaciones específicas que mejoran su rendimiento según la arquitectura, y basa su paralelismo en memoria compartida en el propio paralelismo de BLAS. Por otra parte, cuenta con una versión para paralelismo en memoria distribuida llamada ScaLAPACK.

LAPACK, al igual que BLAS, se ha empleado de manera indirecta a través de SuiteSparse y MKL, en los trabajos que se presentan en los capítulos 6, 7, 8 y 9 que emplean la factorización QR.

3.3.5 SuiteSparse

SuiteSparse (Davis, Amestoy y Duff 2021) es una colección de algoritmos orientados a matrices dispersas, entre los que se encuentran factorizaciones de matrices como LU, Cholesky y QR, así como herramientas para el cálculo y particionado de grafos y distintas utilidades como el cálculo del rango de una matriz o multiplicación dispersa de matrices. Esta librería está implementada en C++, y puede ser integrada dentro de Matlab, así como en programas escritos en C o C++. Actualmente también está implementada para el uso de GPU a través de Nvida CUDA.

Para el desarrollo de esta tesis se ha empleado el paquete SuiteSparseQR (Davis 2011), una implementación de la factorización QR dispersa basado en el método multifrontal. Esta implementación obtiene gran rendimiento el entornos *multicore* gracias al uso de funciones de LAPACK y BLAS multihilo, lo que permite acelerar los procesos. Las factorizaciones multifrontales se basan en dividir la matriz en pequeñas submatrices que se pueden tratar como matrices

densas y factorizar de manera paralela. La factorización está guiada por un árbol que representa la dependencia entre las factorizaciones parciales.

Esta implementación de la factorización QR es especialmente recomendable para su uso desde Matlab, ya que tiene una mejor administración de la memoria y minimización del relleno que la factorización nativa de Matlab, y mejor eficiencia, lo que hace que sea un paquete muy competitivo e ideal para la fase de testeo de algoritmos que empleen esta factorización.

Esta librería se ha empleado en los trabajos presentados en los capítulos 6, 7 y 8 para la aplicación de la factorización QR, de la cual se ha analizado su eficiencia.

3.3.6 PETSc

Portable, Extensible Toolkit for Scientific Computation (PETSc) es una librería de código abierto desarrollada para lenguaje C, C++, Fortran y Python, así como la posibilidad de integrarse con Matlab (Balay y col. 2021). Proporciona un conjunto de rutinas y estructuras de datos para el desarrollo de aplicaciones científicas de forma escalable.

PETSc hace uso de rutinas de BLAS y LAPACK, así como del estándar MPI de paso de mensajes para implementar el paralelismo de memoria distribuida, y es una de las librerías paralelas de cálculo numérico más usadas, con aplicaciones en campos como la biomedicina, geotecnia, dinámica de fluidos, entre muchos otros campos científicos.

Esta librería está diseñada por módulos y tiene herramientas de bajo nivel como puede ser el manejo de vectores y matrices, como herramientas de más alto nivel como precondicionadores, métodos de Krylov, métodos no lineales o de optimización. Además, cuenta con utilidades de *profiling* que proporcionan información sobre las operaciones de coma flotante, el rendimiento paralelo y el uso de memoria de un programa.

Con todo ello, PETSc conforma una potente librería para paralelismo en memoria distribuida de gran utilidad para investigadores.

3.3.7 SLEPc

La librería Scalable Library for Eigenvalue Problem Computations (SLEPc) nace como una extensión de PETSc, como software dedicado al cálculo paralelo de valores y vectores propios de matrices dispersas de gran escala. Desarrollada en la Universitat Politècnica de València (Hernandez, Roman y Vidal 2005; Roman y col. 2015), esta librería hace uso de las estructuras de datos definidas en PETSc para la manipulación de vectores y matrices, así como del estándar MPI para implementar paralelismo en memoria distribuida.

SLEPc proporciona funciones para el cálculo de valores propios y singulares así como la resolución de problemas basados en la descomposición en valores singulares SVD, o el problema polinomial de valores propios. Además, incluye *solvers* para valores propios no lineales y transformaciones espectrales.

Al estar integrado con PETSc permite combinar el uso de ambas librerías sin cambiar de paradigma ni tipo de estructuras, lo que hace que sea todavía más interesante su uso ya que aumentan las posibilidades de cálculos a aplicar. Es por ello que también es ampliamente utilizada en la computación científica, teniendo diversos campos de aplicación como ciencias nucleares, ciencias de materiales, recuperación de información entre muchas otras. Además, es utilizada en centros de computación de prestigio como el *Barcelona Supercomputing Center*, lo que indica su calidad.

En el capítulo 6 se ha empleado SLEPc (y por tanto, PETSc), para realizar la reconstrucción de imagen mediante la descomposición en valores singulares, utilizando el *solver* proporcionado por la librería.

3.3.8 Intel MKL

Intel's Math Kernel Library (MKL) es una librería desarrollada por Intel (Wang y col. 2014) para aplicaciones relacionadas con el álgebra numérica, en campos de matemáticas, ingeniería, finanzas, etc. Las funciones de MKL están altamente optimizadas para sistemas Intel de altas prestaciones, como pueden ser sistemas *multicore* o *manycore*. Además, también puede ser utilizada en distintas arquitecturas como AMD, aunque con más bajo rendimiento.

Esta librería incluye funciones de álgebra lineal a través de otras como BLAS, LAPACK, Sparse BLAS, ScaLAPACK, entre otras. Además, proporciona un conjunto de transformadas de Fourier, optimización de sistemas no lineales, operaciones vectoriales, estadísticas, entre otras. Además, muchas de las ruti-

nas de la librería tienen paralelismo implícito, lo que evita que sea el programador quien tenga que paralelizar el código para optimizarlo.

Dado que todas sus funciones están especialmente optimizadas para arquitecturas Intel, que son las más extendidas, se ha convertido en la librería de álgebra más utilizada, por su alto rendimiento y gran colección de operaciones. Además, está disponible para sistemas operativos Windows, Linux y MAC, y se puede usar en programas desarrollados en lenguaje C, C++ o Fortran.

La librería MKL ha sido empleada en el capítulo 9 para la implementación del método QR.

3.3.9 libflame

Esta librería está dedicada al álgebra lineal con matrices densas de altas prestaciones con APIs para Matlab y C, aunque fácilmente adaptable a otros lenguajes mediante interfaces, y se ha desarrollado siguiendo la metodología Formal Linear Algebra Method Environment (FLAME) (Bientinesi, Quintana-Ortí y Geijn 2005; Gunnels y col. 2001) Dicha metodología se basa en emplear una notación basada en bucles que se asemeja a la forma en la que se expresan los algoritmos mediante imágenes.

En libflame (Zee 2012; Zee y col. 2009) la programación se hace por objetivos, identificando el objetivo final al que se quiere llegar, así como el estado antes de entrar al bucle que deberá cumplirse antes y después de cada iteración y conforma la invariante del bucle. Posteriormente, mediante predicados se van realizando las operaciones del propio algoritmo, así como la prueba de corrección, ya que los propios predicados garantizan la correctitud de los cálculos, al verificar que se cumplen las condiciones.

El modelo de programación de esta librería está basado en objetos por bloques, una abstracción de alto nivel que evita el manejo de índices cuando se trata con matrices y vectores, consiguiendo de esta manera simplificar el uso de éstos.

Además del paralelismo en memoria compartida que se consigue empleando rutinas de BLAS multihilo, como hace LAPACK, libflame consigue paralelismo de mayor nivel con un *runtime* propio llamado *SuperMatrix*. Dicho *runtime* es encargado de analizar en tiempo de ejecución las dependencias de datos entre los bloques del algoritmo, planificando la ejecución de las tareas en diferentes hilos según esas dependencias. De esta manera se consigue una mayor tasa de tareas paralelas, superando en ocasiones a LAPACK (Zee y col. 2009) pese a ser una librería con muchos más años de desarrollo.

Esta librería se ha empleado para el desarrollo del método QR Out-Of-Core mediante un algoritmo a bloques que se presenta en el capítulo 9.

Capítulo 4

Estudio Multiparamétrico del método LSQR en Grid

En este capítulo se presenta el artículo (Chillarón y col. 2017), publicado en *Procedia Computer Science* en el volumen correspondiente a los *Proceedings* del congreso *The International Conference on Computational Science (ICCS)* celebrado en 2017.

En este artículo se presenta el estudio multiparamétrico llevado a cabo en un entorno Grid para el método LSQR con regularización STF y aceleración FISTA. Dicho método contiene parámetros que pueden influir en la convergencia y en la calidad final de las reconstrucciones, por lo que el objetivo de este estudio es realizar una exploración completa de todas las posibles combinaciones para encontrar la configuración óptima del método, maximizando la calidad de las imágenes obtenidas.

Para llevar a cabo esta exploración de los parámetros se ha utilizado un paralelismo de grano grueso, lanzando un trabajo por cada posible combinación de parámetros en un entorno de computación Grid. En este tipo de entornos la disponibilidad de software no estándar es baja, ya que está compuesto de muchas organizaciones diferentes que aportan sus recursos y no tienen un ad-

ministrador conjunto, por lo que es complicado gestionar los sistemas de modo que se pueda unificar el software disponible en todos ellos e incorporar nuevo.

Además, si se trata de software que requiere de licencia se añade un grado de complejidad. Este ha sido el caso del trabajo desarrollado, ya que el código había sido implementado en Matlab. Cabe mencionar que se planteó el uso de otras librerías de código abierto como PETSc, implementada en C, por lo que es más portable. Sin embargo, debido a la heterogeneidad de los recursos de cómputo del Grid, y por ser un lenguaje compilado, habría que generar un ejecutable diferente para cada plataforma, considerando sistema operativo y arquitectura, lo que se convierte en un trabajo tedioso.

Por tanto, la opción escogida finalmente fue emplear el compilador propio de Matlab para generar un ejecutable binario del método. Dicho ejecutable podrá ser utilizado en máquinas que no cuenten necesariamente con una instalación de Matlab completa, sino con su Runtime, más ligero y rápido de instalar. De esta manera, el problema de portabilidad está resuelto, pero es necesario hacer una instalación del Runtime en tiempo de ejecución en cada elemento de cómputo.

Para lograr este objetivo se plantean dos vías distintas: el uso de los elementos de almacenamiento (SE) del Grid para disponer del instalador de Matlab Runtime, que se descarga e instala con el propio trabajo a ejecutar, o el uso de contenedores Docker que lleven el Runtime instalado y únicamente sea necesario descargarlos y desplegarlos para ejecutar los programas. El hecho de usar Docker en un sistema Grid cuenta con el inconveniente de la falta de privilegios, lo que ha sido solventado con la herramienta udocker.

Además, puesto que los datos de entrada para la reconstrucción son pesados, será necesario que estén almacenados en los elementos de almacenamiento para que puedan ser descargados en tiempo de ejecución de los trabajos.

En el artículo, que se presenta a continuación, se estudia el rendimiento de los elementos de almacenamiento del Grid, analizando el número de trabajos fallidos así como el tiempo medio de descarga según el número de veces que se hayan replicado los datos. Por otra parte, se hace un análisis de la eficiencia de utilizar los SE o bien contenedores Docker, explicando el flujo de tareas para el lanzamiento de cada trabajo en los dos casos. Además, se analizan los resultados obtenidos para el estudio multiparamétrico, con el cual se han encontrado las configuraciones óptimas para el método LSQR+STF+FISTA.

En cuanto al rendimiento de los elementos de almacenamiento del propio Grid, se puede decir que realizar réplicas de los datos es esencial para un estudio con

descargas masivas como es el caso. Estudiando las estadísticas de 100 trabajos encargados de descargar la matriz del sistema más pesada (para resolución 512×512), se ha determinado que el número de réplicas de un dato en diferente nodos de almacenamiento influye directamente tanto en el tiempo medio de descarga como en el número de trabajos fallidos. En este ejemplo, con una sola réplica de la matriz 28 de los 100 trabajos fallan, y el tiempo de descarga es de una media de 1934 segundos. Sin embargo, replicando la matriz 6 veces el número de trabajos fallidos se reduce a 6, y el tiempo medio de descarga a 470 segundos.

Por otra parte se ha estudiado el número de trabajos que pueden ser enviados por hora, lo cual no es trivial en estudios con un número elevado de trabajos. En este caso, se han realizado 16,800 ejecuciones que se han dividido en 4200 trabajos para compensar el tiempo de descarga de los datos con tiempos de ejecución más largos. El lanzamiento de estos 4200 trabajos se ha dividido en cinco (para diferentes resoluciones), y se ha iniciado el lanzamiento en cinco hilos diferentes de 840 trabajos. Los resultados obtenidos muestran que se pueden llegar a lanzar alrededor de 1400 trabajos por hora, pero esta cifra disminuye ligeramente si el lanzamiento es continuado por varias horas. Esto podría deberse a una saturación del sistema de gestión de trabajo debido al alto número de trabajos que se están manejando al mismo tiempo.

En cuanto a la comparativa de la instalación de Matlab Runtime localmente y el uso de contenedores con el Runtime ya instalado, los resultados obtenidos son ligeramente peores cuando se emplean los contenedores. El tiempo de espera crece según aumenta el tamaño del problema a resolver, lo cual es únicamente dependiente de la plataforma Grid. El tiempo de descarga es más alto en los elementos de almacenamiento, puesto que los servidores del repositorio de imágenes *Docker Hub* son más eficientes. Sin embargo, el tiempo de ejecución de los métodos es notablemente más alto cuando se emplean contenedores. Esto podría ser debido a que en este caso no se hace uso de la totalidad de los recursos de un elemento de cómputo como en los trabajos que se ejecutan con normalidad. La diferencia del tiempo de ejecución disminuye al aumentar el tamaño del problema.

Por último, en este estudio se ha hecho una exploración de los parámetros que influyen en la calidad y velocidad de convergencia, que son: resolución de la imagen a resolver, número de vistas, número de iteraciones internas del LSQR, aplicación o no de STF, aplicación o no de FISTA, y parámetro alfa de STF si aplica. Para cada resolución se ha obtenido la configuración óptima, priorizando la calidad de imagen por encima de la eficiencia temporal. De este modo, se han logrado reconstrucciones de alta calidad empleando 30 vistas para un

fantoma matemático, independientemente de la resolución de la imagen. Además se ha observado que el número de iteraciones necesaria para que el método converja, así como si aplicar o no STF/FISTA y cada cuantas iteraciones, es dependiente de la información que contiene la matriz del sistema.

Este tipo de estudio en sistemas masivamente paralelos es fácilmente reproducible, y puede ser una herramienta muy útil en la exploración de nuevos métodos al ahorrar muchas horas de cómputo haciendo uso de paralelismo de grano grueso para ejecutar miles de tareas en recursos distintos.

Procedia Computer Science

Volume 108, Pages 1195-1204, 2017

Combining Grid Computing and Docker Containers for the Study and Parametrization of CT Image Reconstruction Methods

Mónica Chillarón[1], Vicente Vidal[1], Damián Segrelles[2], Ignacio Blanquer[2] and Gumersindo Verdú[3]

1 Departamento de Sistemas Informáticos y Computación (DSIC), Universitat Politècnica de València, Valencia, Spain.
2 Instituto de Instrumentación para Imagen Molecular (I3M), Universitat Politècnica de València, Valencia, Spain.
3 Instituto de Seguridad Industrial, Radiofísica y Medioambiental (ISIRYM), Universitat Politècnica de València, Valencia, Spain.

Abstract

Computed tomography (CT) is one of the most widely used methods in Medical Imaging. Despite of its relevance in the diagnosis of diseases with a high impact in our society (such as cancer), it is one of the most potentially harmful modalities. CT requires a high X-ray dose to be induced to the patients. Solving the CT Image Reconstruction problem iteratively in order to approximate the solution allows working with only a subset of the input data required by direct methods. This directly implies a reduction of the radiation received by the patient and a strong reduction on the potential morbidity. Therefore, we aim to study the feasibility of such methods for their actual application, with the purpose of concluding if they are accurate and can obtain good quality images with a lower dose of X rays. This paper discusses the use of containers within a Grid Computing platform to perform a thorough study of all the possible configurations and parameters of various methods being developed to reconstruct CT images iteratively, which could lead to find the optimal configuration of the parameters. The work compares two approaches for managing the software dependencies of the code: store the software libraries on a Storage Element and using containers for executing the job.

Keywords: CT, Medical Imaging, Reconstruction, Grid, Containers.

4.1 Introduction and Background

Nowadays, the importance of computing science applied in medicine is clear. Physicians and medical staff are daily assisted by computer systems for diagnosis and therapy, enabling a personalized care of patients. Computing has allowed the development of new medical diagnosis methods based on the principle of capturing images (i.e Computerized Tomography - CT), from a human body and apply on them a computational process to observe and diagnose abnormalities such as malignant tumours or bone lesions.
In this paper, we focus on the computational process of CT image reconstruction (Radon 1986), which reconstructs a three-dimensional image through a set of X-ray projections of a given body part, captured from different angles. The method calculates the attenuation suffered by the X-ray beams passing through the elements (i.e bone, tissue, cartilage, etc.) of the studied body part (Hounsfield 1973; Brooks y Chiro 1976). It is very useful in the diagnosis of many diseases with a high impact in the society (i.e cancer), allowing a very precise representation of the inside of the studied body parts, facilitating the identification of foreign bodies such as tumours or blood clots that before its application could only be discovered through surgery. However, the high dose of radiation induced to the patient from CT devices limits its use in the clinical practice, especially for the more vulnerable patients (e.g children and pregnant women) or patients whose diseases would require frequent CT scans, such as cancer patients, to assess the evolution of tumours. It has been proved that X-rays in high doses are a source of cancer by themselves, which sometimes could pose more risk than the disease. Every modern hospital is applying dose watching programs and dose monitoring to assess the amount of radiation their patients are exposed to. For this, it is vital to apply new reconstruction methods to reduce the radiation dose.
Since the introduction of iterative methods in the field of image reconstruction, such as (Andersen 1989; Andersen y Kak 1984; Yu y Zeng 2014), the possibility of working with fewer projections than the used by direct methods has arisen, due to the redundancy of information produced in the data acquisition process. Thereby taking into account this redundancy, the application of efficient methods based on less data input (X-ray projections) is possible, which directly leads to less induced radiation. Therefore, the objective of this work is to perform a multi-parametric study to explore the behaviour of every possible configuration of the iterative method chosen to reconstruct CT images. The final intent of the study is the identification of the optimal parameters that lead to a high-quality reconstruction, valid to the clinical practice, using the lowest amount of projections possible reducing the dose of radiation to potentially improve the health of patients requiring a CT diagnostic test.

However, if we want to analyse all configurations, we need to consume days or weeks of computing time. In this case, Grid Computing seems an appropriate approach because it provides a big amount of computing and storage resources where the different configurations can be executed in parallel as a batch jobs in a High-Throughput Model. However, we need to adapt the iterative method to the Grid paradigm, with the added complexity of the management of the software dependencies that the Grid jobs have. Similar works to perform a parameter exploration of biological systems models on a Grid platform can be found at (Mosca y col. 2009; Merelli y col. 2011). Thus, we compare in this paper a traditional embedding of software libraries in Storage Elements (SEs), with the use of containers to encapsulate those Grid jobs making use of the tool udocker (Gomes y col. 2018), which provides a user-space execution of containers.

4.2 Materials and Methods

In this section we introduce the iterative reconstruction method and the relevant parameters to analyse in the study. After that, we present the grid infrastructure and the code of the study. Finally, we discuss the issues related to the set up of the experiment for its execution in the grid infrastructure, using two methods: using the traditional way for storing the software libraries needed by the Grid jobs, and using containers to encapsulate them.

4.2.1 Reconstruction method and relevant parameters

The reconstruction method chosen for the study is the one proposed by (Flores, Vidal y Verdú 2015), which combines the LSQR (Paige y Saunders 1982) algorithm for the iterative resolution of the equation system representing the CT Image Reconstruction problem, with a Soft Thresholding Filter to eliminate noise and artifacts (STF, Yu y Wang 2010), and an acceleration stage that helps the method to converge more rapidly (FISTA, Beck y Teboulle 2009). As a whole, this method seems to bring good results, especially for reconstructions employing a limited number of projections or views, although it has not been studied in depth in the referenced works. The reconstruction method contains certain parameters that will directly influence both the quality of the reconstruction and the number of iterations required to reach convergence. For each chosen phantom, we need to set:

1. The resolution of the reconstructed image.
2. The number of projections taken.

3. The number of internal iterations of the LSQR algorithm.
4. Whether or not the filtering technique is applied.
5. In case of applying the filter, it must also be considered the alpha parameter of the STF algorithm, that as explained in (Yu y Wang 2010) influences the weight assigned to the diagonal gradient in the filtering algorithm and can consequently change the quality of the obtained image.
6. Whether or not the acceleration phase is applied.

Therefore, the objective is to study all of these parameters in depth to determine the values that achieve the optimal image quality. In this way, the technicians in charge of performing the reconstruction can configure the method to fit their necessities, i.e they may need to obtain a reconstruction that reaches a desired quality, and having previous knowledge of the behaviour of the method they can decide accordingly the number of views to employ. With the ranges of values chosen for the experiment we need a total of 16800 executions of the method for a single phantom. The projections have been generated for a phantom corresponding to the mathematical representation of the human head developed by the Forbild Phantom Group (FORBILD Phantom Group 2021), employing the medical imaging program CONRAD (Maier y col. 2013) to generate the phantom images for each resolution, which will also be taken as a reference to measure the quality of the reconstruction.

4.2.2 Input data

The problem of CT image reconstruction is represented by the equations system defined in (4.1). In order to solve it, we need two sets of input data. First, the system matrix A (4.4), which is calculated once for each desired resolution of the reconstructed image. Also the projections vector g (4.2), which will be different for each resolution and for each projected section of an object. Both data sets must be generated for a CT scanner model with the same features.
The matrix A is of size MxN, where M is the number of rays traced, and N the pixel size of the image to be reconstructed. It is obtained by discretizing the scanning space in pixels and measuring the influence of each beam that is traced in each pixel, as defined by the equation (4.4), where $a_{i,j}$ represents the contribution of pixel j to the ray i. To calculate the weights of each beam we have applied the Forward Projection method proposed by Joseph in (Joseph 1982). The resulting matrix is rectangular, disperse and relatively large.
The projections vector is the calculation of the attenuation experienced by rays passing through the object of study. The size of the vector for each projection angle is 1xn, being n the number of detectors forming the scanner. By

concatenating all of them we obtain a one-dimensional vector representing the whole object.
The algorithm we use to solve the system is LSQR, which iteratively approximates u (the reconstructed image, (4.3)), until it reaches a convergence criterion. In this case, we consider the process complete when either the relative residual norm ‖g-Au‖/‖g‖ reaches the desired tolerance of 1e-6 or the method iterates 10000 times without converging.
It should be noted that when we talk about reducing views or projections when using this algorithm, it means that projections are selected to cover the 360^{o} but with spaces greater than 1^{o} between the chosen angles, using the corresponding submatrix of A to solve the problem. Depending on the number of views to be used, the selected angles are calculated according to the equation (4.5). In this case, we generated the data for different resolutions of the reconstructed image (N).The resolutions chosen are 32x32, 64x64, 128x128, 256x256 and 512x512 pixels. The matrix A obtained for every case is large, from 200MB for the 32x32 case up to 3.6GB for the resolution 512x512.

$$A * u = g \tag{4.1}$$

$$g = [g_1, g_2, ..., g_M]^T \in R^M \tag{4.2}$$

$$u = [u_1, u_2, ..., u_N]^T \in R^N \tag{4.3}$$

$$A = a_{i,j} \in R^{MxN} \tag{4.4}$$

$$\Theta_i = \begin{cases} (360/views) * (i-1) & if \quad 1 <= i <= (views/2) \\ 1.5 + (360/views) * (i-1) & if \quad (views/2) < i < (views) \\ (360-1) & if \quad i = views \end{cases} \tag{4.5}$$

4.2.3 Grid infrastructure

The platform employed to carry out the study is the European Grid Infrastructure (EGI). The infrastructure is publicly funded to give scientists access to over 650,000 logical CPUs and more than 500 PB of disk storage to drive research and innovation in Europe.
For this work, we have employed "Biomed" Virtual Organisation, which gives support to Medical Image Analysis. EGI includes Unified Middleware Distribution (UMD) as a middleware software which offers the end-users a set of services that allows them to access and orchestrate all computing and storage resources of the EGI infrastructure.

4.2.4 Sequential Code of the iterative reconstruction method

When we are facing the adaptation of a program to a Grid platform such as EGI, one important fact is that it comprises heterogeneous resources, which are managed by different organizations, and therefore nothing guarantees that all the Computing Elements (CEs) will have the necessary software. In turn, it can not be forgotten that in this type of infrastructure the users do not have the permissions that enable them to install software that requires privileges, which makes difficult the task of designing the tests.
The original code developed in the test phase of the method was implemented in the programming language of the numerical computing program MATLAB. When adapting the code available for its use in Grid, a problem arose when trying to use the MATLAB code, since with it being proprietary software it is not available in the Computing Elements of the "Biomed" VO. Finally, we came up with the solution to run MATLAB code previously compiled using MATLAB Compiler (The MathWorks, Inc. 2021b). In this way, the code is fully portable, since the compilation process generates a binary executable that makes use of the MATLAB license that has been used for the compilation. However, to run the program, we previously have to install the MATLAB Runtime in each Computing Element to be used, but it does not need privileges. The Runtime installer file is 1.23 GB in size. The installation of the Runtime is non-interactive, and when it finishes it only requires to modify the environment variable `LD_LIBRARY_PATH` adding the installation folder to indicate where to look for the MATLAB libraries. Since this installation process is necessary for all jobs, we came up with two different ways of doing it:

- Making the Runtime installer available in an SE of the grid so that all the jobs can download it and proceed to perform the installation in the CE.
- Creating a Docker image (Merkel 2014) from an Ubuntu base image with the installation of MATLAB Runtime previously made, which is published in Docker Hub. In this way, the grid jobs will simply have to pull the image and create a container that runs an instance of it. Then the method program can be run inside the aforesaid container. However, since Docker is not available in every CE of Biomed, we made use of the tool "udocker" (Gomes y col. 2018), an application developed by the Portuguese *Laboratório de Instrumentaçao e Física Experimental de Particulas* (LIP) that allows the use of Docker without having to install it, which would require root privileges.

Nº of replicas	Failed Jobs	Average download time (sec.)
1	28	1934
2	12	1219
4	8	902
6	6	470

Tabla 4.1: SE usage statistics.

Set up of Storage Elements

The sequential code of the iterative method requires the previously generated input data explained in Section 4.2.2 (System Matrix and Projections Vector for each image resolution), the generated executable and the MATLAB Runtime installer, as well as the reference images, to compare the quality obtained. Consequently, it is necessary to make them available in the Grid so that the different CEs that are going to execute jobs can download them. As already explained, the necessary files are heavy, so it will be necessary to make use of the SEs for this type of assumptions, because it is not possible for these files to travel with the job through the Input Sandbox. In order to test the performance of the SEs and their efficiency, we carried out a small experiment. A total of 100 jobs were launched, measuring the time required to download the matrix corresponding to resolution 512x512, which is the heaviest, verifying in this way if the downloads were to cause bottlenecks that could delay the execution of the works excessively. The experiment was run first with only one replica, and then with two, four and six total replicas distributed between different SEs provided by EGI, to study the performance variation. The results of the 400 jobs executed are presented in Table 4.1.
As we can see, the number of failed jobs is reduced considerably when we replicate the file. Also, the average time to download it diminishes too, up until we have four replicas. When going from four to six replicas, the improvement is not so noticeable, especially regarding the number of failed jobs. However, by looking at the time statistics, we have observed that a larger number of replicas achieve lower downloading times, concentrating the best 50% of those corresponding to 6 replicas in times lower than 200 seconds, while the best 50% of those corresponding to 4 replicas are concentrated in times lower than 500 seconds. Therefore, not only for the slight improvement of the times, but also for preventing failed jobs, we chose to make 6 replicas in order to re-launch the minimum jobs possible.

4.2.5 Grid Jobs Structure

The main script, that we will call Starter, will be the main executable for all jobs. In Figure 4.1a we show the process that is performed. It can be observed that the process that the script performs is different depending on whether the local installation of the Runtime is made, or the Docker image with the included installation is used. It should be noted that for efficiency reasons, since the download of the data can take a considerable time, the executions of the method have been grouped into groups of four. Namely, for a combination of the parameters, the method is studied applying only the LSQR phase, then combining it with filter, combining it with acceleration, and finally combining it with both. In this way, the 16,800 executions required will be carried out by 4200 grid jobs.

4.2.6 Grid Study Workflow

Once we have the code of the main process for each job implemented, it is necessary to automate the creation of the JDL files corresponding to each job, the launching and monitoring of each one of them, and finally the collection of results. Due to the high number of jobs required it is essential to develop the tools that perform all this process automatically since it is not feasible to do it manually. This whole process has been developed using Bash scripts. Figure 4.1b shows the flowchart of the jobs life cycle management, the steps of which consist in the creation of the JDL files that will result from the combination of all the values of the input parameters, then the uploading and replication of the required files to an SE, followed by the launching of all the resulting jobs, the monitoring of their status, and finally the retrieval of all the results when the jobs are completed.

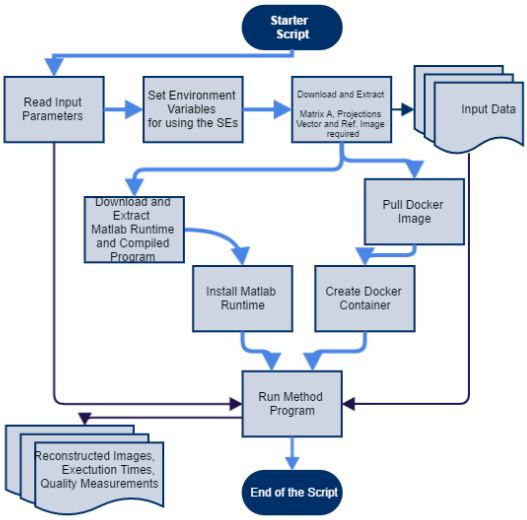
4.3 Results and Discussion

4.3.1 Grid platform performance

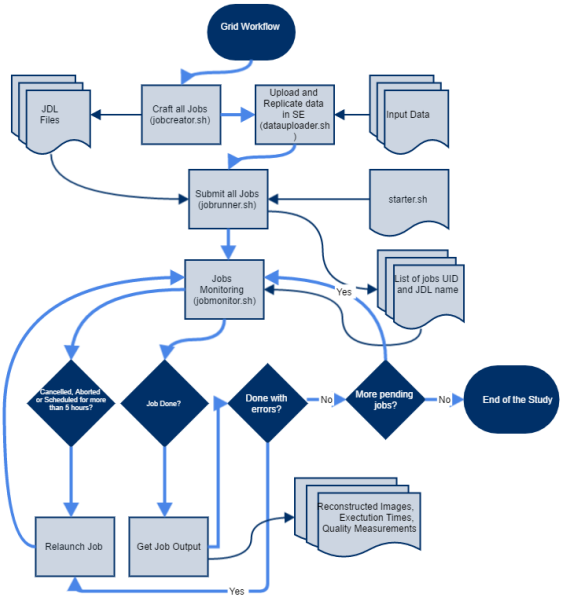
Once the study has been performed, we have assessed the performance provided by the Grid platform. Figure 4.2 shows the number of jobs per hour that have been submitted, dividing the launch into five threads, one per resolution, for a single phantom. In this way, the 4200 necessary jobs have been submitted in a time a little more than three hours. As can be observed in Figure 4.2, there is a small decrease in the number of jobs submitted every hour, which may be due to the saturation of the Workload Management Systems (WMS), since

they are managing an increasingly large number of jobs. However, this slight deterioration is not too important, since there are still more than 1250 jobs being submitted every hour. Figure 4.3a shows the average execution times of the jobs divided by resolution when downloading the Runtime from an SE and installing it locally. It can be observed that for small resolutions (32x32 and 64x64), the time that the jobs remain in the Scheduled state, and the time required to perform the data download is greater than the time required for the execution of the method. However, when increasing the size of the problem, the application of the method requires more time, reaching the maximum for resolution 512x512. Regarding the data download time, we notice that the average time remains very similar for all resolutions, regardless of the size of the matrices to be downloaded. This seems to indicate that the influence of the number of jobs that download the same file is greater than the size of the file. Finally, it is observed that the time that the jobs take to start running rises as we increase the size of the problem, which may be due to the WMS giving a higher priority to the jobs that have a lower estimated execution time. In relation to the jobs executed with Docker containers, Figure 4.3b shows the average life times of the jobs. As we observe, the waiting time remains similar to the previous case, since it does not depend on the work to be carried out. In turn, it is appreciable how the download time decreases, since in this case it is not necessary to download the MATLAB Runtime installer from an SE. The pull time of the Docker image from Docker Hub and the creation of the required container is relatively small, getting the container ready for its use in a slightly shorter time than was required to perform the local installation.

However, the increase of the execution time is notable, except in the case of resolution 512x512. This may be due to the fact that running the method inside a Docker container is slower than doing so directly over the CE because the containers do not make use of the 100% machine resources.
In spite of what is shown in Figures 4.3a and 4.3b, in which the average times are shown, it should be noted that the method being studied is of iterative type, and therefore does not end until it reaches the convergence criterion. It is, therefore, impossible to make a previous estimate of the execution time in each possible case, which may mean that sometimes when the method converges with few iterations, the waiting time and download time is greater than the execution time. However, observing the time statistics we can conclude that the execution times corresponding to the method application are generally much longer than the download time. For this reason, it can be determined that despite the generation of a delay due to the necessary data download, it is not exceedingly large, so it is still profitable to carry out the study in the grid platform instead of being carried out on a local machine.



(a) Starter Script Workflow



(b) Grid Study Workflow

Figura 4.1: Implemented Workflows.

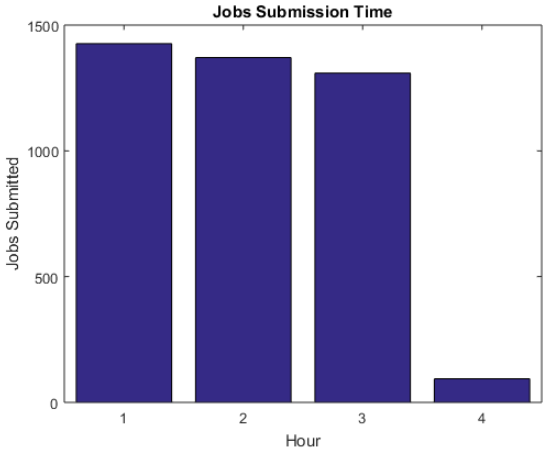
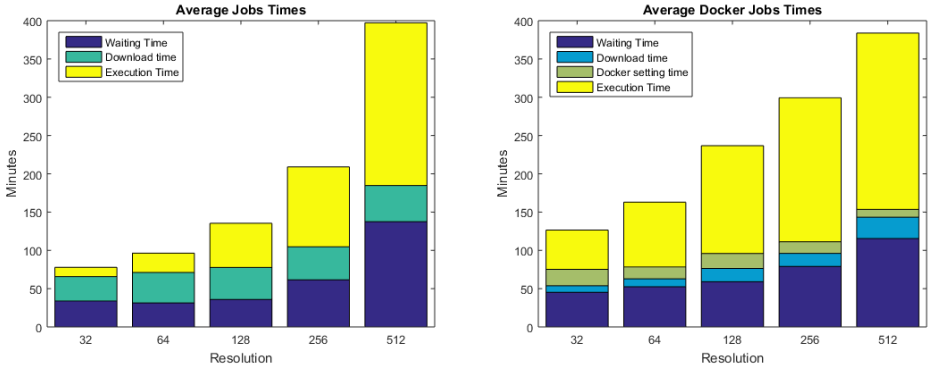


Figura 4.2: Jobs Submission Time



(a) Regular Jobs **(b)** Docker Jobs

Figura 4.3: Average Jobs Life Time.

4.3.2 Reconstruction method parameters

With the in-depth exploration of all the desired parameters, we can perform an analysis of the results of the reconstructions. To determine the best reconstruction, we have considered the optimal combination between required iterations and quality obtained. The quality of the images has been measured using the metric PSNR (Flores, Vidal y Verdú 2015), and the best results are shown in Figure 4.4.

For small resolutions, i.e 32x32 and 64x64, we get good results for any number of views chosen (from 30 to 180), provided we do not apply the filtering and acceleration phase. Applying the filter in these cases causes the method to diverge and therefore there is a significant increase in the number of required iterations, reaching the maximum iterations and stopping despite not having converged.
From resolution 128x128 to 512x512, the filter and acceleration are necessary, since their combination with the algorithm LSQR significantly reduces the number of iterations and increases the image quality. If they are not applied, the method generally does not converge before 10000 iterations. In addition, it is observed that as the number of views is reduced, the number of required iterations increases, since the solution has to be approximated from less information. Introducing these techniques causes that we also have to study how many internal iterations of LSQR to apply before applying them. In all

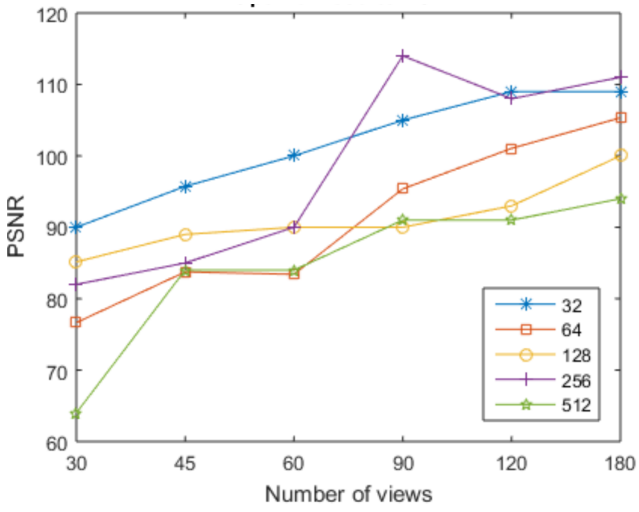
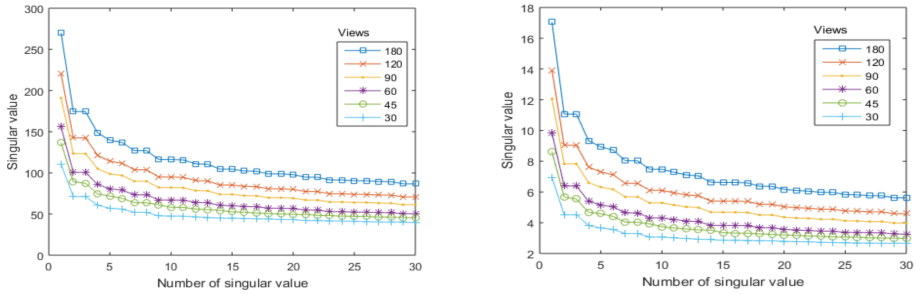


Figura 4.4: Evolution of optimal PSNR.

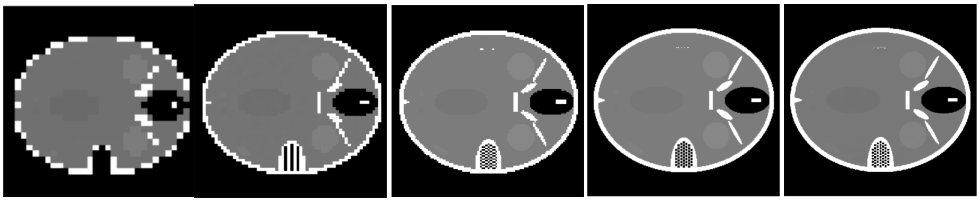


(a) Res. 32x32 **(b)** Res. 512x512

Figura 4.5: First 30 singular values.

cases we have observed that reducing the number of views causes the number of internal iterations of LSQR to be reduced, so the filter and acceleration phase is more frequently applied. The same thing happens when increasing the resolution.

All of the above can be justified by studying the characteristics of the singular values of the matrices that are formed for each case. We can observe that by increasing the resolution of the image to be reconstructed and reducing the number of projections, the singular values of the image are concentrated in a smaller interval. For instance, in Figures 4.5a and 4.5b we can see the first 30 singular values in every case. As observed, the singular values of the matrix corresponding to the resolution of 32x32 are concentrated in a much broader interval than the 512x512 cases. For example, for 30 views and resolution 32x32, the first 30 singular values vary in 70 units, while the ones corresponding to resolution 512x512 and 30 views vary in only 4 units. This makes the method converge more slowly since LSQR is an algorithm based on Krylov subspaces. For this reason, when the singular values are more concentrated, it is necessary to help the method with the filter and the acceleration to improve the process. For every resolution we have found configurations that obtain very satisfactory results, as can be observed in Figure 4.4, which allows us to perform high-quality reconstructions from a very small number of views, in this case 30, as can be seen in Figure 4.6. In addition, it is achieved in a relatively low number of iterations, guaranteeing a good average time of reconstruction.



(a) Res. 32x32
Total Iter: 381

(b) Res. 64x64
Total Iter: 957

(c) Res. 128x128
Total Iter: 2639
LSQR iter: 13

(d) Res. 256x256
Total Iter: 3325
LSQR iter: 6

(e) Res. 512x512
Total Iter: 3576
LSQR iter: 4

Figura 4.6: Best reconstructions with 30 views.

4.4 Conclusion

In this work, we have performed a multiparametric study of a new iterative method for the reconstruction of CT images using a Grid platform. Although Grid computing is not fit for the actual application of CT Image Reconstruction techniques since time is critical in this field, it can be really useful to study the quality of new methods in the developing phase.
The code of the method has been adapted for its execution in the Grid, making use of MATLAB Runtime, and the necessary tools have been developed for the launching and monitoring of the required jobs. In turn, the method has been executed in two different ways, through a local installation of MATLAB Runtime in the CE after downloading it from an SE, and by using a Docker container that runs an instance of an image which includes a previous installation of the Runtime. In this way, it has been possible to make a comparison of both techniques, obtaining slightly better execution times in the first case. However, this is possibly due to the large size of the problem to be solved, and it is likely that for smaller problems the use of Docker containers on Grid platforms using tools such as "udocker" can be very beneficial to extend Grid utility, since it allows users to make use of software that is not available in the CE without the need to have root permissions. In addition, now that the use of Docker on a Grid platform has been proved to be feasible, for future works we can develop optimised C versions of the methods using specific libraries to facilitate the use of matrices. In this way, we would not have to perform a static compilation of the program to make it work in any CE, we could just compile it inside the Docker container which is an easier approach.
The obtained results have allowed us to determine the optimal configurations that allow us to apply the method reducing the X-ray dose emitted in the traditional CTs up to an 80%, since we can use only 30 views instead of the 360 used in the conventional methods, conserving the image quality. Besides,

with all the data we have gathered, we can conclude that the quality of the resulting image and the convergence speed of the process relies heavily on the characteristics of the matrix A, being the filter and acceleration technique a necessary step for high resolutions.
With all the above, we have achieved in-depth knowledge of the behaviour of our method, which will certainly help to apply it correctly, always ensuring convergence and minimum image quality.

Acknowledgments

This work has been supported by Universitat Politècnica de València and partially funded by TIN2015-66972-C5-4-R, ENE2014-59442-P-AR and TIN2013-44390-R of the "Ministerio de Economía y Competitividad" of Spain, as well as the Spanish "Generalitat Valenciana" PROMETEOII/2014/008 project.

Capítulo 5

Evaluación de filtros de imagen para la combinación con LSQR

El artículo presentado en este capítulo, publicado en la revista *PLOS ONE* en 2020 y con referencia (Chillarón, Vidal y Verdú 2020b) lleva por título *Evaluation of Image Filters for their integration with LSQR Computerized Tomography Reconstruction Method.*

En él se realiza un estudio de cuatro filtros de imagen conocidos: filtro Gaussiano, mediana, filtro Wiener y Bilateral. En el estudio se analiza el efecto de cada filtro para eliminar dos tipos de ruido: Gaussiano y Speckle. Con este estudio se pretende analizar el comportamiento de estos filtros con el fin de evaluar su utilidad para el problema de reconstrucción de imagen TC, en concreto para la reconstrucción algebraica iterativa con reducción de proyecciones.

Para evaluarlos se han seguido dos estrategias diferentes. En la primera, los dos tipos de ruido se añaden sobre el propio sinograma con reducción de vistas, considerando diferentes tamaño de ventana y valor de varianza del ruido. Después, se aplican los filtros sobre el sinograma con ruido, cuantificando la cantidad de ruido que han conseguido eliminar mediante la métrica PSNR. Posteriormente, se utiliza el sinograma filtrado mediante el método LSQR pa-

ra evaluar el efecto que tiene este proceso de filtrado en las imágenes finales cuando se emplea un método algebraico iterativo con reducción de vistas.

En este supuesto estaríamos simulando un sinograma no tratado, con datos en bruto. Este paso no suele ser necesario en los escáneres comerciales puesto que los propios fabricantes incorporan sus técnicas para limpiar los sinogramas antes de ser reconstruidos, y no proporcionan los datos brutos. Se ha realizado este proceso porque se considera que puede ser un paso necesario durante el desarrollo de nuevos escáneres con adquisición dispersa.

En ambos tipos de ruido se ha comprobado que el filtro Bilateral es el que consigue un PSNR más alto midiendo el sinograma filtrado con el sinograma sin filtrar. Con el ruido Gaussiano, la diferencia es grande con respecto al resto de filtros. Sin embargo, para el ruido Speckle, los resultados obtenidos son comparables a los obtenidos con el filtro Gaussiano con sigma igual a 0.5. De estos resultados se puede determinar que ni el filtro Wiener ni el de mediana obtienen una buena eliminación del ruido en el sinograma.

Con el fin de visualizar el efecto del filtrado sobre las imágenes finales se ha realizado una reconstrucción de cada sinograma con LSQR y 200 iteraciones. Para ambos tipos de ruido el filtro Bilateral proporciona reconstrucciones más limpias a nivel visual, si bien se han perdido estructuras internas del fantoma con el proceso de filtrado y reconstrucción que no son recuperables debido al ruido. En el caso del ruido Speckle, confirmando lo que mostraba el PSNR del singorama, el filtro Gaussiano también aporta buen rendimiento, obteniendo una imagen similar en cuanto a conservación de estructuras internas.

La segunda estrategia de eliminación de ruido ha sido añadir el ruido Gaussiano y Speckle sobre el propio fantoma, para después aplicar los cuatro filtros. Los resultados muestran que para los dos tipos de ruido, el filtro Bilateral es el único que mejora tanto en PSNR como en SSIM en ambos casos, a pesar de no eliminar todo el ruido de la imagen. Que obtenga mejores resultados con las dos métricas con respecto a la imagen con ruido significa que es capaz de eliminar ruido de la imagen sin eliminar también las estructuras internas, cosa que no consigue ninguno de los otros filtros. Se puede apreciar que el filtro Bilateral obtiene mejor rendimiento eliminando el ruido Speckle, y no deforma en ningún caso las estructuras internas, lo cuál sí hacen los demás filtros, si bien los bordes no están tan bien definidos como en el fantoma original.

A la vista de dichos resultados, se ha decidido incorporar el filtro Bilateral como paso extra en la reconstrucción LSQR+STF+FISTA, con resultados muy satisfactorios. Para una resolución de imagen de 512×512 píxeles, proyec-

tando ahora una imagen anatómica real correspondiente a un TC de abdomen en lugar de un fantoma matemático, se ha comprobado que la incorporación del filtro Bilateral en el proceso de reconstrucción aumenta la calidad de las imágenes obtenidas, tanto midiendo el PSNR como el SSIM. Esto se cumple para todas las reconstrucciones realizadas, variando el número de vistas desde 60 hasta 180. Por ejemplo, para 60 vistas y 5 iteraciones internas de LSQR, el PSNR pasa de 35.62 a 38.99 incorporando el filtro, y el SSIM de 0.8052 a 0.9607, lo cual es un incremento importante.

Por todo ello, de este estudio se concluye que este filtro es muy apropiado para el problema de reconstrucción algebraico con disminución de proyecciones, ya que consigue eliminar ruido de la imagen sin alterar las estructuras internas como sucede con otros filtros. Además, a parte de poder convertirse en una herramienta para la reducción del ruido sobre el sinograma, es especialmente útil cuando se combina con las técnicas STF y FISTA, obteniendo los mejores resultados cuando las tres se combinan con LSQR en la mayoría de los casos.

PLoS ONE

Volume 15(3): e0229113, 2020

Evaluation of Image Filters for their integration with LSQR Computerized Tomography Reconstruction Method

Mónica Chillarón[1], Vicente Vidal[1] and Gumersindo Verdú[2]

1 Department of Computer Systems and Computation, Universitat Politècnica de València, Valencia, Spain.
2 Department of Chemical and Nuclear Engineering, Universitat Politècnica de València, Valencia, Spain.

Abstract

In CT (computerized tomography) imaging reconstruction, the acquired sinograms are usually noisy, so artifacts will appear on the resulting images. Thus, it is necessary to find the adequate filters to combine with reconstruction methods that eliminate the greater amount of noise possible without altering in excess the information that the image contains.

The present work is focused on the evaluation of several filtering techniques applied in the elimination of artifacts present in CT sinograms. In particular, we analyze the elimination of Gaussian and Speckle noise. The chosen filtering techniques have been studied using four functions designed to measure the quality of the filtered image and compare it with a reference image. In this way, we determine the ideal parameters to carry out the filtering process on the sinograms, prior to the process of reconstruction of the images.

Moreover, we study their application on reconstructed noisy images when using noisy sinograms and finally we select the best filter to combine with an iterative reconstruction method in order to test if it improves the quality of the images.

With this, we can determine the feasibility of using the selected filtering method for our CT reconstructions with projections reduction, concluding that the bilateral filter is the filter that behaves best with our images. We will test it when combined with our iterative reconstruction method, which consists on the Least Squares QR method in combination with a regularization technique and an acceleration step, showing how integrating this filter with our reconstruction method improves the quality of the CT images.

5.1 Introduction

In the field of health sciences, the advances in computer science and technology have allowed medicine to improve in aspects that before would have been unthinkable. For instance, the invention of magnetic resonances (MRI) or computerized tomographies (CT). Therefore, it can be said that the quality of life has been improved thanks to technology and the use of information technology as an advanced scientific instrumentation for medical applications. One of the breakthrough that we can highlight in this regard is the improvement and optimization of images for medical diagnostic purposes.

The disadvantage of the CT scanning process is the ionizing radiation used to project a body part. Each type of tissue absorbs a greater or lesser amount of radiation depending on its density. Thus, they appear in different tonalities and we can differentiate muscle tissue, bone, elements of the cardiovascular system, etc., in the images. The problem lies in the amount of radiation necessary to obtain images of acceptable resolution and sharpness. Taking a single X-ray test as a reference, a CT scanner can entail a radiation dose hundreds of times higher. In the particular case of a thorax image, the effective radiation for obtaining it by a CT scan is approximately 400 times greater than the dose necessary for the same result with a single X-ray (Rehani y col. 2004).

From all of the above arises the need to reduce the effective radiation dose. This means reducing the dose for the generation of the images, which results in less exposure of the patient, but also a lower definition of the image.

Until a few years ago, image reconstruction methods based on filtered back-projection were the more commonly used mainly because they can be executed on older computers, since they have a low computational cost and the reconstruction can be done in a relatively low time. In turn, the quality of the resulting images is more than correct when a high number of projections is used, a situation in which the patient absorbs a high dose of radiation. However, the quality of the images gets worse when the number of projections is reduced.

That is why the algebraic methods of approximation started to be applied to reconstruct CT images, such as the LSQR (Least Squares QR) algorithm. They are capable of working with fewer views, as we have shown in our previous works (Chillarón y col. 2017; Flores, Vidal y Verdú 2015; Flores y col. 2014; Parcero y col. 2017). But working with less projections also means more noise in the images.

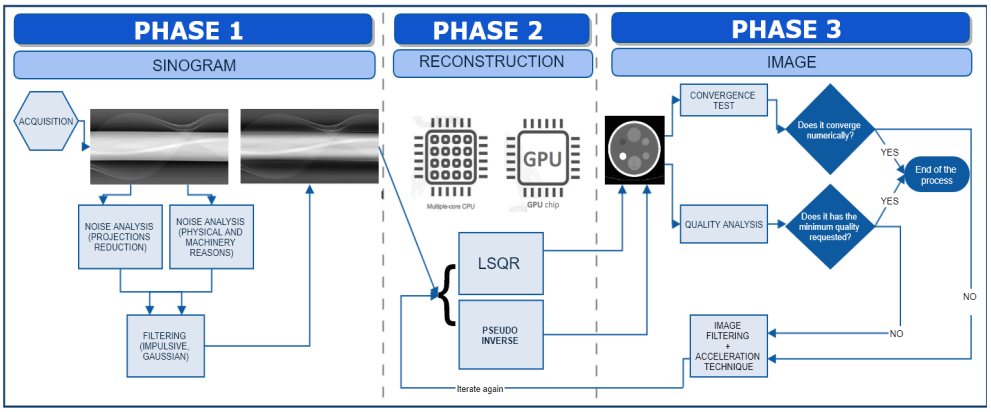


Figura 5.1: Complete CT image reconstruction process. Process of reconstructing a CT image, showing the steps from the acquisition to the final image.

In studies about Gaussian and Speckle in CT images, MRI or Ultrasounds (Kumar y col. 2014; Senthilraja, Suresh y Suganthi 2014), the filtering techniques and algorithms proposed are applied to an already reconstructed image. Nevertheless, they could also be applied to the projection data. Therefore, the main goal of this paper is to find the most appropriate filter that can be applied either to sinograms or on CT images, so we can improve the quality of our CT reconstructions. In particular, we aim to select a filter that can be combined with an iterative reconstruction method (LSQR) as an additional step on the reconstruction algorithm when we use few projections, reducing the image noise at the same time we reduce the radiation dose.

We will explain the phases of our CT reconstruction process in Section 5.2, and also describe the chosen image filters. In Section 5.3.2 we analyze the behavior of the filters when Gaussian and Speckle noise is added on a sinogram with few projections, as well as the subsequent evaluation of the reconstructions obtained when using the filtered sinograms. In addition, in Section 5.3.3 we will test the filters on phantom images with added noise to see if we can improve the final result. In Section 5.3.4 we show the results of combining the best filter with our few-view reconstruction method. We conclude with Section 5.4 where we sum up the obtained results.

5.2 Materials and Methods

5.2.1 CT image reconstruction process

When the sinograms are acquired and few views (projections) are used in the reconstruction phase of the image, artifacts of the stair-step type (Barrett y Keat 2004) and Gaussian and/or impulsive noise might appear. Fig 5.1 shows the complete process of reconstructing a CT image in two dimensions.

Phase 1 consists of the acquisition (or generation if we simulate it) of the sinogram. In this step we also study and try to eliminate the noise that may be present in the data.

In phase 2, called the algebraic reconstruction phase of the CT image, we perform the resolution of the equations system that models the CT, $Af = g+w$, where $A = (a_{i,j}) \in \mathbb{R}^{MxN}$ is the system matrix, $g \in \mathbb{R}^M$ the projections vector, $w \in \mathbb{R}^M$ the noise vector and $f \in \mathbb{R}^N$ the desired image. The dimension M is the product of the number of detectors that the CT scanner has (in our case 1025) multiplied by the number of projections or views taken. N denotes the resolution of the image (128×128 pixels, 256×256 pixels, etc). This phase has been analyzed in previous works (Chillarón y col. 2017; Flores, Vidal y Verdú 2015; Flores y col. 2014; Parcero y col. 2017; Chillarón y col. 2018), using several different methods to solve the equations system. Nevertheless, in those works, the approach did not include an image filter combined with the iterative process of resolution, which we will study in this paper.

Last, in phase 3, the quality of the image obtained in phase 2 is analyzed and it is decided how to proceed. We can measure the quality by image quality metrics (such as Peak Signal-to-Noise ratio) or iterate until we reach a desired relative residual $norm(g - Af)/norm(g)$ (usually 10^{-6}). Both in phase 1 (the generation of the sinogram) and in phase 3 (improvement of the image), image filters must be used to reduce noise, which is the main task on which the present work focuses.

To carry out the proposed studies, it is necessary to generate the system matrix A, and the projection vector g. Both data sets will have to be generated once for each desired image resolution, which has been done by the Joseph Forward Projection method (Joseph 1982). This method calculates the weight of each ray on each pixel (forming the matrix A) weighted by the corresponding coefficients of linear attenuation of the rays. For this simulation, the Forbild Head Phantom (FORBILD Phantom Group 2021) phantom was used, which mathematically represents an approximation of the existing structures in a

human head. We also use an abdomen CT image selected from the dataset DeepLession (Yan y col. 2018) to test the quality of the reconstructions.

The sinogram vector g is formed selecting the desired projections angles from the complete generated projections. Depending on the number of views used, the amount of noise induced will vary. For these simulations, we modeled a fan-beam axial scanner.

5.2.2 Iterative reconstruction method

The reconstruction method (phase 2 and 3) that is used consists of three processes that are repeated until convergence is achieved: a first process that solves the equations system. After this, an approximate solution image is obtained. Then a filtering of the said image, followed by an acceleration process that prepares the method to re-iterate if convergence has not been achieved.

For solving the equations system, we use the Least Squares QR method (LSQR) (Paige y Saunders 1982) since it is one of the most stable methods. The method solves the system $Af = g + w$ for $w = 0$ by minimizing $min||Af - g||_2$.

The non-linear filter *Soft Thresholding Filter* (Yu y Wang 2010; Yu y Zeng 2014) is applied after the LSQR. This filter acts as a regularization technique when we reduce the number of projections. It helps convergence, conserving not only the vertical-horizontal but also the diagonal gradient, causing the image to be sharper without losing its structure at the edges, which is of vital importance in medical CT imaging tests. Finally, the acceleration technique *Fast Iterative Shrinkage Thresholding Algorithm* (FISTA) defined in (Beck y Teboulle 2009) is applied, which reduce the number of total iterations needed.

5.2.3 Selected image filters

Considering the variety of image filtering techniques, it is necessary to limit the number of methods to test. Taking as reference the work (Senthilraja, Suresh y Suganthi 2014), the following filtering techniques have been chosen.

Gaussian filter

It is the result of smoothing the image by means of a Gaussian function (Jain 1989). At the expense of reducing or eliminating noise, there is a risk of losing a large amount of detail due to the fact that the edges are not preserved. Thus obtaining a blurred and unclear image.

The Gaussian filter is applied to a 2D image as defined in (5.1), where G is the Gaussian mask with coordinates (x, y), σ is the parameter that defines the standard deviation. If the value of σ is large, the image smoothing effect will be greater. The smoothing can be done by convolutioning a window of the original image I *(x, y)* of size $w \times h$ with a Gaussian mask G as illustrated in the Eq. (5.2). The filtered image is obtained by calculating the sum of products between all the pixels of the input image window and the Gaussian matrix.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{5.1}$$

$$f(x, y) = \sum_{i=0}^{w-1} \sum_{j=0}^{h-1} G(i, j) I(x - i, y - j) \tag{5.2}$$

Median filter

The median filter is a non-linear method (Lim 1990). It is widely used as it is very effective in eliminating noise and preserving edges. It is particularly effective in eliminating 'salt and pepper' noise.

The median filter works by moving through the image pixel by pixel, replacing each value with the median value of the neighboring pixels. The neighbor pattern is called "window", which slides, pixel by pixel over the entire image.

In the limits of the image, there are no previous or subsequent values, the value of the repeated pixel itself is used in the zones that do not have values. It is also possible to fill empty spaces with zeros or ones.

Wiener filter

By inverse filtering, it eliminates the additive noise and inverts the blurring of the image simultaneously (Lim 1990). It considers images and noise as random variables. The objective is to find an estimate of the original image such that the mean square error between them is minimized.

The filter estimates the local mean (5.3) and the variance (5.4) around each pixel, where η is the local neighborhood $N \times M$ of each pixel in the image A. It then filters (5.5) pixel by pixel using these estimates, where v^2 is the variance of the noise. If the noise variance is not provided, the average of all estimated local variances is used.

$$\mu = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a(n_1, n_2) \tag{5.3}$$

$$\sigma^2 = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a^2(n_1, n_2) - \mu^2 \tag{5.4}$$

$$b(n_1, n_2) = \mu + \frac{\sigma^2 - v^2}{\sigma^2}(a(n_1, n_2) - \mu) \tag{5.5}$$

Bilateral filter

It is a non-linear filter and edge-preserving (Tomasi y Manduchi 1998). It replaces the value of each pixel with a weighted average of the pixels next to its location. The weighting is generally based on a normal distribution according to the values of the pixels.

The bilateral filter is defined as (5.6), where the normalization term (5.7) ensures that the filter preserves the energy of the image. I^{filtered} is the filtered image; I is the original input image to be filtered; x is the coordinates of the pixel to be filtered; Ω is the window centered on x; f_r is the range of the kernel to smooth the differences in intensities; g_s is the spatial range for smoothing coordinate differences. The last two functions can be Gaussian functions.

$$I^{\text{filtered}}(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} I(x_i) f_r(||I(x_i) - I(x)||) g_s(||x_i - x||) \tag{5.6}$$

$$W_p = \sum_{x_i \in \Omega} f_r(||I(x_i) - I(x)||) g_s(||x_i - x||) \tag{5.7}$$

The weight W_p is assigned by spatial closeness and intensity difference. Consider a pixel located in (i, j) that needs to be filtered with its neighboring pixels and one of its neighboring pixels is in (k, l). The weight assigned per pixel (k, l) to eliminate the noise of the pixel (i, j) is given by (5.8). Here, σ_d and σ_r are smoothing parameters and $I(i, j)$ and $I(k, l)$ are the intensity of the pixels (i, j) and (k, l) respectively . After calculating the weights, we must normalize them with (5.9), where I_D is the intensity without noise of the pixel (i, j).

$$w(i, j, k, l) = e^{(-\frac{(i-k)^2+(j-l)^2}{2\sigma_d^2} - \frac{||I(i,j)-I(k,l)||^2}{2\sigma_r^2})} \tag{5.8}$$

$$I_D(i, j) = \frac{\sum_{k,l} I(k, l) * w(i, j, k, l)}{\sum_{k,l} w(i, j, k, l)} \tag{5.9}$$

5.3 Results and Discussion

5.3.1 Added noise

Before analyzing the behavior of the filters we must determine the value of the variance parameter that should be used to add both the Gaussian and Speckle noise. To do this, we tried adding noise with different variances from 0.0001 to 0.005. With our phantom projections for resolution 256x256 and 60 views, we determine that the best variance value for both types of noise is 0.0005. This adds enough noise to the sinogram to appreciate the results of the filtering methods but we preserve most of the relevant information. The experiments have been conducted using the Matlab R2018B functions to add noise.

5.3.2 Sinogram filtering

For the application of the filters, we need to consider both the variance and window size. We have applied windows from 3x3 to 9x9 pixels, and a variance value from 0.3 to 0.9. The PSNR results varying these parameters for the filters with both types of noise are presented in Figs 5.2 and 5.3. From these Figures we extract the optimum parameters for each filter.

After filtering the sinogram with added Gaussian noise, we observe the bilateral filter (shown in Fig 5.4e) is the filtering method that achieves the best results of

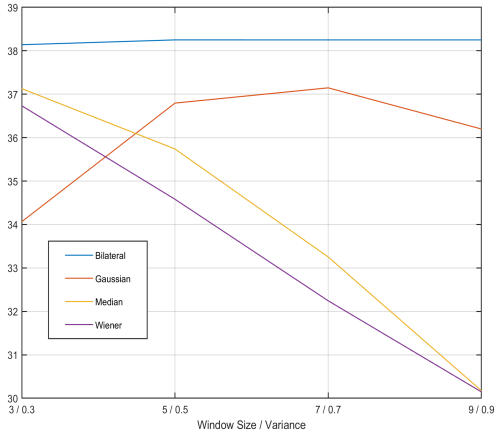


Figura 5.2: PSNR results with Gaussian noise varying the parameters. Results of filtering the sinogram with Gaussian noise using the four filters.

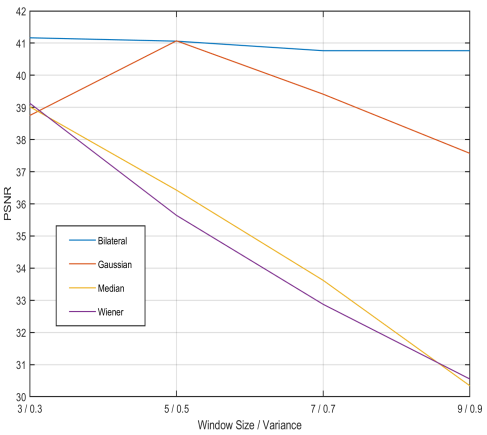


Figura 5.3: PSNR results with Speckle noise varying the parameters. Results of filtering the sinogram with Speckle noise using the four filters.

SSIM (Structural Similarity Index) and PSNR (Peak Signal-To-Noise Ratio) (Hore y Ziou 2010), in particular for 5x5 windows, although the results are very similar with every window size in this case. The second best results are obtained using the Gaussian filter (Fig 5.4b) with a sigma value of 0.7. Neither the median or the Wiener filter get good quality results since they alter too much the structures of the phantom. Regarding the sinograms with Speckle noise, the bilateral filter (Fig 5.5e) is also the best method, using a 3x3 window. Next, the Gaussian filter (Fig 5.5b) with a sigma value of 0.5, which gets very close to the results of the bilateral filter. As before, the Wiener and median filters don't improve the image. It must be noted that the reconstructions shown in Figs 5.4 and 5.5 were obtained with only 200 iterations of LSQR, so the method did not reach convergence.

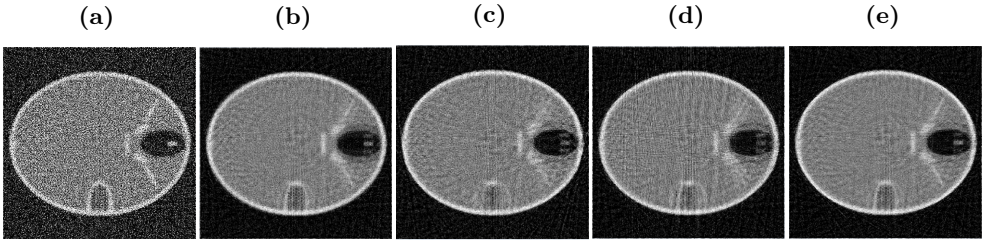


Figura 5.4: Gaussian noise reconstructions. A: Reconstructed Image with unfiltered Gaussian noise on the sinogram. B: Reconstruction using the Gaussian filter on the sinogram. C: Median filter. D: Wiener filter. E: Bilateral filter.

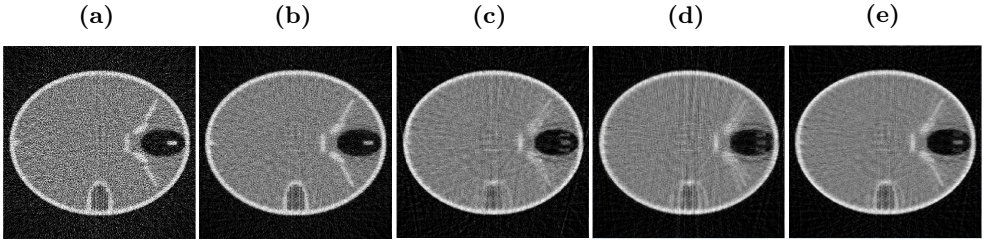


Figura 5.5: Speckle noise reconstructions. A: Reconstructed Image with unfiltered Speckle noise on the sinogram. B: Reconstruction using the Gaussian filter on the sinogram. C: Median filter. D: Wiener filter. E: Bilateral filter.

5.3.3 Phantom filtering

In an analogous way to the previous case, we add noise to the phantom images and then remove it using the selected filters. This can be useful when you have reconstructed images that still contain noise. The variance of the noise, as well as the window size and variance chosen for the filters are the same as before. After filtering the image with added Gaussian noise, we observe the bilateral filter (Fig 5.6e) is the only method that improves both the SSIM and PSNR (Tables 5.1 and 5.2), but the resulting image still has visible noise. The median filter obtains a good SSIM value since it smooths the image and the edges are better defined, but in Fig 5.6c we can observe that the internal structures are not well preserved, in particular the ear structure is altered. The Wiener filter (Fig 5.6d) alters too much the structures of the phantom and the Gaussian filter (Fig 5.6b) still leaves too much noise. Regarding the images with speckle noise, the bilateral filter (Fig 5.7e) is yet again the best method. This time,

it eliminates more noise on the image, so the edges are better defined and the SSIM shows that. The other filters behave similarly to the previous case, and each of them obtain a lower PSNR than the noisy image.

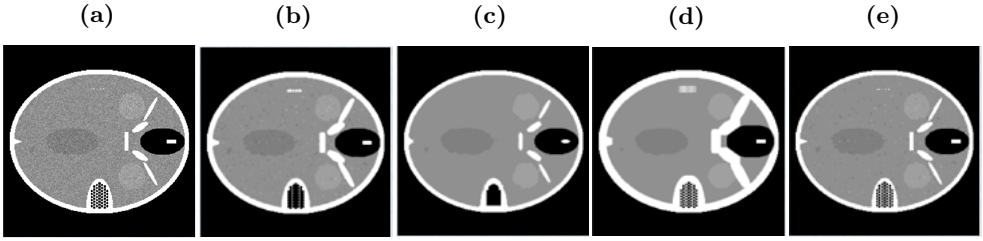


Figura 5.6: Gaussian noise filtering on the phantom image. A: Phantom with added Gaussian noise. B: Phantom filtered using Gaussian filter. C: Median filter. D: Wiener filter. E: Bilateral filter.

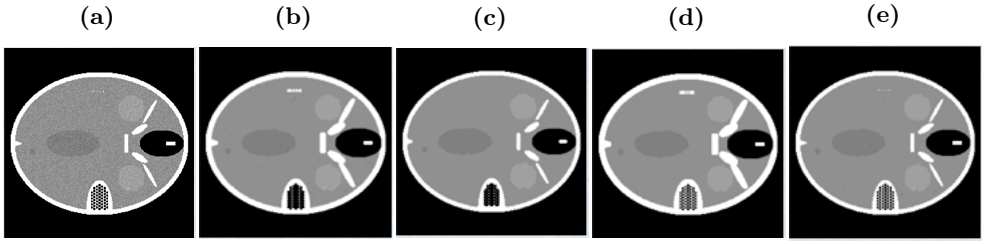


Figura 5.7: Speckle noise filtering on the phantom image. A: Phantom with added Speckle noise. B: Phantom filtered using Gaussian filter. C: Median filter. D: Wiener filter. E: Bilateral filter.

	SSIM	PSNR
Noise	0.24	50.66
Gaussian Filter	0.25	30.43
Median Filter	0.78	22.03
Wiener Filter	0.66	36.04
Bilateral Filter	0.43	57.64

Tabla 5.1: Phantom with Gaussian noise filtering results.

	SSIM	PSNR
Noise	0.68	54.71
Gaussian Filter	0.78	23.49
Median Filter	0.93	22.01
Wiener Filter	0.82	32.61
Bilateral Filter	0.94	62.06

Tabla 5.2: Phantom with Speckle noise filtering results.

5.3.4 Combination with iterative reconstruction

Since we have verified that the Bilateral filter is the best of the selected filters, we are going to integrate it with our iterative reconstruction technique. Thus, there are four steps in the proposed methodology to reconstruct the image from the sinograms. The first process, the LSQR technique, is used in all the combinations of the methodology and is the one that solves the equation $Af = g$. Then it can be combined with the regularization process, STF (Yu y Wang 2010; Yu y Zeng 2014) , since the system matrix is rank deficient when we reduce the number of projections and it can help to avoid the streaks artifacts. Also, it can be combined with a filter that eliminates Gaussian and Speckle noise, in our case we will use the Bilateral filter. And finally, it can be combined with the FISTA (Beck y Teboulle 2009) acceleration technique which, as we have seen in (Parcero y col. 2017), considerably reduces the number of iterations. The pseudocode of this process is shown in Algorithm 2, where we can specify which of the additional steps apart from LSQR we want to perform. For LSQR, we call the function that applies *iter* iterations of the method using the matrix A, projections vector g and initial image solution f to obtain the new approximation of the image f. For this research, the Matlab *lsqr* function has been used. The Bilateral filter, STF and FISTA methods are applied to image f, and they overwrite it. It is worth mentioning that the computational cost of the methodology is dominated by the cost of the LSQR process.

For this experiment, we have used an abdominal CT image selected from the dataset DeepLesion (Yan y col. 2018), and projected with Joseph method for resolution 512x512 pixels. In Table 5.3 we show the results of the reconstruction for a different number of projections, evaluating the results using from 5 to 30 internal iterations of LSQR in steps of 5, giving a tolerance of 1e-06 (input parameters *iter* and *tolerance* in Algorithm 2). If the process does not reach convergence, the maximum number of iterations performed is 10000. For each number of projections we show the three best results, obtained when applying the STF and/or Bilateral filter and/or FISTA every 15, 10 or 5 iterations of LSQR.

Algorithm 2 Iterative reconstruction method

Input: A, g, iter, tolerance, maxiter, bilateral, regularization, acceleration
Output: f

Initialisation :
residual=1, $f = 0_N$, totaliter=0
while residual>tolerance **and** totaliter<maxiter **do**
  f=LSQR(A, g, f, iter)
  residual=$||g - A * f||/||g||$
  if $(residual > tolerance)$ **then**
    if $(bilateral == True)$ **then**
      f=BilateralFilter(f)
    end if
    if $(regularization == True)$ **then**
      f=STF(f)
    end if
    if $(acceleration == True)$ **then**
      f=FISTA(f)
    end if
  end if
  totaliter+ =1
end while
return f

Regardless of the number of projections used, the combination of the four processes is the best option, in some cases doubling the quality of the SSIM metric with respect to the second best option. For this, we conclude that the Bilateral filter is a good contribution to our method. Also, note that the more projections used, the less applications of the regularization and filtering techniques are needed for the reconstruction. This is logical since there is more information available and therefore fewer artifacts are generated. The Bilateral filter improves the quality of the image a greater amount when the number of projections is higher. However, the use of many projections is not desirable since a greater X-ray dose is induced to the patient. In some cases, when we have a high number of projections (180 and 150) the results are very similar using 10 or 15 internal iterations. Here, we opt to choose 15 since it means less applications of the additional steps and thus the reconstruction is faster.

Number of Projections	LSQR iterations	STF	Bilateral filter	FISTA	SSIM	PSNR
180	15	X	X	X	0.9954	51.97
180	15	X	X		0.9849	45.94
180	15	X		X	0.9710	42.50
150	15	X	X	X	0.9935	50.17
150	15	X	X		0.9191	39.57
150	15	X			0.8854	38.69
120	10	X	X	X	0.9840	48.48
120	10	X	X		0.8950	39.01
120	10	X		X	0.8584	37.85
90	10	X	X	X	0.9762	44.53
90	10	X		X	0.7729	34.40
90	10	X	X		0.7667	34.06
60	5	X	X	X	0.9607	38.99
60	5	X		X	0.8052	35.61
60	5	X	X		0.8083	32.46
45	5	X		X	0.7355	31.74
45	5	X	X	X	0.6995	31.10
45	5	X	X		0.6378	30.01
30	5	X	X	X	0.6977	28.87
30	5	X		X	0.6363	27.68
30	5	X	X		0.5888	27.31

Tabla 5.3: Reconstruction results combining the filter.

When we perform more than 15 internal iterations, the results don't improve in any case. We can observe this better in Fig 5.9 where can see the PSNR results for every number of internal iterations when we vary the number of views and combine the different STF, Bilateral and FISTA steps. Here, we observe how the PSNR increases with the number of views, but varies when we change the number of internal iterations. We also show that we obtain higher PSNR values when we use the Bilateral filter.

Although the quality seems low compared with our previous studies with phantom images such as (Chillarón y col. 2017), since this image is much more complex, not all the reconstructions are low quality. In Fig 5.8 we can see the best resulting images for 180, 90, 60 and 30 projections. As we observe, the image for 30 is not good, we see streak artifacts and the internal structures are blurred and distorted. From 60 views, we start getting better images, with al-

most no artifacts. It is not until 90 views that we get less blurry structures and better preserved edges. From 90 to 180 projections, the images get sharper, but the internal structures we see are the same.

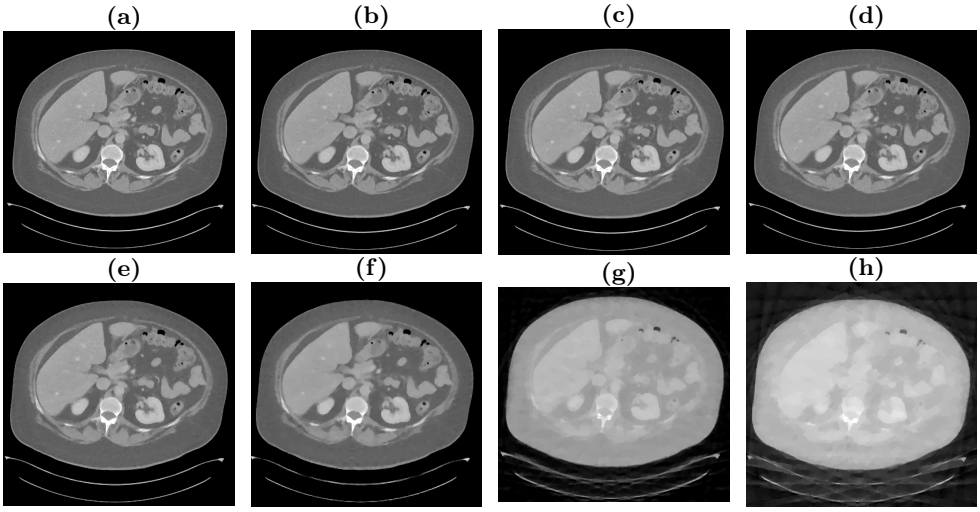


Figura 5.8: Reconstructions of an abdominal CT image. A: Reference CT image selected from dataset DeepLesion. B: Reconstruction using 180 projections. C: 150 projections. D: 120 projections. E: 90 projections. E: 60 projections. E: 45 projections. E: 30 projections.

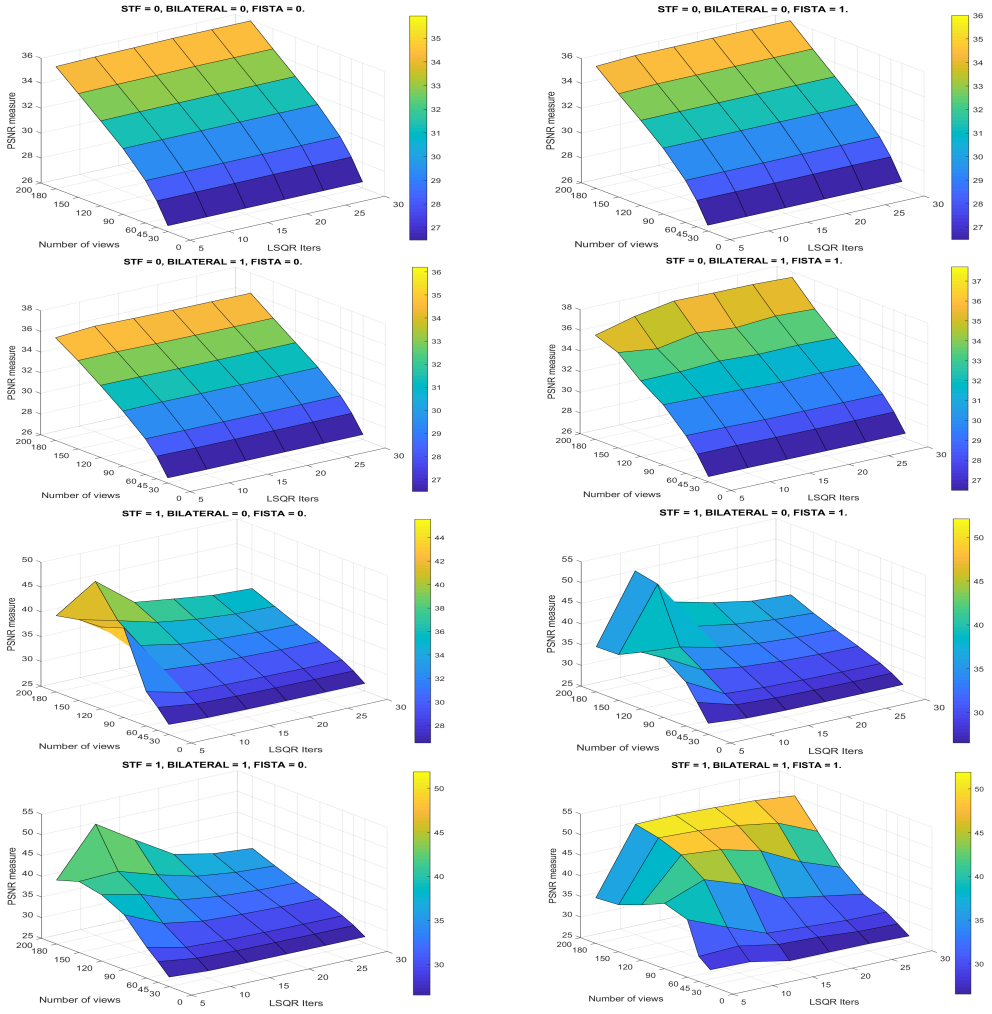


Figura 5.9: Evolution of the PSNR. Results for different number of views and internal LSQR iterations varying the combination of the additional steps.

5.4 Conclusion

In this work, a study of different filtering techniques on a sinogram of 60 views has been carried out. We added noise to the projections of a mathematical phantom, in order to simulate the appearance of artifacts in the acquisition of a real scanner. To evaluate the filters, it has was necessary to select a level

of noise that allows us to see the effect of the filtering techniques. For this reason we chose a 0.0005 variance for both Gaussian and Speckle noise. The best reconstructions are obtained with the sinograms filtered using the bilateral method.

The filtering process has also been carried out on the reference image of the mathematical phantom with a resolution of 256x256 pixels. Again, the bilateral filter has been the one that has obtained better quality values of the filtered images with respect to the reference image for the two types of added noise. The results are significantly better for the speckle noise, improving the SSIM in a 38% and the PSNR in a 13%. As it has been possible to verify, the bilateral filter has the capacity of preserving the contours of all type of forms present in the images, at the same time as it eliminates great part of the noise.

We have also been able to test the validity of the Bilateral filter for its combination with the LSQR+STF+FISTA reconstruction method. The results for 512x512 reconstructions show that integrating the filter within the reconstruction process can improve the final quality of the images.

Supporting information

S1 Dataset. **https://doi.org/10.5281/zenodo.3603080** . The dataset containing projection data used to reconstruct in the last section of the results in this paper is publicly available and they can be downloaded freely from the following permanent location in zenodo.org: `https://doi.org/10.5281/zenodo.3603080`

S2 Dataset. **https://doi.org/10.5281/zenodo.3603104** . The dataset containing resulting reconstructed images in last section of the results in this paper is publicly available and they can be downloaded freely from the following permanent location in zenodo.org: `https://doi.org/10.5281/zenodo.3603104`

Capítulo 6

Estudio de métodos algebraicos directos para reconstrucción de imagen TC

El presente capítulo contiene el estudio preliminar sobre métodos directos aplicado al problema de reconstrucción de TC, publicado en el artículo con referencia (Chillarón y col. 2018) como parte de los *Proceedings* del congreso *International Conference on Computational Science* de 2018, publicados en *Lecture Notes in Computer Science*.

En él se exploran dos nuevos métodos algebraicos directos: QR y SVD. Dichos métodos no han sido explorados hasta al momento para la aproximación de reducción de proyecciones, por lo tanto el objetivo del trabajo consiste en verificar el comportamiento de ambos y estudiar su viabilidad en este problema.

Puesto que la matriz del sistema no es invertible debido a sus grandes dimensiones, es necesario trabajar con la matriz pseudo-inversa, que puede formarse mediante diferentes factorizaciones. Además, mediante las factorizaciones se puede trabajar con matrices del sistema que tengan rango deficiente.

Para el estudio se ha empleado la función SPQR de la librería SuiteSparse, que realiza la factorización QR de una matriz dispersa. Para la descomposición en valores singulares se emplea la librería SLEPc. Las pruebas se han realizado para diferentes resoluciones de imagen (32x32 hasta 512x512) y número de proyecciones empleadas (30, 60 y 90).

En este caso se ha empleado el *cluster* Rigel por la alta demanda de memoria de ambos métodos. En dicho *cluster*, los nodos para ejecuciones con altos requerimientos de memoria están diseñados para paralelismo en memoria compartida, por lo que el sistema de gestión no permite reservar más de un nodo al mismo tiempo para hacer uso de memoria distribuida. Por este motivo, únicamente se ha podido trabajar con 250GB de memoria RAM, tanto para memoria compartida mediante SuiteSparse con paralelismo multihilo a través de BLAS, como para SLEPc, que utiliza paralelismo mediante paso de mensajes en memoria distribuida.

En ambos casos, el número de procesos o hilos utilizados ha sido 32, puesto que cada nodo cuenta con 32 *cores* físicos. Aunque el tiempo de factorización no es crítico, puesto que ésta se realiza una vez y se almacena para posteriores usos, es deseable reducirlo al máximo para evitar posibles errores, así como para tener la capacidad de recalcular las factorizaciones si algún parámetro del escáner cambia.

En cuanto a la gestión de memoria de cada método se puede concluir que el método SPQR logra mejores resultados, ya que consigue realizar la factorización para los casos de resolución 256×256, con 60 y 90 vistas. Para los mismos casos, la SVD falla por falta de memoria, por lo que sólo permite factorizar la matriz de 30 vistas y 256×256 píxeles. El hecho de poder trabajar con una factorización QR sin almacenar la Q explícitamente, ya que se puede resolver el sistema sin emplearla como se muestra en el artículo, permitiría un ahorro extra de memoria que puede hacer la diferencia cuando la cantidad de memoria es limitada.

Por otra parte, se ha verificado que el coste temporal es menor en el caso de la factorización QR, llegando a ser hasta 10 veces menor en algunos casos. Pese a que el coste teórico de las dos factorizaciones es similar (siendo la SVD ligeramente más costosa), la diferencia del tiempo de resolución obtenido puede ser debido al coste de las comunicaciones, que añade un tiempo extra por cada mensaje MPI que sea enviado. Este sobrecoste puede llegar a ser importante en problemas de grandes dimensiones, y habría que realizar un estudio en profundidad sobre el número de procesos óptimos a emplear.

En cuanto a la calidad de las reconstrucciones mediante ambos métodos, se ha determinado que siempre que la matriz del sistema tenga rango completo, las reconstrucciones son óptimas, y no contienen ruido ni artefactos de ningún tipo. Sin embargo, como ha ocurrido para la resolución 256×256 con 30 y 60 vistas, la matriz no es de rango completo, por lo que el problema está mal condicionado. En estas circunstancias, la solución obtenida es una aproximación con calidad muy pobre, siendo mejores las reconstrucciones obtenidas mediante el método SVD ya que se visualizan mejor las estructuras.

Con el fin de poder determinar a priori el número de vistas necesarias para trabajar con matrices de rango completo se ha calculado el rango en cada caso, observando que para resolución la 256×256 es necesario emplear 90 vistas, mientras que para resoluciones menores es suficiente con 30. Para 512×512 píxeles, no se logra llegar a obtener rango completo con 180 vistas, y será necesario emplear al menos 260 como se demostrará en posteriores trabajos.

Pese a no haber podido computar las factorizaciones para la resolución más alta, y haber obtenido reconstrucciones de baja calidad cuando el rango de la matriz es deficiente, este estudio nos proporciona el conocimiento necesario para comenzar a aplicar métodos algebraicos directos en la reconstrucción de imagen TC, con el incentivo de obtener resultados de mucha más calidad que los métodos iterativos cuando las matrices son de rango completo.

Lecture Notes in Computer Science

Volume 10861, 2018

CT medical imaging reconstruction using direct algebraic methods with few projections

Mónica Chillarón[1], Vicente Vidal[1], Gumersindo Verdú[2] and Josep Arnal[3]

1 Departamento de Sistemas Informáticos y Computación (DSIC), Universitat Politècnica de València, Valencia, Spain.
2 Instituto de Seguridad Industrial, Radiofísica y Medioambiental (ISIRYM), Universitat Politècnica de València, Valencia, Spain.
3 Departamento de Ciencia de la Computación e Inteligencia Artificial (DCCIA), Universidad de Alicante, Alicante, Spain.

Abstract

In the field of CT medical image reconstruction, there are two approaches you can take to reconstruct the images: the analytical methods, or the algebraic methods, which can be divided into iterative or direct.
Although analytical methods are the most used for their low computational cost and good reconstruction quality, they do not allow reducing the number of views and thus the radiation absorbed by the patient.
In this paper, we present two direct algebraic approaches for CT reconstruction: performing the Sparse QR (SPQR) factorization of the system matrix or carrying out a singular values decomposition (SVD). We compare the results obtained in terms of image quality and computational time cost and analyze the memory requirements for each case.

Keywords: CT, Medical Imaging, Reconstruction, Matrix factorization, QR, SVD, Few projections.

6.1 Introduction and Background

In medical imaging, CT (computerized tomography) (Kak y Slaney 2001; Radon 1986) is one of the most significant tests to perform a diagnosis. Thus, it is imperative to develop reconstruction algorithms that provide high-quality images as well as high computational time efficiency. Having fast algorithms is essential to have short-term results of a CT scan available to medical professionals.
The reconstruction methods can be divided into two types, depending on their nature. On the one hand, we find the analytical algorithms. They are based

on the application of the Fourier transform on the data obtained from the projection of x-rays on an object, called sinogram. On the other hand, we have the algebraic methods, whether direct or iterative, which make a mathematical approach to the reconstruction problem.
In clinical practice, the most widespread methods are the analytical ones. This is due to the reduced computational time cost involved, which can be vital in emergency diagnoses. However, algebraic methods allow reducing the number of projections, and therefore the radiation to which we expose the patient.
Since hardware elements evolve at high speed and also their cost decreases constantly, using algebraic approaches to solve these problems has become possible. Although they involve a high computational cost, nowadays it is less significant thanks to parallel computing (multiple CPUs, GPU computing, clusters, etc.) (Golub y Ortega 1993). If we add the enhancement of main memory in new equipments, it is now feasible to implement these methods for large size problems as is our case.
The aim of this work is to achieve the resolution of the CT image reconstruction problem by means of two direct algebraic factorization methods. The first one, called multifrontal sparse QR (SPQR), and implemented in the library 'SuiteSparseQR, a multifrontal multithreaded sparse QR factorization package' (Davis 2011), allows solving the equation of reconstruction problem more directly than the iterative methods we studied previously (such as ART (Andersen 1989), SART (Andersen y Kak 1984) or LSQR (Paige y Saunders 1982)). This is achieved by calculating the QR factorization of the sparse system matrix A to simulate its pseudo-inverse.
The second method is the singular values decomposition (SVD) of the system matrix A, which is carried out through the parallel implementation of the method included in the SLEPc (Hernandez, Roman y Vidal 2005) to simulate, as in the previous case, the pseudo-inverse of the matrix.
Since these methods are not widespread in clinical practice, the validity of both approaches will be tested. We will check the quality of the reconstructed image by working with a smaller number of projections in order to reduce the radiation induced to the patient, and analyze the computation resources necessary to perform the decomposition of the matrix A. This point will be vital for the use of these algebraic methods, since the matrices are large and can suppose a high demand for RAM.

6.2 Materials and Methods

6.2.1 CT image reconstruction problem

In a CT reconstruction problem, using an algebraic approach we need to solve the equations system defined by the equation (6.1), where A (6.2) represents the system matrix, a sparse matrix that measures the weight of the influence that each ray has on the reconstructed image (Brooks y Chiro 1976; Hounsfield 1973). The size of the matrix A is MxN, where M is the number of traced rays, and N is the size in pixels of the image to be reconstructed. The vector g (6.3) represents the sinogram or projections vector and u (6.4) the solution image. Both the system matrix and the sinogram vector have been calculated with Joseph's Forward Projection method (Joseph 1982).

$$A * u = g \tag{6.1}$$

$$A = a_{i,j} \in \mathbb{R}^{MxN} \tag{6.2}$$

$$g = [g_1, g_2, ..., g_M]^T \in \mathbb{R}^M \tag{6.3}$$

$$u = [u_1, u_2, ..., u_N]^T \in \mathbb{R}^N \tag{6.4}$$

The projections have been generated for a phantom corresponding to the mathematical representation of the human head developed by the Forbild Phantom Group (FORBILD Phantom Group 2021), using the medical image program CONRAD (Maier y col. 2013) to generate the phantom reference images. This is defined by simple geometric objects that represent head elements with different densities such as bone, tissue, gray matter, etc.

When reconstructing a CT image for a given physical configuration of the scanner, the associated matrix A is always the same. What changes is the sinogram (g), that represents the studied object. If it were possible to calculate a pseudo-inverse A^+ of the matrix A (Katsikis, Pappas y Petralias 2011; Golub y Van Loan 2013), the image could be obtained more directly without solving a large equations system. This means the computational cost would be reduced to a matrix-vector product. But this idea not viable since the explicit calculation of the matrix A^+ is prohibitive for a high resolution and many views in the reconstruction. The reason why is that A^+ can be dense and contaminated due to rounding errors.
Another possibility is to calculate implicitly an approximation of the range and use it to simulate the pseudo-inverse. The two methods that we have con-

sidered to solve this approach are explained in the following subsections.

6.2.2 Projections reduction

In the acquisition of the data, the important factors that influence the reconstruction of the image are two, the number of samples per projection (detectors) and the number of projections. A reduced number of any of these two variables will cause the formation of artifacts in the reconstructed image when using analytical methods: rings by the Gibbs phenomenon, streaks and lines and Moiré pattern, amongst others (Barrett y Keat 2004).
To prevent such problems, we should take into account the Nyquist theorem (Nyquist 1928), which says the sampling rate must be greater than twice the bandwidth of the sampled signal. If we undersample we get the aforementioned artifacts. The classical analytical procedure of image reconstruction (filtered back projection FBP) is a fast process but it needs a complete set of projections to obtain high-quality images. The exact number depends on the physical characteristics of the scanner, but typically the minimum number of projections to take is 360.
If the aim is to reduce the dose absorbed by the patient, we should take the fewer number of projections possible, so we should use different methods that can work with less projections, such as iterative or direct methods. In our previous works (Chillarón y col. 2017; Flores, Vidal y Verdú 2015; Parcero y col. 2017; Flores y col. 2014) it has been shown that it is possible to use few views, between 30 and 90, while using iterative methods to reconstruct high-quality images. However, the computational cost is high for these algorithms.
In order to take an approach that is not iterative, we propose the SPQR and SVD methods. In this way, the computational cost of the reconstruction could be reduced, so we could reconstruct images faster in real time. We will check if its application to the CT image reconstruction allows reducing the number of views in equal measure than the methods that we used previously.

6.2.3 Sparse Pivoting QR factorization

The first new direct method used to solve the equations system is the Multifrontal Sparse QR (SPQR) (Davis 2011). It is a method that performs a QR (Golub y Van Loan 2013) factorization of a large sparse matrix in a sequence of dense frontal matrices. In this way, we can form the pseudo-inverse of the system matrix.
The software used to perform this factorization is implemented in the SuiteS-

parseQR library, which uses BLAS and LAPACK, and Intel Threading Building Block to exploit parallelism. This allows processing heavy matrices, which is key in the reconstruction of CT images since A is large for high resolutions. The QR factorization of the matrix A comprises its decomposition as a product of two matrices (6.5), where Q is orthogonal and R is upper triangular.

$$A = QR \tag{6.5}$$

$$AP = QR \tag{6.6}$$

However, when A is considerably large and sparse, this decomposition can be prohibitive since Q could not be sparse. Here, the decomposition with pivoting (6.6) must be used. The resolution would be:

- If $m>=n$ we have to solve (6.7). That means a matrix-vector product of Q^T by the vector g, the resolution of an upper triangular system with the matrix R and a rearrangement with permutation matrix P.
- If $m<n$ we have to solve (6.8), that is reordering the vector g, then solve a triangular system and finally make the product of the resulting vector by the matrix Q .

$$u = P * (R^{-1}(Q^T * g)) \tag{6.7}$$

$$u = Q * (R^{-T}(P^T * g)) \tag{6.8}$$

6.2.4 Q-less Sparse QR factorization

If we want to spare memory, we can compute the QR decomposition without saving the Q explicitly. In this way we would have to transform the equations to obtain the solution vector u, which would be as follows:

- If $m>=n$ we have to solve the system $Q*R*g = u$. By multiplying both sides of equation by A^T, we obtain: $(Q*R)^T*Q*R*g = A^T*u$. Since Q is orthogonal, performing the matrix operations to solve for u, we reach the solution equation (6.9).

$$u = R^{-1} * R^{-T}(A^T * g) \tag{6.9}$$

- If $m<n$ we would work with the QR decomposition of A^T. By a similar reasoning to the previous case, the solution of the system would corres-

pond to calculate (6.10), which requires the same operations but in a different order.

$$u = A^T * R^{-1} * (R^{-T} * g) \tag{6.10}$$

Note that the Q matrix is not used for the calculation of u and therefore this process requires fewer memory resources.

6.2.5 Singular Values Decomposition

The singular values decomposition performs the k-order factorization (6.11) (Berry, Pulatova y Stewart 2005; Katsikis, Pappas y Petralias 2011; Golub y Kahan 1965). In this factorization, $U \in \mathbb{R}^{Mxk}$ and $V \in \mathbb{R}^{kxN}$ are matrices whose columns are orthogonal and are called left-hand and right-hand singular vectors, where k is the number of singular values computed. $\Sigma \in \mathbb{R}^{kxk}$ is a diagonal matrix that has the singular values in decreasing order. The software we use to apply this algorithm is the SVD solver included in the eigenvalues calculation parallel library SLEPc. Applying this factorization, we can transform the equations system into the product (6.13), taking the pseudo-inverse as (6.12).

$$A = U\Sigma V^T \tag{6.11}$$

$$A^+ = V\Sigma^{-1}U^T \tag{6.12}$$

$$u = A^+ g \tag{6.13}$$

Note that the storage cost of this decomposition is of order *(M*k) + (k*N)* elements. When k grows, the memory resources necessary to carry out the decomposition increase. Once the decomposition of the system matrix has been calculated, resolving the problem consists on the product of two dense matrices by a vector,thus the computational cost is of order *((M + N)*k)* flops.
Since we can do this decomposition ′offline′ , it can be calculated and stored and it can be ready when a CT image has to be reconstructed. Therefore, we can replace an iterative method with a much simpler and less computationally expensive matrix multiplication problem. In our application, we are interested in the case where $k = N$, in which case we have enough information to reconstruct the image with great precision.

6.3 Results and Discussion

In order to check the validity of both methods, we used a cluster belonging to the Universitat Politècnica de València to perform the factorizations. The cluster consists of 4 RX500S7 servers with four Intel Xeon E5-4620 8 core processors (32 cores per node) and 256GB DDR3 RAM (8GB / core ratio) per node.

6.3.1 Memory requirements

For the SPQR factorization, we have used the code developed for Matlab. We reserve a single node in the cluster. In this node which we can consume a maximum of 250GB of main memory. With these resources, it has been possible to compute the QR decomposition corresponding to the system matrix for an image resolution from 32x32 up to 256x256 with 30 views. Besides, using the Q-less QR factorization we have been able to use up to 90 views with the 256x256 resolution, which is important as we will explain in section 3.3. For the analogous case of resolution 512x512, considering 30 views, the matrix has not been factorized due to lack of RAM.
Regarding the SVD decomposition, all cases of 30 views have been computed, except for the 512x512 resolution case. For this process, we have reserved up to 32 processors with 15GB of RAM per processor, which means up to 240GB. Despite making use of distributed memory, we can not reserve over one server node at a time, similar to the sequential case. In addition, the SVD process requires more memory than the SPQR, so with the same characteristics, it has not been possible to reach the case of 90 views for the 256x256 resolution.

6.3.2 Computational time efficiency

Although the factorization time is not critical in this scenario, it is convenient to have an approximation of the computation time required for each matrix size. In addition, we want to make a comparison between the two proposed methods. Table 6.1 shows the results in seconds of computational time cost to carry out the decompositions.
As we can observe, the factorization time in both cases is satisfactory. Although it may seem that with the SVD decomposition it is too high, we must bear in mind that this calculation will be performed only once and 'offline', which makes it a non-critical calculation. As shown in the table, the computational time efficiency of the SPQR method is greater, which can be justified because this calculation is performed in a single node, and therefore does not require dis-

	Factorization Method	
	SPQR	SVD
Resolution	Time (secs.)	
32x32 30 views	1.2	9
64x64 30 views	5.6	50
128x128 30 views	151	1900
256x256 30 views	3270	12000
256x256 60 views	12336	-
256x256 90 views	16400	-

Tabla 6.1: Factorization time.

	SPQR		SVD	
Resolution	PSNR	SSIM	PSNR	SSIM
32x32 30 views	228	1	238	1
64x64 30 views	213	1	224	1
128x128 30 views	255	1	221	1
256x256 30 views	38	0.03	30	0.29
256x256 60 views	32.5	0.03	-	-
256x256 90 views	150	1	-	-

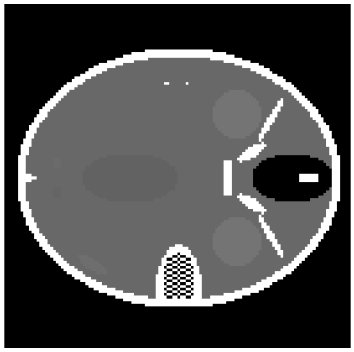
Tabla 6.2: Reconstructed images quality.

tributed memory as does the SVD. Since SVD implements the communication through MPI messages, it generates an extra time associated with interprocess communications, which makes the temporary efficiency much worse.

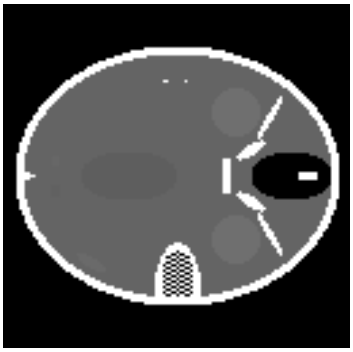
6.3.3 Image Quality

Regarding the quality of the images obtained by performing a reconstruction with both methods, the results are reflected in Table 6.2. To measure the quality, we used the PSNR metric (Hore y Ziou 2010), which shows the noise level of a image compared to a reference image, and also the SSIM metric (Hore y Ziou 2010), which indicates the structural similarity between both images, based on the shape of the elements it contains.

In both cases we can see that up to the resolution 128x128, the results of the reconstruction are very good, getting a PSNR greater than 200 in all cases (considering that a PSNR close to 100 can be considered a perfect reconstruction to the perception of the eye human). In addition, all cases obtain an SSIM equal to 1, which means that structurally the image is accurate. The result of

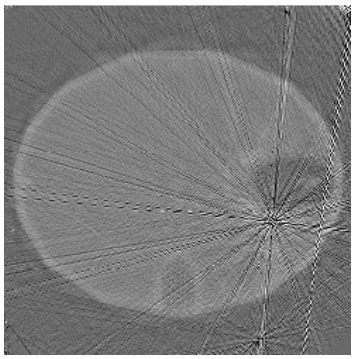


(a) Reference image 128x128

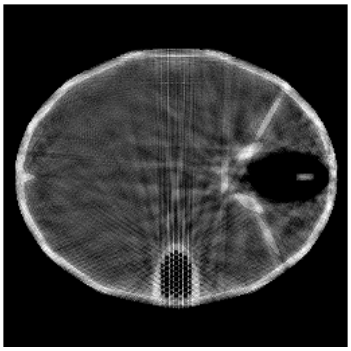


(b) SVD reconstruction 128x128 30 views

Figura 6.1: Reference and reconstructed 128x128 images.



(a) SPQR reconstruction



(b) SVD reconstruction

Figura 6.2: Reconstructions 256x256 30 views.

the reconstruction by SVD with resolution 128x128 can be seen in Fig. 6.1b. If we compare it with the reference image for the same resolution (Fig. 6.1a), we observe that they are almost identical and the reconstruction does not have artifacts or noise.
However, for the 256x256 resolution and 30 views, the result is of very poor quality for both methods, being the PSNR of both around 30. The reconstructed images, that we can see in Fig. 6.2 are very noisy and although we get to perceive elements of the phantom, it is not seen clearly enough.

The poor quality of this reconstruction is due to the fact that for this particular case (256x256 and 30 views), the sub-matrix extracted from the system matrix

	Number of views				
	180	120	90	60	30
Resolution	Nº of columns / Rank				
32x32	1024/1024	1024/1024	1024/1024	1024/1024	1024/1024
64x64	4096/4096	4096/4096	4096/4096	4096/4096	4096/4096
128x128	16384/ 16384	16384/16384	16384/16384	16384/16384	16384/16384
256x256	65536/65536	65536/65536	65536/65536	65536/49380	65536/30093
512x512	262144/149551	262144/100842	262144/76000	262144/ 50963	262144/25608

Tabla 6.3: Rank study.

is rank deficient, as shown in Table 6.3. Therefore, the equations system is ill-conditioned, which leads to missing information necessary for the reconstruction. We observe that in the rest of the cases with smaller resolutions the sub-matrix retains the full rank, so we have no problem when reconstructing using these algebraic methods.
If we analyze the rank of the sub-matrices when we work with the largest resolutions, we can observe that we need to use more views in order to reach full rank. For 256x256 pixels reconstructions, we should use at least 90 views, which means we still reduce the number of views with respect to traditional methods. For 512x512 resolution, we need over 180 views, approximately 260, to keep the full rank on the sub-matrix. That means that even if we could still reduce a few views compared with other methods, the resulting reconstruction problem would have larger dimensions than we can compute.

We have verified that the rank of the matrix used has direct effect on the reconstructed image. In Fig. 6.3 we present the singular vector of a full-rank sub-matrix. As we can see, we have a few dominant values, approximately 1000, and then they start to decrease. When we reach the 8000th value, it looks like they stabilize and are close to 0. If we look to the detail window on the plot, we can observe that the last 1600 values, even if they are close to 0, vary between 0.03 and 0.01, which is still a significant number in this case. Therefore, this means we can not disregard any singular value.

In Fig. 6.4 we can observe how varying the number of singular values, we get very different reconstructions. As explained before, we do not reach optimal quality until we use the full rank. We observe the same effect with SPQR if we vary the number of views.

In Fig. 6.5 we see the difference between taking 30, 60, and 90 for the 256x256 resolution matrix. With 60 projections we increase the rank, so the image is slightly better than with 30, as can be observed in the edges of the phantom. But it is not until we reach 90 views that the image is of good quality.

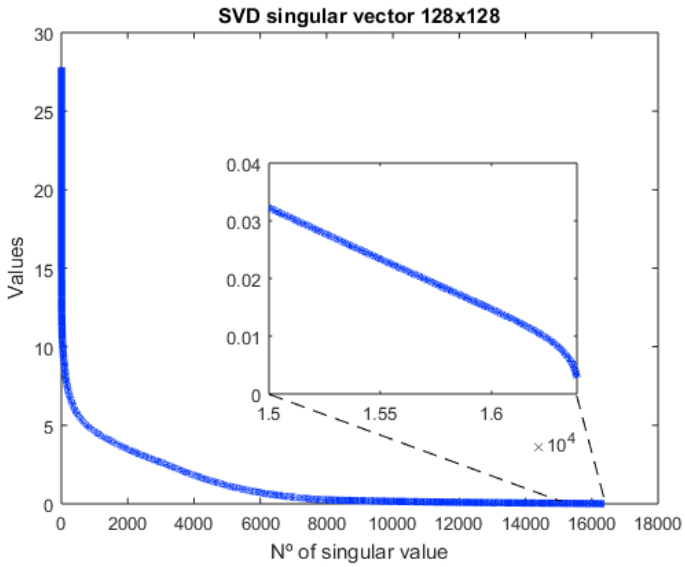
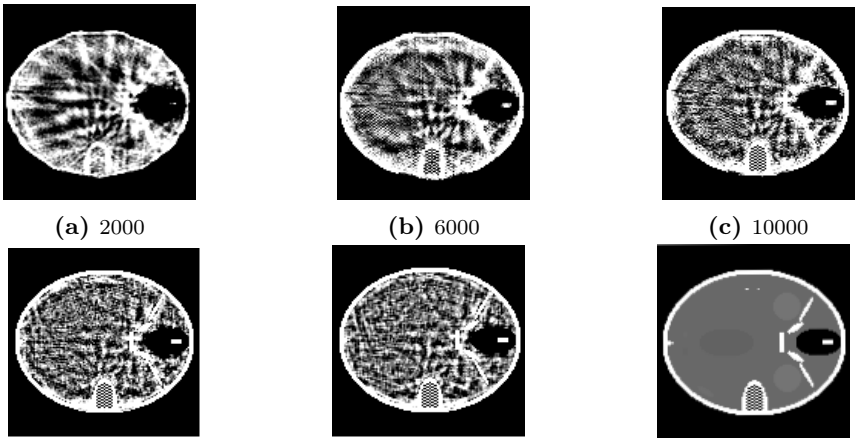


Figura 6.3: Singular vector 128x128.

6.4 Conclusion

In the present work, we have proposed two sparse matrix factorization methods that can be used for CT image reconstruction. In this way, we simulate the use of the pseudoinverse of the system matrix to transform the initial problem solution into a simpler one. The main challenge of these methods is the high consumption of memory to perform the computation, so we had to make use of a high performance computing cluster.
In this cluster it has been possible to calculate the SPQR factorization up to a resolution of 256x256 pixels and 90 views. In addition, the SVD up to 256x256 and 30 views. Although with the SVD method we have not obtained satisfactory results for higher resolutions, through SPQR we have managed to perform a reconstruction of very high quality with 90 views and full rank. This translates into a very significant difference in the doses of x-rays to which patients should be exposed. Taking into account that the direct methods are based on the Nyquist theorem, for our scanner configuration, they would use significantly more views. However, we are reducing them to 90 in the case of resolution 256x256 and to 30 in lower ones.
Since the calculation of the factorizations for this type of reconstruction problem can be done before the moment of the reconstruction itself and stored,

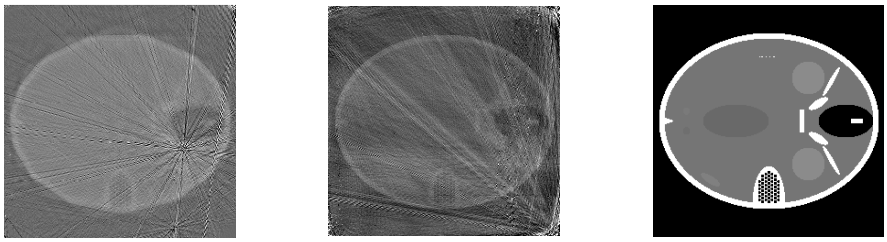
the computation times required for all cases are acceptable, the SPQR method being faster because it does not need distributed memory.
In addition, we have transformed the problem of reconstruction into a problem of matrix-vector multiplication and resolution of a triangular system in the case of the QR and a matrix-vector product in the case of the SVD. These resolutions are highly parallelizable in both CPU and GPU, which means that having the matrices stored we could accomplish the reconstructions faster than with the previous methods.
Regarding the image quality obtained, it is very satisfactory for all the cases in which the sub-matrix generated is full range, obtaining images of higher quality than those obtained by iterative methods that we have presented in previous works, such as LSQR. However, the moment the sub-matrix becomes rank deficient, the reconstructed image is of very poor quality due to the lack of information. In future works, it is proposed to introduce regularization algorithms to increase the range of sub-matrices, or a filter in the image that, as with the LSQR method, are combined in a certain way that allows a more quality approximation to be made despite having an ill-conditioned problem.
In addition, it is necessary to analyze the implementation of the factorizations in order to improve the use of RAM memory, and in this way to factorize the corresponding matrix for 512x512 pixels, which is the objective resolution. For this problem, we set out to use out-of-core computing techniques that make use of the disk to avoid main memory problems.
In conclusion, we can say that we have tested the viability of the SPQR and



(a) 2000 **(b)** 6000 **(c)** 10000

(d) 14000 **(e)** 15300 **(f)** 16384

Figura 6.4: Reconstruction varying singular values 128x128.



(a) 30 views **(b)** 60 views **(c)** 90 views

Figura 6.5: SPQR 256x256 reconstructions.

SVD direct algebraic methods applied to CT image reconstruction. In spite of not having reconstructed images of very high quality, we have verified that with these methods it is possible to reduce the dose of radiation to a great extent if the matrix can be computed.
At this point of our work we have two resolution options: For rapid reconstructions with low dose and medium resolutions, factoring methods. For high-resolution reconstructions and slightly worse quality, the LSQR + FISTA + STF iterative method, which is slower but guarantees good results.

Acknowledgements

This research has been supported by "Universitat Politècnica de València" and the Spanish Ministry of Economy and Competitiveness under Grant TIN2015-66972-C5-4-R co-financed by FEDER funds, as well as "Generalitat Valenciana" under PROMETEOII/2014/008 project and ACIF/2017/075 predoctoral grant.

Capítulo 7

Análisis del método QR mediante la librería SuiteSparse: escalabilidad y calidad.

En este capítulo se presenta un estudio en profundidad sobre el método QR para reconstrucción la de imagen TC mediante la librería SuiteSparse. Este estudio se publicó como parte de los *Proceedings* del congreso *International Conference on Computational Science* de 2019 publicados en *Lecture Notes in Computer Science*, y su referencia completa es (Chillarón, Vidal y Verdú 2019).

Partiendo de los resultados mostrados en el capítulo anterior se escogió el método QR con pivotamiento como método de reconstrucción de tipo algebraico directo por su mejor gestión de memoria y menor coste temporal en los recursos hardware disponibles en el momento del estudio. De este modo, se procedió al análisis de la eficiencia del paralelismo proporcionado mediante BLAS, tanto para la factorización como para la resolución del sistema, lo cual no había sido medido en el trabajo anterior. Además, se plantean dos estrategias de resolución: empleando la matriz Q formada explícitamente, o trabajando con las reflexiones de Householder para evitar formar la Q, almacenando únicamente los vectores de Householder.

Para estudiar el comportamiento del método en ambos casos se ha empleado el *cluster* Rigel, utilizando un único nodo como en el estudio anterior. Además, puesto que ya se había resuelto exitosamente con anterioridad, se va a resolver únicamente el caso de resolución 256×256 y 90 vistas, ya que es de rango completo.

Se ha comparado el tiempo de factorización mediante ambos métodos, variando el número de hilos de BLAS empleados, utilizando potencias de 2 hasta 32, que es el número de *cores* físicos de la máquina. En los resultados se observa que es alrededor de 3 veces más rápido factorizar la matriz trabajando con las reflexiones de Householder, lo que es una reducción considerable puesto se reduce en horas el tiempo necesario para obtener la factorización de la matriz.

En cuanto a la escalabilidad, se ha determinado que mediante las reflexiones de Householder se obtienen *SpeedUps* mayores que calculando la Q explícitamente, excepto en el caso de 32 hilos en el que ocurre lo contrario. Sin embargo, en ninguno de los dos casos el *SpeedUp* se acerca al ideal, siendo la eficiencia menor de 0.5 cuando se emplean más de 4 hilos. Para este problema, aumentar el número de hilos no tiene un impacto muy positivo en cuanto a aprovechamiento de recursos se refiere.

Por otra parte, también se ha estudiado la escalabilidad para el paso de reconstrucción con ambas estrategias. Además, en este paso se han considerado dos supuestos diferentes: resolver un solo lado derecho o corte, y resolver 128 lados derechos en la misma ejecución.

En los resultados se observa que el tiempo de reconstrucción es ligeramente peor mediante las reflexiones de Householder cuando se resuelve un único corte, independientemente del número de hilos empleados, siendo aproximadamente un minuto más lento en el peor de los casos que empleando la Q explícita. Sin embargo, al aumentar el número de lados derechos a 128, los resultados cambian. En este caso, las reconstrucciones con las reflexiones Householder son entre 1.75 y 2.8 veces más rápidas, dependiendo del número de hilos.

En cuanto a la eficiencia, se ha comprobado que el nivel de paralelismo logrado en ambos casos es muy bajo, obteniendo un *SpeedUp* máximo de 1.35 para un corte mediante Q explícita, y de 1.4 para 128 cortes mediante las reflexiones de Householder. De este modo, la eficiencia se sitúa por debajo de 0.5 en todos los casos cuando se emplean más de 2 hilos, llegando a ser de aproximadamente 0.044 para 32 hilos, lo cual implica un aprovechamiento prácticamente nulo de los recursos empleados.

Por otra parte se ha estudiado la calidad de las imágenes obtenidas para determinar si existen diferencias según el método empleado para la factorización. Para ello, se ha reconstruido una imagen de TC real correspondiente a un corte de un abdomen, observando que ambas reconstrucciones son de muy alta calidad. Sin embargo, mediante la métrica MAE se observa un menor error en las reconstrucciones obtenidas con las reflexiones de Householder, y obtiene un PSNR más alto lo que implica menos ruido en la imagen. Pese a que estas diferencias son mínimas y no son apreciables por el ojo humano, son lo suficientemente significativas como para determinar que el método que utiliza reflexiones de Householder es más estable numéricamente y por lo tanto preferible.

Como conclusiones de este análisis sobre el método QR se determina que emplear las reflexiones de Householder siempre es más positivo que emplear la Q de manera explícita, por obtener un tiempo de factorización mucho menor, menos requerimientos de memoria, mayor estabilidad numérica así como reconstrucciones más eficientes para múltiples lados derechos.

Con todo ello, el método QR se posiciona como una buena alternativa a los métodos iterativos cuando lo que se pretende es priorizar la calidad por encima del porcentaje de dosis reducido. Sin embargo, es necesario emplear nuevas estrategias que permitan aumentar el tamaño del problema para poder trabajar con resoluciones más grandes.

Lecture Notes in Computer Science

Volume 11538, 2019

Parallel CT reconstruction for multiple slices studies with SuiteSparseQR factorization package

Mónica Chillarón[1], Vicente Vidal[1] and Gumersindo Verdú[2]

1 Departamento de Sistemas Informáticos y Computación (DSIC), Universitat Politècnica de València, Valencia, Spain.
2 Instituto de Seguridad Industrial, Radiofísica y Medioambiental (ISIRYM), Universitat Politècnica de València, Valencia, Spain.

Abstract

Algebraic factorization methods applied to the discipline of Computerized Tomography (CT) Medical Imaging Reconstruction involve a high computational cost. Since these techniques are significantly slower than the traditional analytical ones and time is critical in this field, we need to employ parallel implementations in order to exploit the machine resources and obtain efficient reconstructions.
In this paper, we analyze the performance of the sparse QR decomposition implemented on SuiteSparseQR factorization package applied to the CT reconstruction problem. We explore both the parallelism provided by BLAS threads and the use of the Householder reflections to reconstruct multiple slices at once efficiently. Combining both strategies, we can boost the performance of the reconstructions and implement a reliable and competitive method that gets high-quality CT images.

Keywords: CT, Medical Imaging, Reconstruction, Matrix factorization, QR , Few projections, Parallel QR, SuiteSparseQR.

7.1 Introduction and Background

In recent years, medical tests such as Magnetic Resonance Imaging (MRI) (Brown y col. 2014) have gained prominence in clinical practice. MRIs are not harmful to the patient, since the image is produced from the application of magnetic fields on a body. In contrast, Computerized Tomographies (CT) (Brooks y Chiro 1976) project X-rays, which induce a dose of radiation that can be harmful to the patient. However, despite being harmful, CT scans are still necessary.

On the one hand, they obtain better images than the MRI for certain types of objects of interest (bones and tumors) while the magnetic resonance is mostly applied to soft tissues since it achieves greater contrast between different tissues. On the other hand, not all people are suitable for both tests. MRI is not recommended for patients who have a pacemaker or metal implants in their body, while they can undergo a CT scan. But CT is contraindicated for pregnant patients, infants and children, due to the dose of radiation induced with the test.

For all the above, although the current perception is that the CTs are losing ground to the MRI, it is not entirely true, so we believe that it is necessary to continue improving CT scanners and also the techniques of image reconstruction they use.

In our previous works (Flores, Vidal y Verdú 2015; Parcero y col. 2017; Flores y col. 2014; Chillarón y col. 2017; Chillarón y col. 2018), we have studied the option of working with algebraic methods instead of the traditional analytical methods to reconstruct CT images. In this way, we can solve the problem mathematically by either iterative or direct algebraic algorithms. While other works focus on reducing radiation by applying X-rays with lower voltage (Padole y col. 2015), we focus on taking fewer shots. With this approach, we can work with a low number of projections and still get high-quality images. This means that we could reduce the radiation dose to which the patient is exposed, which is our main objective.

Using iterative methods, we can reduce the number of views or projections that we use to a really small number if we make a good selection of the projection angles (Andersen y Kak 1984; Yu y Zeng 2014; Chillarón y col. 2017). However, with the direct methods (Rodríguez-Alvarez y col. 2018; Chillarón y col. 2018), we have reached the conclusion that it is necessary that the matrix of the CT system has full rank, which will determine the number of projections required according to the image resolution that you want to achieve. Regardless, both approaches need a lower number of X-ray projections than other methods.

However, the types of methods we use require more computational resources. In addition, the time needed to reconstruct the images is much higher than with the analytical methods (minutes or hours versus milliseconds). Therefore, if we want the algebraic approach to be employed, we have to reduce the reconstruction time to the maximum. For this we can use High-Performance Computing (HPC) techniques, exploiting the hardware resources of the machine through parallel implementations of the algorithms.

In this paper, we focus on the analysis of the performance of the QR (Golub y Van Loan 2013) factorization employed to reconstruct CT images. To do this, we make use of the SuiteSparseQR factorization package (Davis 2011), analyzing the effect of using a different number of BLAS threads in operations. In addition, we compare the time efficiency when we reconstruct a single slice, or when we have a volume formed by multiple slices, which fits best a real situation.

In Section 7.2.1, we will describe the CT image reconstruction problem. We also make a brief description of the scanner we simulate and the dataset DeepLesion, used to take as reference. In Sections 7.2.2 and 7.2.3 we explain how to perform the CT reconstructions using the explicit QR factorization or the Householder form. The results of the study are discussed in Section 7.3, analyzing on the one hand the performance of the QR factorization using a different number of threads (Sect 7.3.1) as well as the performance of the reconstruction step (Sect 7.3.2). Additionally, in Section 7.3.3 we show the obtained images measuring their quality. Finally, in Section 7.4 we summarize the work done and discuss the future lines of work.

7.2 Materials and Methods

7.2.1 Algebraic CT image reconstruction

When dealing with the reconstruction of CT images in an algebraic way, it is necessary to model the associated problem. Initially, we only have the data obtained through the scanner using X-rays. Therefore, it will be necessary to transform this data into an image that represents the projected object or body part.

We pose the problem as a system of linear equations as proposed in equation (7.1). Here, g is the data acquired by the scanner. It is usually called projections vector or sinogram for fanbeam CTs. As we see in (7.2), g is a vector of size M, which depends on the physical parameters of the scanner. We calculate M as the product of the number of detectors of the CT and the number of projections taken. It is worth mentioning than one projection (one X-ray shot) obtains as many data as detectors we have.

The system matrix A is defined in equation (7.4). This is a sparse weight matrix that represents the influence a of each ray beam traced (i) on each image pixel (j). As we can see, the number of rows of A is M, and the number

of columns is N, which is the resolution in pixels of the reconstructed image we want to get.

In our case, both the matrix and the projections vector is simulated. We calculate both using the forward projection ray-tracing algorithm proposed by Joseph (Joseph 1982), simulating a CT scanner with 1025 detectors and taking equiangular projections around the 360 degrees of rotation. The images we projected are a selection of the DeepLesion dataset (Yan y col. 2018), which contains thousands of CT images of numerous patients for the study of different types of lesions.

Finally, u is the solution of our equations system. It is the reconstructed image. If we store in vector form, as we see in (7.3), its size is the total number of pixels on the image. For instance, for a final image of resolution 256×256 pixels, u is a vector of size 1×256^2. It is very easy to go from vector form to image form and vice versa.

$$A * u = g \tag{7.1}$$

$$g = [g_1, g_2, ..., g_M]^T \in \mathbb{R}^M \tag{7.2}$$

$$u = [u_1, u_2, ..., u_N]^T \in \mathbb{R}^N \tag{7.3}$$

$$A = a_{i,j} \in \mathbb{R}^{MxN} \tag{7.4}$$

7.2.2 QR Factorization applied to the CT problem

Ideally, the previously modeled problem could be solved very simply by obtaining the inverse of the system matrix as shown in the equation (7.5), provided it had full rank and the matrix was square. But in our application, this matrix dimensions can be really large for the highest resolutions and rectangular (more rows than columns). It is not feasible to explicitly compute the inverse since it requires a high computational cost and really advanced hardware. Besides, it is a highly unstable computation, so the errors could spoil the resulting image.

For this reason, we need to solve the problem with an iterative method as we did in (Flores, Vidal y Verdú 2015; Chillarón y col. 2017) or apply a factorization to the matrix so we can solve it directly. The factorization we propose is the QR with pivoting (Golub y Ortega 1993), as described in (7.6). Here, Q is an orthonormal matrix (its columns are orthogonal unit vectors so $Q^tQ = I$). R

is an upper triangular matrix and P is the permutation matrix used to reduce the filling.

With this decomposition done, we can emulate the inverse of the matrix A as shown in (7.7), or the pseudoinverse if the matrix is non-invertible (Quintana y col. enviado; Hansen 2000). Now we can solve the problem as (7.8). And since the factorization can be performed only once and stored for future use, every time we need to reconstruct we will only have to perform a matrix-matrix product, a permutation and solving one upper triangular equations system. This means faster reconstructions.

In addition, here g can be a matrix $\mathbb{R}^{MxS}$ with S columns (the number of slices we want to reconstruct), so we can get multiple images within the same operation, which also reduces the time per slice.

$$u = A^{-1} * g \tag{7.5}$$

$$A * P = Q * R \tag{7.6}$$

$$A^{-1} = PR^{-1}Q^T \tag{7.7}$$

$$u = P * (R^{-1}(Q^T * g)) \tag{7.8}$$

7.2.3 Q-less factorization using Householder reflections

As we mentioned, our matrices can get relatively big depending on the desired image resolution, so it is possible the computational resources needed to compute the decomposition are extensive. Even if the system matrix A is sparse, Q can be dense. As a way to spare main memory resources, we could decide to not calculate the Q matrix explicitly. Instead, we could perform the factorization using Householder reflections, and store only the set of Householder vectors, which describe transforms to be applied as shown in (7.9). Here H_i are the successive reflection matrices to apply to our right-hand side g. Apart from main memory, this technique will also reduce the computation time.

$$Q = H_1 H_2 \cdots H_{N-2} H_{N-1} \tag{7.9}$$

7.2.4 SuiteSparseQR factorization package

SuiteSparseQR (Davis 2011) is an implementation of the multifrontal sparse QR factorization method. It uses both BLAS and Intel's Threading Building Blocks, a shared-memory programming model for modern multicore architectures to exploit parallelism. The package is written in C++ with user interfaces for MATLAB, C, and C++. It works for real and complex sparse matrices.

In our case, we are using the MATLAB interface, making use of BLAS parallelism and working with real sparse matrices.

7.3 Results and Discussion

In order to test the performance of the method we run both the matrix factorization and the reconstructions in a server with four Intel Xeon E5-4620 8c/16T processors (8 cores/processor, 32 cores/node) and 256GB DDR3 RAM memory (ratio 8GB/core).

We use a matrix corresponding to resolution 256x256 pixels and 90 projections. Thus, the size of the matrix is 90200x65536, with 40871478 non-zero elements and 0.0069 density. In the next subsections we show the experimental time results when using a different number of BLAS threads for the two alternatives of the factorization method, using one physical processor per thread.

7.3.1 Factorization step

In the fist step we need to compute the factorization shown in equation (7.6). As explained in Section 7.2.2, the factorization of the matrix can be done once before the reconstruction of the images. Therefore, the computational speed is not going to be that important in this case. Even so, it is desirable to reduce it to the maximum, both to save computing time and to avoid possible system failures that can abort our process and spoil hours of work.

As we can see in Table 7.1, it is much faster to store the factorization in Householder form than to form the Q matrix explicitly. Regardless of the number of BLAS threads employed, it is around 3 times faster, which is a significant difference.

In Figures 7.1 and 7.2 we can see the Speedup and Efficiency of both of the methods. We compute the Speedup for p processors as $S_p = T_1/T_p$, being T_1 the time with 1 processor and T_p the time with p processors. The Efficiency is

$E_p = S_p/p$. In a perfect parallel algorithm, the Speedup is equal to the number of processors and the Efficiency is 1, which means we are taking advantage of the 100% of the resources.

However, we can observe that we get lower results. The Householder factorization has slightly better performance except when using 32 processors, with a Speedup of 8 versus the 10.2 of the explicit Q factorization. In Figure 7.2 we can see that with more than 4 processors we are using less than a 50% of the computational resources. Since we are working with sparse matrices with a low density it is usual to get lower efficiency that with dense matrices.

	Factorization Method		
BLAS threads	Explicit Q	Householder	Improvement Factor
	Time (secs.)		
1	172784	57628	3.00
2	119407	31955	3.74
4	72186	21105	3.42
8	56325	18481	3.05
16	36451	10422	3.50
32	16805	7191	2.34

Tabla 7.1: Factorization time.

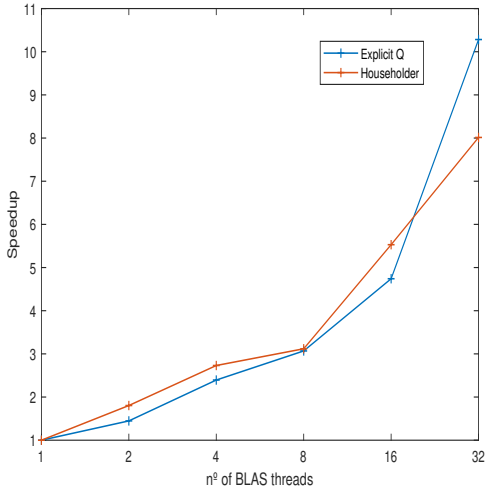


Figura 7.1: Factorization Speedup.

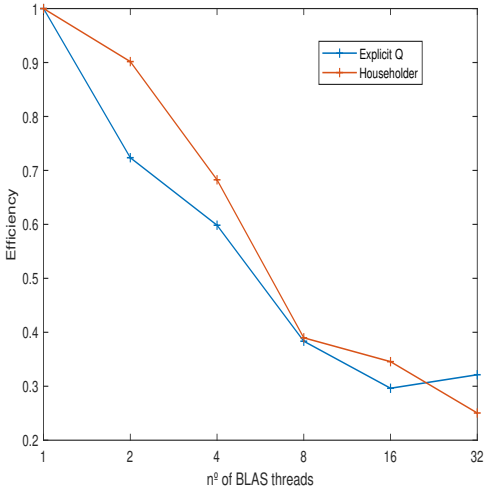


Figura 7.2: Factorization Efficiency.

7.3.2 Reconstruction step

We also need to verify which method is better in the reconstruction phase and if we get good performance when using more resources. In Table 7.2 we observe the results for reconstructing just 1 slice (1 right-hand side vector) or 128 slices, as per equation (7.8). As we can see, when we want to reconstruct only 1 slice, the Explicit Q factorization gets lower computational times. Besides, they improve slightly when using more processors, up to 16. With the Householder form, we only get better performance with up to 8 processors. However, the performance here is worse than in the factorization step, as we can see in Figures 7.3 and 7.4. The Speedup is very low regardless of the number of threads used, and the highest efficiency we get is only 0.5 using 2 threads.

When we are dealing with multiple right-hand sides, in this case 128, the performance is different. We can see in the table that in this case, it is faster to perform the reconstruction using the Householder reflections. We get the reconstructions around twice as fast (2.8 times faster for 16 threads). On the other hand, the Speedup and Efficiency for each of the methods is not better than in the previous case, as we can see in Figures 7.5 and 7.6. We get very low Efficiency, wasting resources.

	1 Slice		128 Slices	
BLAS threads	Explicit Q	Householder	Explicit Q	Householder
	Time (secs.)			
1	239	281	2432	1392
2	226	272	2370	1381
4	221	256	2351	1370
8	210	232	2638	1296
16	177	253	2539	910
32	222	269	2607	1111

Tabla 7.2: Reconstruction time.

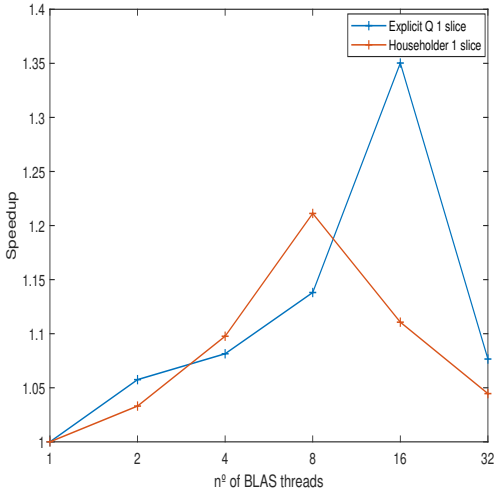


Figura 7.3: 1 slice rec. Speedup.

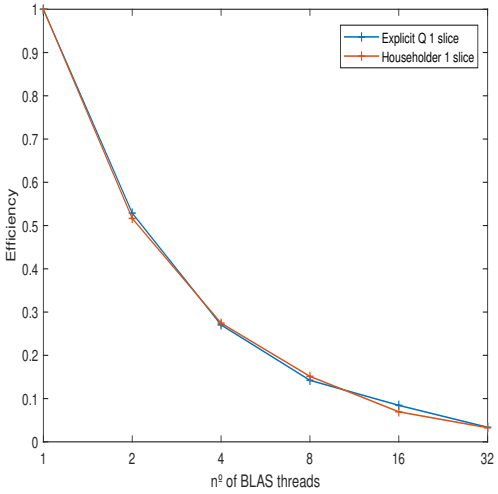


Figura 7.4: 1 slice rec. Efficiency.

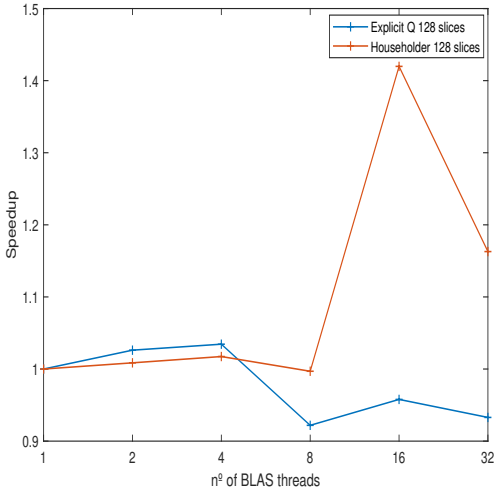


Figura 7.5: 128 slices rec. Speedup.

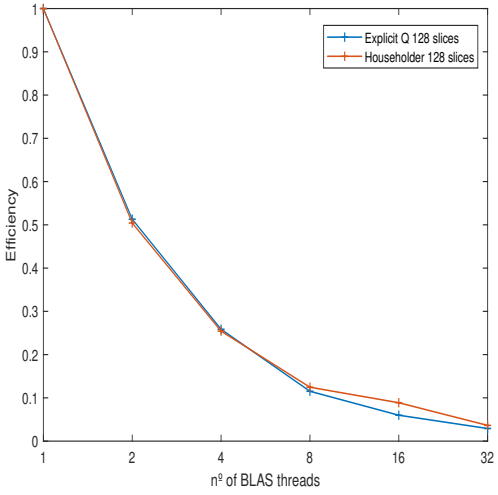


Figura 7.6: 128 slices rec. Efficiency.

7.3.3 Image Quality

In our preliminary work (Chillarón y col. 2018) we concluded that if we work with a full-rank system matrix, the images reconstructed by the QR factorization have really high quality. In order to verify this for the matrix we are studying here, we show the quality results, measuring with Mean Absolute

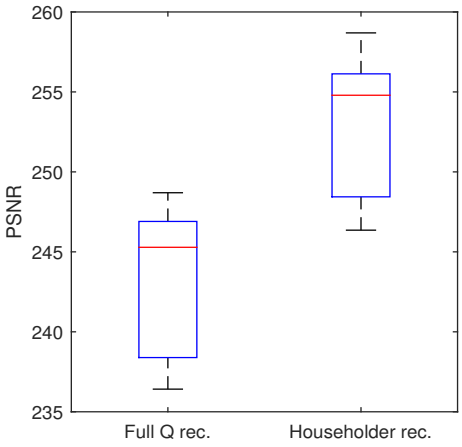


Figura 7.7: PSNR results.

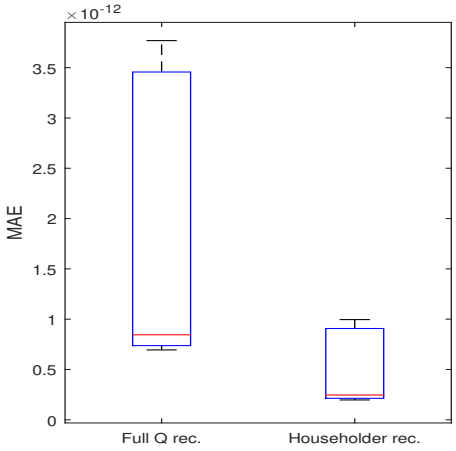


Figura 7.8: MAE results.

Error (MAE), PSNR (Peak Signal-to-Noise Ratio) (Hore y Ziou 2010). We also measure the SSIM (Structural Similarity Index). Since we are reconstructing 128 slices, we show the minimum, maximum and average results of both reconstruction techniques.

In Figures 7.7 and 7.8 we can see the PSNR and MAE results respectively. We don't show the SSIM results, since all the images get a SSIM equal to 1. This means that all the reconstructions have the same structure as the reference image (we are not losing significant internal structures). As we observe in the Figures, the reconstructions obtained through the Householder matrix have better quality, which reflects the better numerical stability of the factorization. It gets around 10 units of PSNR higher and lower error. But both of the methods get really high quality. Notice that we are getting MAEs of order 10^{-12}, which is almost insignificant.

In Figure 7.9 we show the reference CT image of an abdomen and the reconstructed images with both techniques. As we can see, the images are almost identical to the human eye, since we can not discern significant differences or information loss.

(a) Reference **(b)** Full Q rec. **(c)** Householder rec.

Figura 7.9: Abdomen CT Reconstructions.

7.4 Conclusion

In this work, we have conducted a study of the efficiency of applying a direct algebraic technique to the CT image reconstruction problem. We have compared two methods, the QR factorization forming the matrix Q explicitly, or using the matrix Q in the form of Householder reflections. To verify the performance of the methods we have used the SuiteSparseQR library, which includes a parallel implementation of these algorithms. We have used a server to get the experimental time performance, using up to 32 processors.

Regarding the quality obtained by both methods, we have determined that the Householder factorization is more numerically stable, so the images have less error. However, we speak of very small magnitudes that are not perceptible to the human eye.

On the other hand, we have verified that the time used to perform the factorization of the system matrix is much higher if we form the matrix Q in an explicit manner. This can lead to errors as we expose ourselves to system failures, power outages, etc. In addition, we have verified that the reconstructions are faster when we reconstruct several images simultaneously. We get less time per slice in general, and less time using the Householder form in particular. Since in reality we are going to reconstruct more than one slice at a time (usually a CT study has at least 100 slices), we determine that it is more efficient to use Householder reflections for our application. In this way we can reconstruct volumes with high quality in less than 30 minutes.

Finally, we have determined that the efficiency that is achieved using fine-grained parallelism in many-core servers is not good. We observe that we do

not take advantage of all the allocated resources, obtaining very low Speedups. Still, the time used to solve 128 slices with 32 threads is slightly lower than using the 32 threads to solve each slice independently using coarse-grain parallelism.

At this point, it is still necessary to employ HPC computers since this implementation requires a high amount of main memory. That is the reason we are not able to compute the problem for higher image resolutions. As future work, we plan to work with out-of-core techniques that read the data stored in blocks from the hard drive when the particular block is needed for the computation, instead of having it always loaded in main memory. In this way, we could achieve to reconstruct bigger problems in workstations with lower amount of RAM memory and thus lower cost.

Acknowledgements

This research has been supported by "Universitat Politècnica de València", "Generalitat Valenciana" under PROMETEO/2018/035 co-financed by FEDER funds, as well as ACIF/2017/075 predoctoral grant, and the "Spanish Ministry of Economy and Competitiveness" under Grant TIN2015-66972-C5-4-R and TIAMHA co-financed by FEDER funds.

Capítulo 8

Comparativa de los métodos QR y LSQR: evaluación con múltiples imágenes reales

El artículo presentado en este capítulo corresponde a la publicación realizada en la revista *Radiation Physics and Chemistry* en el año 2020, con referencia (Chillarón, Vidal y Verdú 2020a).

En este artículo se realiza un estudio comparativo del método algebraico directo QR utilizando reflexiones de Householder con el método algebraico iterativo LSQR+STF+FISTA empleando los parámetros óptimos extraídos de estudios anteriores. Para ello, se ha utilizado el banco de imágenes *DeepLesion* (Yan y col. 2018), que contiene una colección de imágenes TC reales de distintas zonas del cuerpo. Además, este *dataset* está centrado en la detección de lesiones mediante técnicas de inteligencia artificial, por lo que dichas lesiones están marcadas y etiquetadas. De este modo, se van a emplear las lesiones marcadas como guía para evaluar la calidad que obtienen ambos métodos.

Para ello, se han seleccionado 8 imágenes correspondientes a diferentes tipos de lesiones: de hueso, abdomen, mediastino, hígado, pulmón, riñón, tejido blando y pelvis. El propósito del estudio, además de analizar la diferencia de calidad

general, es determinar si todas las lesiones escogidas son detectables en las reconstrucciones obtenidas por ambos métodos.

Las imágenes empleadas se han reproyectado mediante el método de Joseph, realizando previamente un reescalado puesto que originalmente tienen una resolución de 512×512, la cual no se puede alcanzar todavía por falta de memoria. Por tanto, se va a emplear el mismo caso que en el capítulo anterior, con resolución 256×256 y 90 vistas.

Guiándonos por las métricas de calidad objetivas, el método QR obtiene reconstrucciones de mucha más calidad, situándose su PSNR siempre por encima de 200, mientras que el LSQR obtiene PSNR ligeramente superior a 60 para todos los casos, lo cual supone una gran diferencia en cuanto a ruido en la imagen. Sin embargo, los resultados del SSIM son menos distantes. Mediante el método QR se obtiene un SSIM de 1 en todos los casos, puesto que las soluciones son perfectas. Con LSQR se obtienen valores de SSIM por encima de 0.996 en todos los casos, y en la mitad de ellos por encima de 0.999, siendo así reconstrucciones que conservan casi íntegramente todas las estructuras relevante de las imágenes.

Prueba de ello es que en todos los casos ha sido posible distinguir las lesiones marcadas mediante ambos métodos. Visualmente, las reconstrucciones no muestran grandes diferencias. Sin embargo, en algunos casos, el método LSQR obtiene imágenes que muestran *oversmoothing*, o un efecto de suavizado excesivo, lo que podría resultar en pérdida de información de ciertos tejidos. Además, las imágenes contienen ruido aunque no se apreciable visualmente, pero sí lo es al variar el tamaño de la ventana cuando se visualizan distintas zonas que con las reconstrucciones de la QR por tener variaciones en los valores de unidades Hounsfield.

Como conclusiones de este estudio se puede determinar que el método directo QR puede obtener mejores reconstrucciones cuando la matriz del sistema es de rango completo, con la principal ventaja de poder acotar el tiempo de reconstrucción y la calidad obtenida. Por otra parte, el método LSQR aporta la ventaja de poder disminuir el número de vistas en mayor cantidad, lo que es clave a la hora de reducir la dosis. Sin embargo, la calidad de las reconstrucciones es ligeramente menor para rango completo, y aún menor para rango deficiente. Además, por ser iterativo, no está garantizada la convergencia, ni el número de iteraciones necesarias, con lo cual no es posible determinar el tiempo a priori a no ser que se delimite a un número de iteraciones fijo. Además, como ya se ha estudiado con anterioridad, hay diversos parámetros de los

que depende la calidad y la convergencia, por lo que este método necesita ser supervisado por un usuario que escoja los parámetros si no está automatizado.

Ambos métodos presentan sus ventajas e inconvenientes, pero aportan dos aproximaciones diferentes según el problema a resolver y el objetivo a conseguir: reducción de dosis máxima, o calidad de imagen óptima. En todo caso, tener la posibilidad de elección es algo positivo que extraemos de este estudio para nuestra línea de investigación.

Radiation Physics and Chemistry

Volume 167: 108289, 2020

CT image reconstruction with SuiteSparseQR factorization package

Mónica Chillarón[1], Vicente Vidal[1] and Gumersindo Verdú[2]

1 Departamento de Sistemas Informáticos y Computación (DSIC), Universitat Politècnica de València, Valencia, Spain.
2 Instituto de Seguridad Industrial, Radiofísica y Medioambiental (ISIRYM), Universitat Politècnica de València, Valencia, Spain.

Abstract

SuiteSparseQR is a factorization package for sparse matrices oriented to parallelism in multicore architectures. It employs BLAS and LAPACK as well as Intel's Threading Building Blocks to achieve high performance. Through the SPQR method implemented in this package we can use the QR decomposition to reconstruct CT images efficiently. In this paper, we analyze the behavior of the package applied to the reconstruction of medical CT images, studying the quality of the obtained image. To this purpose, we use the image dataset DeepLesion, which provides various CT studies of different lesions in different organs or tissues. We also compare it to our previous iterative reconstruction method called LSQR. This new method is promising since the computations are simplified if we compare it to the iterative options and the reconstructions are high-quality, as the results show.

Keywords: SuiteSparse, QR, SPQR, LSQR, CT, Reconstruction, Medical Imaging, DeepLesion.

8.1 Introduction

Currently, Computerized Tomography (CT) studies are well established in clinical practice for the diagnosis of multiple diseases as well as for their monitorization. Unlike other medical imaging methods for diagnostics such as Magnetic Resonance Imaging (MRI) or ultrasounds, the high power X-rays generated during CT scanning is a hazard for all patients, especially for vulnerable patients such as children or pregnant women, and also cancer patients who require regular scans to assess the evolution of the disease (Kak y Slaney 2001).

Several studies have shown the generalized increase of cancer incidence all over the world, and also the projection of cancer-provoked deaths by 2020 and 2030 (Rahib y col. 2014). For instance, the risk of having any type of cancer is about a 30% nowadays in the European Union, according to the *The International Agency for Research on Cancer ("IARC")*. The statistics for 2018 are summarized in ("Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries"). Also the projections show an decrease of mortality rates but the incidence is going up in general terms, and at a higher pace for several types such as lung, breast or thyroid cancer.

Since the radiation produced by an X-ray machine can induce cancer by itself, it is necessary to develop new methods that can allow reducing it to the maximum while still having a good image quality that helps physicians make accurate diagnostics.

Since the issue of radiation came across, the reconstruction techniques for CT scans have significantly changed. When this clinical test started, they used methods that were based on the Fourier transform. They are called analytical reconstruction methods and to this day they remain being the most used in clinical practice, being all modifications of the filtered-backprojection (FBP) algorithm (Feldkamp, Davis y Kress 1984; Tang y col. 2006). That is due to their fast reconstructing times, since you can have a complete CT scan in a matter of seconds.

However, they are not the best option regarding radiation, since they need a complete set of projections to obtain a good-quality image. If the number of projections is reduced, the resulting images have artifacts such as rings or beam-hardening that can mean they are invalid for diagnosis.

On the other hand we find algebraic methods, that can be either iterative or direct. Iterative techniques such as ART (Andersen 1989), SART (Andersen y Kak 1984) or LSQR (Flores y col. 2014; Flores, Vidal y Verdú 2015; Parcero y col. 2017) perform an approximation to the solution image and they obtain a good balance between radiation dose and image quality since they allow us reducing the numbers of projections taken when doing a CT scan. Nevertheless, they are really slow methods since they iterate to approximate the result image.

Direct methods are not widely explored for this field due to their high computational cost for the large matrices we are working with. But with hardware evolving fast and becoming more affordable, It is now possible to compute factorizations of extremely large matrices. If we also take advantage of parallel

computing or High-Performance Computing (HPC) it is possible to take the direct algebraic approach to the CT image reconstruction problem.

In this paper we focus on the application of the QR matrix factorization for CT image reconstruction. We perform the QR decomposition in a cluster using the software included in the package SuiteSparseQR (Davis 2011). This package gives us a lot of basic sparse matrices algebraic algorithms and also provides the possibility of using BLAS threads as well as Intel Threading Building Block (TBB) that can parallelize the factorization code and allows us to exploit our hardware resources to the maximum. In works like (Rodríguez-Alvarez y col. 2018) they perform the QR factorization to reconstruct small CT images but as they mention, the method is not parelellized or optimized to reduce the fill-in of the matrix, which the implementation in this packages does. In our previous work (Chillarón y col. 2018), we did a preliminary study in which we applied the multifrontal sparse QR (SPQR) method to reconstruct phantom images, in order to determine its feasibility. We concluded the maximum resolution we could reach with the hardware we have available is 256x256 pixels. Besides, we observed we need a full-rank system matrix to reconstruct the images, which in this case is obtained with 90 projection angles.

In Section 8.2, we will describe the method used to simulate a CT scanner, as well as the way to perform the reconstruction of a CT sinogram using the QR factorization or the LSQR method. We also make a brief description of the images dataset DeepLesion, used to take as reference, and the metrics we use to measure the image quality. In Section 8.3 we will analize the performance in terms of quality of the CT reconstructions using the QR decomposition. Moreover, the resulting images will be compared to the best quality reconstructions obtained through LSQR so we can determine which method is a better alternative and discuss the disadvantages of each of them. To conclude, in Section 8.4 we summarize and discuss the advantages of the studied methods and propose a future line of work.

8.2 Material and Methods

8.2.1 Computerized Tomography

The CT image reconstruction problem can be modelled as described in Equation (8.1), so it can be be solved by algebraic methods. In this linear equations system, A (Equation (8.4)) is the system matrix or weights matrix, which measures the contribution of each ray beam traced (i) to each image pixel (j). We

calculate this matrix using the Joseph method (Joseph 1982). The size of A is $M \times N$, being M the size of the x-ray projections (nº of views*nº of detectors) and N the resolution of the image to be reconstructed (32×32, ..., 512×512 pixels). The projection data is g (Equation (8.2)), which is the X-ray sinogram represented in vector form. Last we have the vector u (Equation (8.3)), or the solution image we get when we solve the system (Equation (8.1)) for u.

$$A * u = g \tag{8.1}$$

$$u = [u_1, u_2, ..., u_N]^T \in \mathbb{R}^N \tag{8.3}$$

$$g = [g_1, g_2, ..., g_M]^T \in \mathbb{R}^M \tag{8.2}$$

$$A = a_{i,j} \in \mathbb{R}^{MxN} \tag{8.4}$$

8.2.2 QR reconstruction method

If the rank of the system matrix A is complete, the problem could be solved directly using its inverse as Equation (8.5). But our matrices are too large to explicitly calculate the inverse. It would imply a high computational cost and the accumulating errors could distort the solution image.

Instead, we use the QR factorization (Golub y Ortega 1993) of A (Equation (8.7)), where the matrix Q is orthogonal and R is upper triangular. Besides, we perform the decomposition version with pivoting, where the permutation matrix P is used to reduce the fill-in of the matrices in the factorization process. With this factorization we simulate the inverse as Equation (8.8) (or pseudoinverse as Equation (8.6) if A is rank-deficient).
In this way we can solve the problem with direct algebraic methods as Equation (8.9) and most of the computations can be made in advance and store for later use to reconstruct the image.

$$u = A^{-1} * g \tag{8.5}$$

$$u = A^{+} * g \tag{8.6}$$

$$A * P = Q * R \tag{8.7}$$

$$A^{-1} = PR^{-1}Q^T \tag{8.8}$$

$$u = P * (R^{-1}(Q^T * g)) \tag{8.9}$$

8.2.3 Q-less reconstruction method

An alternative to the traditional QR factorization is using Householder reflections. This means we don't need to explicitly calculate and store Q. Instead, we can store only the transforms to be applied in Householder form as Equation (8.10), where H_i are the successive reflection matrices. By doing this, we significantly reduce the computational cost and we spare main memory resources, which is vital to this scenario.

$$Q = H_1 H_2 \cdots H_{m-2} H_{m-1} \tag{8.10}$$

8.2.4 Least Squares QR reconstruction method

The problem in Equation (8.1) can also be solved by iterative algebraic methods. The Least Squares QR method, presented in (Paige y Saunders 1982), is a good choice for sparse and possibly rank-deficient systems as is our case. We verified its validity in our previous works (Flores y col. 2014; Flores, Vidal y Verdú 2015; Parcero y col. 2017; Chillarón y col. 2017).

This method is based on the bidiagonalization method by Golub and Kahan (Golub y Kahan 1965). It solves the system by minimizing $\|g - Au\|_2$, generating a sequence of approximations u_k such as the $2-$norm of the residue (being the residue $r_k = g - Au_k$ for iteration k) decreases monotonically. The complete LSQR process is shown in Algorithm 3, where α_{i} and β_1 are chosen to normalize the corresponding vectors. For instance, the operation $\alpha_1 v_1 = A^{\mathrm{T}} x_1$ implies computing $\bar{v_1} = A^{\mathrm{T}} x_1$, $\alpha_1 = \|\bar{v_1}\|$ and $v_1 = (1/\alpha_1)\bar{v_1}$.

In our reconstruction method, we combine this iterative method with a regularization technique called Soft Thresholding Filter (STF)(Yu y Wang 2010; Yu y Zeng 2014) and an acceleration step called Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) (Beck y Teboulle 2009). Both of these techniques help with the convergence rate and they are especially needed for ill-conditioned problems. The combination is as follows:

1. Initialization: $u_0 = 0$
2. Iterate:

- Update the current reconstruction using a fixed number of LSQR iterations.
- Perform the filtering step using STF
- Apply acceleration technique FISTA
- Return to step (2) until the stopping criterion is satisfied or we reach a maximum number of iterations

Algorithm 3 LSQR

INPUT: A, g, u_0, iter
OUTPUT: u

1) Initialize
$\beta_1 x_1 = g$
$\alpha_1 v_1 = A^T x_1$
$w_1 = v_1$
$\overline{\phi}_1 = \beta_1$
$\overline{\rho}_1 = \alpha_1$

for i=1:iter **do**
2) Bidiagonalization
$\beta_{i+1} x_{i+1} = A v_i - \alpha_i x_i$
$\alpha_{i+1} v_{i+1} = A^T x_{i+1} - \beta_{i+1} v_i$
3) Update scalars
$\rho_i = (\overline{\rho}_i^2 + \beta_{i+1}^2)^{1/2}$
$c_i = \overline{\rho}_i / \rho_i$
$s_i = \beta_{i+1} / \rho_i$
$\theta_{i+1} = s_i \alpha_{i+1}$
$\overline{\rho}_{i+1} = -c_i \alpha_{i+1}$
$\phi_i = c_i \overline{\phi}_i$
$\overline{\phi}_{i+1} = s_i \overline{\phi}_i$
4) Update solution
$u_i = u_{i-1} + (\phi_i / \rho_i) w_i$
$w_{i+1} = v_{i+1} - (\theta_{i+1} / \rho_i) w_i$
end for

As stopping criterion, we take the relative residual $\|g - Au\|_2 / \|g\|_2$, set to a minimum tolerance which is usually 1e-06. On each iteration, we take the solution modified by FISTA as the initial solution for the next LSQR iteration.

8.2.5 Image Quality Metrics

To measure the quality of the reconstructed images, we have used the metrics PSNR (Peak Signal-To-Noise Ratio) and SSIM (Structural Similarity Index) (Hore y Ziou 2010). The PSNR metric measures the ratio between the image signal and the noise it contains. To calculate it, another metric is used, the so-called Mean Square Error (MSE), which is calculated according to Equation (8.11), and represents the mean of the squared error between the reference image I_0 and the reconstructed image u. Once the MSE is calculated, it is used to calculate the PSNR according to Equation (8.12), in which MAX represents the maximum value that a pixel can take. The higher the PSNR value we get, the better the reconstruction we have.

With SSIM we can measure the internal structures (shapes) of the images compared with the reference image. Therefore, it does not look at the gray levels of the pixels, but the shapes of the reconstructed image with respect to the reference image, and therefore measures what is perceptible to the human eye. It is applied through windows of fixed size, and the difference between two windows x and y corresponding to the two images to be compared is calculated, using Equation (8.13). In this equation, μ_x and μ_y denote the average value of the window x and y, σ_x^2 and σ_y^2 the variance, σ_{xy} the covariance between the windows, and c_1 and c_2 are two stabilizing variables dependent on the dynamic range of the image.

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} ||I_0(i,j) - u(i,j)||^2 \tag{8.11}$$

$$PSNR = 10 * log_{10} \frac{MAX_{I_0}^2}{MSE} \tag{8.12}$$

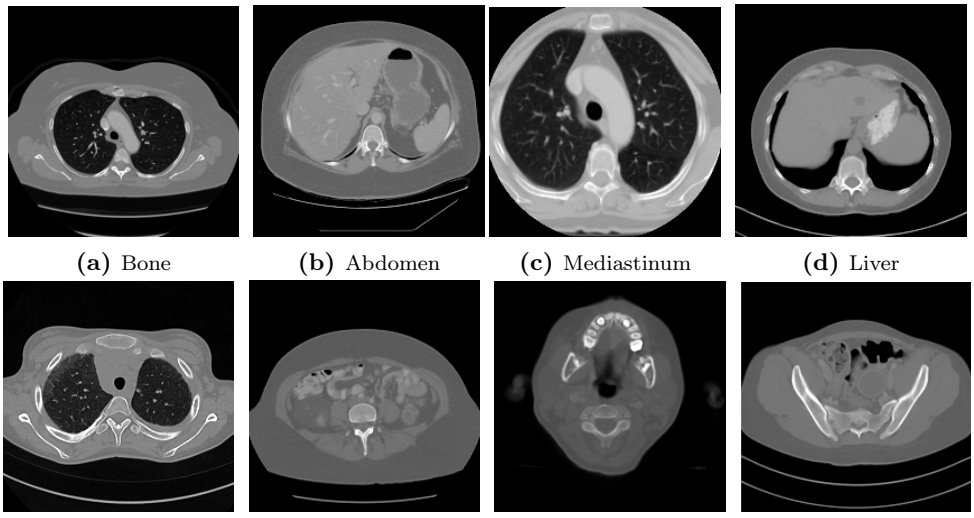
$$SSIM = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{x,y} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{8.13}$$

8.2.6 CT Images Dataset

To test the validity of our reconstructions, we used images pertaining to DeepLesion (Yan y col. 2018), a dataset with 32,735 lesions in 32,120 CT slices from 10,594 studies of 4,427 unique patients. In this images collection we can find CT studies for different types of lesions, with tags that divide them into eight classes: Bone, Abdomen (that are not Liver or Kidney), Mediastinum, Liver, Lung, Kidney, Soft tissue and Pelvis.

We choose one image for each lesion category, shown in Figure 8.1. The images were chosen randomly, one for each type of lesion, and selected from a list of the key slices from each study provided by the authors. Since these images are already reconstructions, it is implied the scanner software used its own algorithms to try to remove Poisson noise, as well as the artifacts due to detectors noise or beam hardening. But the images we use as reference can still have some noise since they are real reconstructions.

This dataset gives us the required variety of images to test our reconstructions, both with direct and iterative methods. For this purpose, we project the selected images with Joseph (Joseph 1982) method as we described in Section 8.2.1. Although the original resolution of the images is 512x512 pixels, we resize them to 256x256 pixels since it's the highest resolution we can compute at the moment. The sinograms were simulated for a scanner with 1025 detectors and using 90 views along the 360 degrees of rotation. Then, we measure the quality of the results with the metrics PSNR and SSIM (Hore y Ziou 2010) and we also try to locate the lesion shown in the reference projected image.



(a) Bone **(b)** Abdomen **(c)** Mediastinum **(d)** Liver

(e) Lung **(f)** Kidney **(g)** Soft Tissue **(h)** Pelvis

Figura 8.1: Selected images classified by type of lesion.

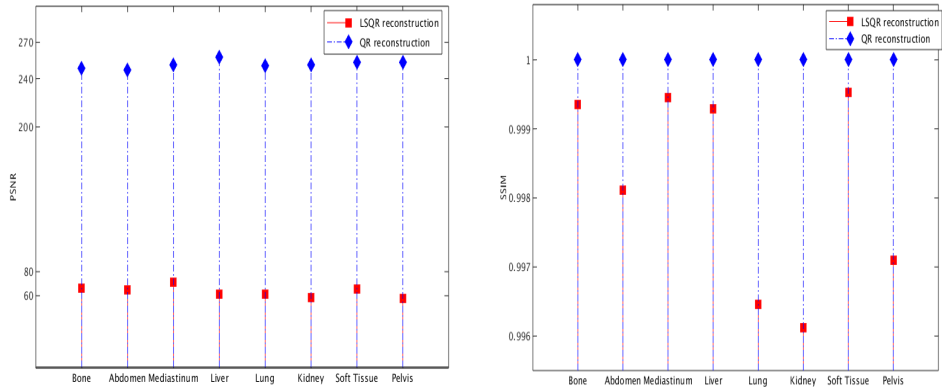
8.3 Results and Discussion

We have used the eight images of selected lesions to reconstruct with each method and compare the results obtained. All the computations have been made in an HPC cluster with four Intel Xeon E5-4620 8c/16T processors (8 cores/processor, 32 cores/node) and 256GB DDR3 RAM memory (ratio 8GB/-core), using up to 32 threads.

With SPQR, the reconstruction is trivial since it consists of computing Equation (8.9), using the QR factorization previously performed and stored. With LSQR, the process is more complex, since as we analyzed in our previous work (Chillarón y col. 2017), in our method (which combines LSQR, the STF filter, and FISTA acceleration) several parameters need to be adjusted in order to get the better result possible. Therefore, the reconstruction has to be supervised since there is no guarantee of convergence.

In Figure 8.2 we show the PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index) of the two techniques for each lesion image. To measure this, we selected the image region that has the body part, eliminating the outer region with only air.

As we can see in Figure 8.2a, the reconstructions obtained by LSQR seem to be of much lower quality than those obtained by QR reconstruction, having a



(a) PSNR results **(b)** SSIM results

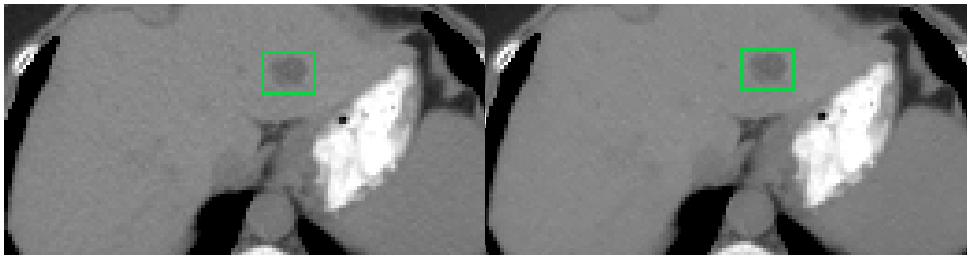
Figura 8.2: Quality of the reconstructions.

difference of around 200 in the PSNR value. The reconstructions with the QR can be considered noiseless with these results, compared to the reference image. The mean squared error (MSE) is of the order of 1e-24 in this case, while for the reconstructions with LSQR it is of the order of 1e-05. Therefore, reconstructions with the iterative method have more noise, while the ones obtained with the QR are practically identical to the reference image. These results are logical since the iterative method is stopped at a desired tolerance, while the reconstruction with the QR is direct and the result is not approximated but calculated.

Regarding the SSIM results, shown in Figure 8.2b, the difference is not as big. In every case, the QR reconstructions SSIM value is 1, which is the maximum possible. Nevertheless, for the LSQR the minimum SSIM is 0.996, which can be considered a high-quality result.

For some lesions (Bone, Mediastinum and Soft Tissue) the SSIM value is above 0.999. This means even though the reconstructions could be noisy, the internal structures of the CT images are well preserved generally speaking. But we could still be losing relevant information with LSQR, since SSIM is not always 1, which doesn't happen with the direct method.

To visually check if we lose information, we show the reconstructed images in Figure 8.4 to Figure 8.11, each one with the lesion's bounding box marked with a green rectangle in the reference image, as well as in the reconstructions. We show the appropriate Hounsfield window to correctly visualize each one of the lesions, as instructed in the dataset information.



(a) QR rec. **(b)** LSQR rec.

Figura 8.3: Liver Lesion (zoomed-in).

In every case, we have been able to locate the lesion in both of the reconstructed images. In addition, we can see the LSQR results are better than expected since the numerical error is not visually discernible when showing the relevant window. So even if the quality metrics indicate a significant difference, we don't perceive it as such. We can notice a slight blurriness in every LSQR reconstruction, but it does not alter the image much.

As an example of this, we provide a zoomed-in image of the reconstructions corresponding to the Liver lesion in Figure 8.3. As we can see, the area of the liver is smoother on the LSQR reconstruction, which differs from the reference image (Figure 8.7). In addition, the edges are slightly more defined on the QR reconstruction, as can be observed in the lesion bounding box. But the difference is still small, as it is for the other lesion reconstructions.

In some particular cases, as can be the image corresponding to the Mediastinum lesion (Figure 8.6), we observe that the QR reconstruction is identical to the reference, but not the LSQR. When windowing the image to the desired Hounsfield Window, we lose some information. That does nor mean the information is not there, just that by windowing we can not see it because the Hounsfield value is not exactly the same in the LSQR reconstruction.

(a) Reference **(b)** QR rec. **(c)** LSQR rec.

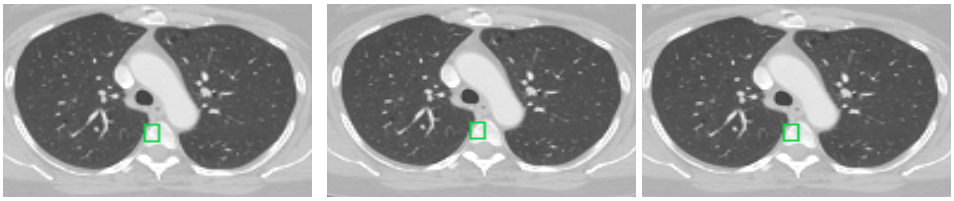


Figura 8.4: Bone Lesion. HU Window=[-1500,500].

Figura 8.5: Abdomen Lesion. HU Window=[-175, 275].

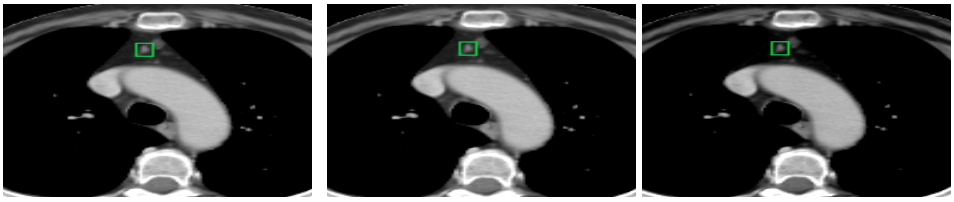


Figura 8.6: Mediastinum Lesion. HU Window=[-175, 275].

Figura 8.7: Liver Lesion. HU Window=[-175, 275].

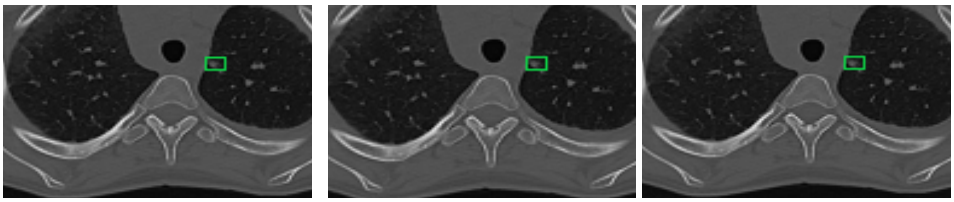


Figura 8.8: Lung Lesion. HU Window=[-1500, 500].

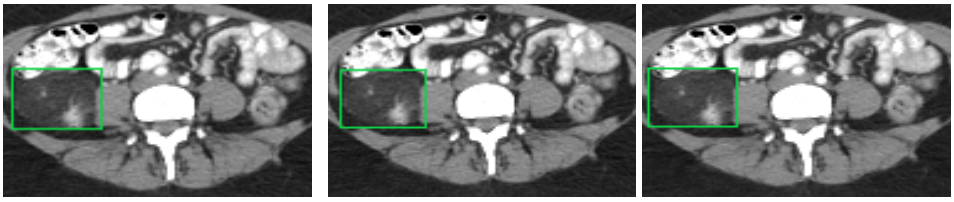


Figura 8.9: Kidney Lesion. HU Window=[-160, 240].

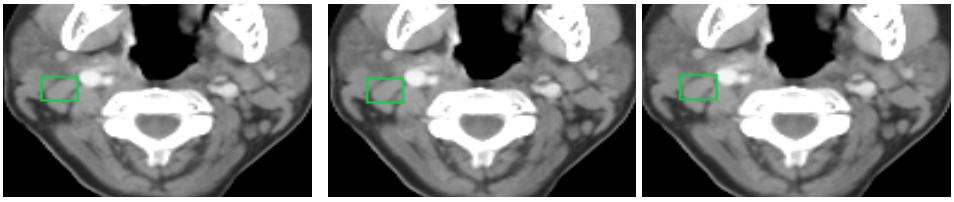


Figura 8.10: Soft Tissue Lesion. HU Window=[-160, 240].

Figura 8.11: Pelvis Lesion. HU Window=[-175, 275].

8.4 Conclusion

In the present work, we have determined the validity of a direct algebraic method for the reconstruction of CT images. To do this, we have used various images belonging to the DeepLesion dataset, representing different parts of the body with different types of lesions.

In our previous work (Chillarón y col. 2018), we determined that the case with the highest resolution that we can solve by means of the QR factorization is for images of 256x256 pixels. This is because we need the system matrix A that models the CT scanner to have full rank to get valid results. Therefore, we have to take at least 90 projections. If we went up to 512x512 pixels resolution, we would have to use 260 views, and the A matrix would be of size 266500x262144 in our simulations. This case is not computable with the algorithms provided by SuiteSparse and the hardware that we currently have. However, it is possible to solve it by means of LSQR, since we don't factorize the matrix which is always sparse, just operate with it performing matrix-vector products. Therefore it needs less computational resources. In addition, for LSQR we do not need the matrix to be full rank. We can obtain good reconstructions with fewer projections, as shown in our previous works (Chillarón y col. 2017; Parcero y col. 2017). Therefore, although using QR we can reduce the number of projections with respect to analytical methods, we do it to a lesser extent than with LSQR.

On the other hand, the reconstruction is much simpler with the direct method, since we can factorize the matrix once before the reconstruction and have it always stored. When reconstructing, we solve Equation (8.9), which is quite fast. In this case, we solved a full volume with 128 slices in less than 30 minutes, depending on the number of cores used, using the Householder form to store Q. If we used the explicit Q, the time required to reconstruct the same volume is doubled. To reduce this time and use less resources, we are working on a new QR factorization algorithm that uses out-of-core techniques to read the data stored in blocks only when it is needed, so we can reconstruct higher resolutions. In addition, since it is a direct operation we can delimit the time it will take depending on the hardware used.

This is not possible with LSQR, since it is an iterative method. We never know how many iterations it will need to converge or if it will converge. We could set it to a number of iterations to delimit the time, but in this way, we can not guarantee a minimum tolerance. In addition, the selection of parameters that influence reconstruction should be supervised. Besides, we currently don't have a tested implementation to solve several right-hand-side vectors with LSQR,

so we need to solve each slice independently, unlike with the QR. All in all, we're talking about several hours to get a full volume reconstruction. However, we are working on implementing a Block LSQR using parallel libraries to reconstruct several right-hand-sides, and our preliminary tests show a similar performance than the QR reconstructions using the Householder form when the parameters are well selected.

Regarding the image quality, we have been able to observe that the reconstructions that are obtained applying the direct method based on the QR decomposition are perfect, practically speaking. It can be said that there is no error with respect to the reference images. Nevertheless, it must be noted that in our simulations we are not considering or removing the possible noise generated in the acquisition process of a real CT scanner. In addition, we have compared them to the reconstructions obtained by our previous method based on LSQR. These reconstructions show worse quality from the numerical point of view, which we have verified analyzing the results with both PSNR and SSIM metrics. However, when viewing the images we see that even LSQR obtains reliable results since we are only able to appreciate a slight blurring in some areas.

For all the above we determined that both methods are valid to reconstruct CT images with an algebraic approach. However, the accuracy of the QR method as well as the simplicity of the process is preferable from the point of view of image quality. Nevertheless, this method requires greater computational resources, as well as a greater number of views than the LSQR, with which we can solve larger problems.

Acknowledgements

This research has been supported by "Universitat Politècnica de València", "Generalitat Valenciana" under PROMETEO/2018/035 co-financed by FEDER funds, as well as ACIF/2017/075 predoctoral grant, and the "Spanish Ministry of Economy and Competitiveness" under Grant TIN2015-66972-C5-4-R and TIAMHA co-financed by FEDER funds.

Capítulo 9

Implementación del método QR con técnicas Out-Of-Core

El artículo presentado en este capítulo ha sido publicado en la revista *Computer Methods and Programs in Biomedicine*, con referencia (Chillarón y col. 2020). En él, se emplea una implementación del método QR mediante técnicas Out-Of-Core a bloques para resolver el problema de reconstrucción de TC.

Tal y como se ha demostrado en trabajos como (Joffrain, Quintana-Ortí y Geijn 2005; Gunter, Reiley y Geijn 2001; Marqués y col. 2011), el hecho de emplear técnicas Out-Of-Core para resolver problemas de álgebra lineal es altamente beneficioso. Además de poder afrontar problemas que no serían asumibles en ninguna máquina por falta de memoria RAM, permite disminuir el coste del hardware necesario para resolver problemas de grandes dimensiones. Esto se debe a que el rendimiento de los programas va a estar dominado por el tiempo de entrada/salida, al haber muchas lecturas y escrituras de bloques según se vayan necesitando para los cálculos, o sean modificados. Es por ello que no va a ser necesario invertir en memoria RAM, cuyo coste es muy elevado, sino que el coste temporal se podrá ver reducido empleando discos de estado sólido (SSD) con altas velocidades de lectura y escritura. De este modo, problemas de grandes dimensiones podrán ser resueltos en máquinas de coste moderado y por tanto accesibles al público general, como se demostrará en este capítulo.

La técnica utilizada en este artículo para realizar la factorización QR había sido presentada previamente en (Marqués y col. 2009), pero únicamente testeada para matrices de pequeñas dimensiones. En este trabajo, se analiza el uso en problemas de grandes dimensiones, así como el efecto de emplear discos de estado sólido en lugar de los discos tradicionales. Además, se implementa la resolución del sistema de ecuaciones empleando la factorización obtenida, paso no realizado previamente, y se analiza su comportamiento.

Tal como definen Marqués y col., el método de la factorización QR mediante un algoritmo a bloques está implementado empleando libflame, y su definición se puede encontrar en el artículo original. Dicha factorización emplea las reflexiones de Householder para no formar la Q de forma explícita. Además de la propia factorización, los autores desarrollan un *runtime* que es capaz de realizar una factorización simbólica para identificar a priori la lista de tareas a realizar. De este modo, se identifican los datos que van a ser necesarios en cada paso de la factorización, pudiendo emplear esta información para cargar en memoria los bloques que vayan a ser necesarios para una tarea antes de que se vaya a ejecutar. Por tanto, si el trabajo de entrada/salida y el trabajo de cálculo se realiza por distintos hilos, se pueden llegar a solapar, disminuyendo el tiempo necesario para realizar la factorización. La misma filosofía se ha utilizado para realizar la resolución del sistema asociado a la reconstrucción de la imagen TC.

En el artículo que se presenta a continuación se definen los tipos de tareas necesarias para factorizar la matriz utilizando la técnica de algoritmo a bloques, así como para resolver el sistema, definiendo los datos (bloques) de entrada y de salida de cada tipo de tarea.

En cuanto a los resultados de la factorización, no se ha estudiado la eficiencia temporal por no ser crítica ya que puede realizarse una sola vez y almacenarse para posteriores usos. Sin embargo, sí se ha analizado el residuo relativo obtenido tras las reconstrucciones, para evaluar la estabilidad numérica del método al aumentar el tamaño del problema. Los residuos relativos son del orden de 1e-12, y aunque aumentan ligeramente al aumentar la resolución, no lo hacen de manera que pueda suponer un problema si se quiere incrementar todavía más el tamaño del problema a resolver.

Por otra parte, la calidad de las imágenes se ha medido utilizando las métricas PSNR y SSIM. Los resultados de PSNR para el fantoma Forbild se encuentran siempre por encima de 200 independientemente de la resolución, y el SSIM siempre es igual a 1. Además, se han reconstruido múltiples imágenes de TC reales seleccionadas de forma aleatoria del *dataset* DeepLesion (Yan y col.

2018), obteniendo un PSNR medio de 220 para 2048 cortes diferentes con resolución 512×512, y SSIM igual a 1.

Con el fin de comparar las imágenes obtenidas con otros métodos, se han analizado las reconstrucciones para el caso de resolución 512×512 y 260 vistas mediante LSQR, y también mediante FBP variando el número de vistas. De los resultados se puede determinar que pese a que LSQR obtiene buena calidad, las imágenes contienen más ruido (PSNR por debajo de 60), además de presentar *oversmoothing* como ocurría en el capítulo anterior. Por otra parte, las reconstrucciones mediante FBP con 260 vistas presentan muchos artefactos y ruido, que se va eliminando de manera progresiva con el aumento de vistas a 360, 720, y finalmente 1610, cuando ya se obtienen imágenes de calidad similar al LSQR, aunque ligeramente peores. Este número de vistas mínimo que necesitan los métodos analíticos viene determinado por el teorema de Nyquist-Shannon, explicado con más detalles en el artículo que se presenta a continuación.

Por último, se ha analizado la eficiencia temporal de la reconstrucción empleando cuatro variantes: algoritmo básico utilizando disco HDD, algoritmo con solapamiento de entrada/salida y disco HDD, algoritmo básico empleando disco SSD y algoritmo con solapamiento en disco SSD. El primero se tomará como referencia para calcular el *SpeedUp* que obtienen el resto de métodos.

Para la configuración básica y disco HDD se ha observado que el tiempo total está dominado por las operaciones de entrada/salida, que son más costosas que el tiempo de cómputo independientemente de cuántos lados derechos (cortes) se reconstruyan al mismo tiempo. Por tanto, la entrada/salida es el cuello de botella de este método, y utilizar discos más rápidos así como solapar la cálculos con las lecturas y escrituras deberían mejorar las prestaciones.

Los resultados muestran que el solapamiento con disco HDD no aporta demasiada mejora, logrando un *SpeedUp* ligeramente mayor a 1. Sin embargo, cuando se emplea un SSD, las prestaciones mejoran notablemente. Para la versión básica con 256 cortes se llega a obtener un *SpeedUp* de 6, que aumenta a 9 empleando solapamiento. Al aumentar el número de cortes, el *SpeedUp* va disminuyendo con respecto a la versión básica con HDD, pero se mantiene por encima de 2.5, lo cual no es despreciable. Con los tiempos obtenidos se observa que en la versión con solapamiento y SSD se ha conseguido eliminar casi en su completitud el sobrecoste de las operaciones de entrada y salida, siendo el tiempo total muy similar al tiempo que requieren los cálculos en todos los casos independientemente del número de cortes.

El tiempo por corte también ha sido analizado, determinando que es más eficiente resolver un número alto de lados derechos en la misma ejecución, puesto que de este modo se compensa mejor el tiempo de las operaciones de lectura y escritura con el tiempo total de cálculo. En este estudio se ha logrado reconstruir 256 cortes con un tiempo medio por corte de 1.65 segundos con la versión que emplea solapamiento y el disco SSD, siendo 14.54 segundos el tiempo medio necesario con la versión básica. Al aumentar el número de cortes a 2048, la mejor versión obtiene un tiempo por corte de 0.72 segundos, mientras que la básica obtiene 2.5 segundos.

En conclusión, el método de reconstrucción QR Out-Of-Core implementado mediante algoritmo a bloques abre la posibilidad de resolver sistemas de grandes dimensiones en equipamiento de coste reducido, sin comprometer las prestaciones en comparación con métodos in-core. En este estudio se ha demostrado que se puede reconstruir un volumen de 256 cortes en 7 minutos con calidad óptima, o un estudio de una zona amplia del cuerpo con 2048 cortes en 25 minutos. Pese a que en tiempo todavía no puede compararse con los métodos analíticos, se ha demostrado que la calidad es muy superior logrando una disminución del número de proyecciones de un 84%, que si se traslada a dosis absorbida por el paciente es altamente relevante.

Computer Methods and Programs in Biomedicine

Volume 193: 105488, 2020

Computed tomography medical image reconstruction on affordable equipment by using Out-Of-Core techniques

Mónica Chillarón[1], Gregorio Quintana-Ortí[2] Vicente Vidal[1] and Gumersindo Verdú[3]

1Depto. de Sistemas Informáticos y Computación, Universitat Politècnica de València, Valencia, 46022 Spain.
2 Depto. de Ingeniería y Ciencia de Computadores, Universitat Jaume I, Castellón, 12071 Spain.
3 Depto. de Ingeniería Química y Nuclear, Universitat Politècnica de València, Valencia, 46022 Spain.

Abstract

Background and objective: As Computed Tomography scans are an essential medical test, many techniques have been proposed to reconstruct high-quality images using a smaller amount of radiation. One approach is to employ algebraic factorization methods to reconstruct the images, using fewer views than the traditional analytical methods. However, their main drawback is the high computational cost and hence the time needed to obtain the images, which is critical in the daily clinical practice. For this reason, faster methods for solving this problem are required.
Methods: In this paper, we propose a new reconstruction method based on the QR factorization that is very efficient on affordable equipment (standard multicore processors and standard Solid-State Drives) by using Out-Of-Core techniques.
Results: Combining both affordable hardware and the new software proposed in our work, the images can be reconstructed very quickly and with high quality. We analyze the reconstructions using real Computed Tomography images selected from a dataset, comparing the QR method to the LSQR and FBP. We measure the quality of the images using the metrics Peak Signal-To-Noise Ratio and Structural Similarity Index, obtaining very high values. We also compare the efficiency of using spinning disks versus Solid-State Drives, showing how the latter performs the Input/Output operations in a significantly lower amount of time.
Conclusions: The results indicate that our proposed method and software are valid to efficiently solve large-scale systems and can be applied to the Computed Tomography reconstruction problem to obtain high-quality images.

Keywords: CT, QR factorization, Medical Image, Reconstruction, Out-Of-Core, Affordable Equipment.

9.1 Introduction

Nowadays, Computed tomography (CT) (Kak y Slaney 2001) is an essential diagnostic medical imaging test in clinical practice. Although it involves the use of X-rays and hence gives ionizing radiation in patients, the information provided is critical in many cases. Therefore, it is extremely important to reduce the radiation dose as much as possible, and thus prevent patients from absorbing a higher dose than the recommended one. Otherwise, CTs could be a hazard to them, since it has been proven the X-rays can be harmful, especially to the most vulnerable patients (De González y col. 2009; Hall y Brenner 2008).

The traditional CT reconstruction employs analytical methods, which are based on the Filtered Back-Projection (FBP) (Tang y col. 2006; Zhuang y col. 2004; Mori y col. 2006). They require a complete set of projections of an object, over 360 degrees of rotation and a number of projections higher than the number of detectors. They are still the most common methods because of their low computational cost and therefore fast reconstruction. However, reducing the X-ray dose is difficult when a high-quality image must be obtained. Several methods (Willemink y col. 2013a; Willemink y col. 2013b) have been developed that reduce the radiation dose by minimizing the tube's current or voltage, and then reconstruct the sinograms with statistical methods that improve the image quality compared to traditional FBP-based methods. There are similar low-dose methods that work with dual-energy spectral CT scanners such as (Wu y col. 2018b; Wu y col. 2018a).

Another common approach to reduce the radiation dose is the use of algebraic iterative methods, which do not require a complete set of projections, nor are they restricted in terms of projection angles (Andersen 1989; Andersen y Kak 1984; Yu y Zeng 2014; Flores, Vidal y Verdú 2015; Flores y col. 2014; Chillarón y col. 2017). These types of methods require fewer projections to reconstruct an image. Some works such as (Kopp y col. 2018; Sollmann y col. 2018) show how the use of sparse-sampling CT scanners in the future and performing the reconstruction of the images with few-views methods could potentially reduce the radiation dose induced to the patients in a significant amount. Nevertheless, they involve a high computational cost, which implies that the reconstructions are much slower than with previous methods. Moreover, since these methods are iterative, convergence is not guaranteed, nor the number of iterations in case of convergence. Several works (Yan Liu y col. 2014; Tang, Nett y Chen 2009; Vandeghinste y col. 2013; Zhu y col. 2013) showed the problems of working with few-view limited-angle CT. The use of few views generates streak artifacts that can mask or conceal important parts

of the image to be reconstructed, which can produce information loss. This is potentially harmful since it can lead to wrong diagnosis. It also poses a problem for secondary applications of the CT images, as shown in (Vandeghinste y col. 2013), where the reduction of the number of views to a minimum number implied an inaccurate segmentation of the blood vessels. Sechopoulos (Sechopoulos 2013) showed that few views led to false positives in computer-aided detection for breast mass detection. Unlike direct methods, iterative methods often generate patchy or blocky artifacts in the reconstructed images due to overregularization (Tang, Nett y Chen 2009; Qi, Chen y Zhou 2015; Wu y col. 2019).

Therefore, direct algebraic methods such as the QR factorization (Rodríguez-Alvarez y col. 2018; Chillarón, Vidal y Verdú 2020a) have been explored recently. Although they usually require a greater number of views than the iterative ones (as was shown in a previous work (Chillarón y col. 2018)), they are much more accurate when the rank of the weights matrix is complete. The main drawback of the direct algebraic methods is that the sparsity of the weights matrix cannot be taken advantage of, since the matrix fills in and becomes dense as the factorization process advances. Moreover, space problems because of an insufficient main memory (RAM) can arise. In this case, it is important to find an efficient approach to tackle large problems without having to acquire expensive and specialized dedicated equipment, which would require a large monetary cost.

In our paper, we present a solution to the CT image reconstruction problem by using the direct solution of linear systems based on the QR factorization. By employing special high-performance software techniques, high-quality images are obtained on affordable computers. Without these techniques, the computer required would be very expensive (tens of thousands of dollars), mainly due to the price of the large main memory required to store the data. With these techniques, computers with a price about one order of magnitude smaller can be employed. A careful application of Out-Of-Core (OOC) techniques allows to read and write blocks of data from/to the hard drive just when they are needed for the calculations, instead of loading the whole matrices into main memory. By applying this method, as well as some other techniques, we can solve large-scale problems, and therefore a fast reconstruction of CT images with high resolutions can be achieved. Our new implementation is time-efficient and also scalable, as can be seen in the results. In addition, both very high quality and a reduction in the number of views (and therefore the absorbed radiation dose) are achieved, compared to analytical methods. In our work, we have checked that the OOC approach is still valid on much larger matrices than previous

works. Moreover, we have assessed the performances on both traditional hard drives (HDD) and modern Solid-State Drives (SSD).

The document is organized as follows: Section 9.2 describes the simulation of our projection data, as well as the simulated scanner parameters. It is also explained how to perform a CT reconstruction using the QR factorization of the weights matrix. Besides, the metrics employed to measure the image quality are introduced, and the QR factorization and the reconstruction algorithm are described in detail. Section 9.3 assesses our new method in terms of numerical stability and image quality. A detailed performance study comparing the different configurations using two types of hard drives is also included. Section 9.4 summarizes and discusses the advantages of the studied method, and we conclude with Section 9.5.

9.2 Methods

9.2.1 CT image reconstruction

To reconstruct CT images with an algebraic approach, we model the problem as:

$$AX = B + W \tag{9.1}$$

where $A = (a_{i,j}) \in \mathbb{R}^{M \times N}$ denotes the so-called system matrix, with dimensions $M \times N$. A is the weights matrix that models the physical scanner, being $a_{i,j}$ the contribution of the i-th ray on the j-th pixel. The dimension M is the product of the number of detectors of the CT scanner multiplied by the number of projections or views taken. N denotes the resolution of the image (256×256 pixels, 512×512 pixels, etc.). $B = (B^j)$ is a matrix of $M \times S$ elements, where S is the number of slices to be reconstructed, and B^j denotes the column j that will correspond to the j-th sinogram. $X = (X^j)$ is a matrix of dimensions $N \times S$, where X^j is the column where the reconstructed image corresponding with the j-th sinogram will be stored. W is the noise contained in the sinograms, which will not be considered in this paper.

The sinograms have been simulated using Joseph method (Joseph 1982). We modeled a fan-beam scanner, using the parameters shown in Table 9.1. As was mentioned before, the number of projections taken depends on the desired reconstruction resolution, and it needs to be adjusted so that matrix A has full rank. The projections are selected according to (9.2), where the symmetry of

the projection data is broken by making an angle shift for every quarter of the circumference to improve the rank.

$$\Theta_i = \begin{cases} (360/v)*(i\text{-}1) & \textbf{if} \quad 1 \leq i \leq (v/4) \\ \Theta_{v/4}\text{+}0.5\text{+}(360/v)*(i\text{-}1) & \textbf{if} \quad (v/4) < i \leq (v/2) \\ \Theta_{v/2}\text{-}0.75\text{+}(360/v)*(i\text{-}1) & \textbf{if} \quad (v/2) < i \leq (3v/4) \\ \Theta_{3v/4}\text{-}0.25\text{+}(360/v)*(i\text{-}1) & \textbf{if} \quad (3v/4) < i \leq v \end{cases} \tag{9.2}$$

To solve the problem in (9.1), first the QR factorization of A is computed (9.3), where Q is orthonormal and R is upper triangular. Then, to reconstruct the images, (9.4) is employed.

$$A = QR \tag{9.3}$$

$$X = R^{-1}(Q^T B) \tag{9.4}$$

It is important to note that the QR factorization does not need to be computed for every image being generated, since it does not depend on B. It can be computed just once and, by storing the results, a lot of computational work can be saved.

9.2.2 Image Quality Metrics

To measure the quality of the reconstructed images, we use two well-established metrics for images: PSNR (Peak Signal-To-Noise Ratio) and SSIM (Structural Similarity Index) (Hore y Ziou 2010). The PSNR metric measures the ratio of the image signal to the noise it contains. To calculate it, another metric is used, the so-called Mean Square Error (MSE), which is calculated according to (9.5), and represents the mean of the squared error between the reference

Source trajectory	360^{o} circular scan
Scan radius	75 cm
Source-to-detector distance	150 cm
X-ray source fan angle	30^{o}
Number of detectors	1025
Pixels of the reconstructed image	512^2
Number of projections	260

Tabla 9.1: Simulated fan-beam scanner parameters.

image I_0 and the reconstructed image I (X in our equations). Once the MSE is calculated, it is used to calculate the PSNR according to (9.6), in which MAX represents the maximum value that a pixel can take. The higher the PSNR value we get, the better the reconstruction obtained.

SSIM measures the internal structures (shapes) of the images compared with the reference image. Therefore, it does not focus on the gray levels of the pixels, but on the shapes of the reconstructed image with respect to the reference image. Therefore, it measures what is perceptible to the human eye. It is applied through windows of fixed size, and the difference between two windows x and y corresponding to the two images to be compared is calculated using (9.7). In this equation, μ_x and μ_y denote the average value of the window x and y, σ_x^2 and σ_y^2 the variance, σ_{xy} the co-variance between the windows, and c_1 and c_2 are two stabilizing variables dependent on the dynamic range of the image.

$$\text{MSE} = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (I_0(i,j) - I(i,j))^2 \tag{9.5}$$

$$\text{PSNR} = 10 \log_{10} \frac{\text{MAX}(I_0)^2}{\text{MSE}} \tag{9.6}$$

$$\text{SSIM} = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{x,y} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{9.7}$$

9.2.3 Out-Of-Core computations

Some problems require the storage of data so large that there are no computers with such a main memory or, in case they exist, their prices are very high. Most operating systems provide a virtual memory system to store data (and programs) that do not fit into the computer's main memory at one time. However, its performances are not very high when employed on structured scientific problems. Hence, in high-performance scientific computing, special techniques, called Out-Of-Core (OOC) or Out-Of-Memory (OOM), are required to efficiently process data stored in the hard drive. These techniques keep the data stored in the hard drive, read them into memory, and write them into disk whenever is needed. The aim of these techniques is to minimize the effect of the slow speed of the read and write operations from/to disks in order to render performances as high as possible.

Traditional approach

In modern computer architectures floating-point operations are much faster than memory accesses. Therefore, the ratio of flops to memory accesses in computations is very important. An increased ratio provides much higher performances since it allows to compute several or even many flops per each memory access, and hence cache memories and other modern features can be fully exploited. For instance, matrix-matrix operations obtain significantly higher performances than matrix-vector operations.

In linear algebra, unblocked algorithms perform one stage at a time (e.g. one column in column-oriented algorithms). In contrast, a blocked algorithm performs several stages (e. g. several columns in column-oriented algorithms) of the traditional (unblocked) algorithm at the same time because this aggregation can take advantage of the more efficient matrix-matrix operations. This number of stages (e. g. columns) that are processed at the same time is usually called the block size.

However, since most usual algorithms in linear algebra proceed on triangular matrices, processing a fixed number of columns (or rows) at the same time can make the data to be processed very large at the beginning, and very small at the end, or vice versa. This can make performances not to be optimal because main memory could be underused in some stages and because of the large variation in the data being transferred. This variation of the transfer size can be a problem when the data are stored in disk since this kind of devices are more sensitive to transfer sizes.

There are usually two common types of algorithms: right-looking algorithms update the rest of the matrix (right part) after the processing of the current (block) column or row, thus requiring $\mathcal{O}(n^3)$ writes. In contrast, left-looking algorithms update the current (block) column or row, with the data previously processed (left part), thus requiring $\mathcal{O}(n^2)$ writes. Since the cost of a write operation in hard drives is usually higher than the cost of a read operation, left-looking algorithms are usually preferred when working on data stored in disk. Great efforts have been made to efficiently solve problems from linear algebra whose data do not fit in RAM and must be stored in disk (Toledo y Gustavson 1998; D'Azevedo y Dongarra 2000; Reiley y Geijn 1999; Gunter y Geijn 2005; Joffrain, Quintana-Ortí y Geijn 2005; Gunter, Reiley y Geijn 2001).

Algorithms-By-Blocks

Like blocked algorithms, Algorithms-By-Blocks also perform several stages of the traditional (unblocked) algorithm at the same time in order to take advantage of the higher speeds of matrix-matrix operations. Unlike blocked algorithms, Algorithms-By-Blocks achieve matrix-matrix operations by raising the granularity of the data. First, the traditional (unblocked) algorithm must be reformulated to perform operations that process only scalar elements. Then, the scalar elements are raised to being square blocks of dimension $b \times b$, and the operations processing them are accordingly raised too so that they correctly process these square blocks. Therefore, in the end the whole computation to be performed is divided into many tasks, each one processing a few square blocks (between one and four, but more usually two or three).

One of the main benefits of this approach is that all blocks are always of the same size (except maybe for the final right and bottom blocks). This brings in the benefit of making the majority of the transfers of the same size. Thus, by tuning the block size for a given machine, all the data transfers will be very efficient, regardless of the stage of the algorithm (first stages or last stages).

Quintana-Ortí *et al.* (Quintana-Ortí y col. 2012; Marqués y col. 2011) developed a runtime that can process Algorithm-By-Blocks very efficiently by applying two techniques: The use of a cache of blocks stored in memory to reuse information, and the overlapping of computation and communications to reduce the cost of the latter.

QR factorization

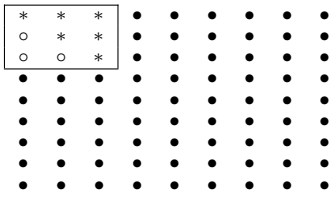
The Algorithm-By-Blocks for efficiently computing the QR factorization was described in 2009 (Marqués y col. 2009). This approach employed the methods and runtime described by Quintana-Ortí *et al.* (Quintana-Ortí y col. 2012; Marqués y col. 2011). However, these works assessed smaller matrices, they did not test modern fast Solid-State Drives (SSD), and they only assessed the QR factorization. In our current work we have checked that this approach is still valid on much larger matrices, we have compared the performances of this approach on both traditional hard drives and modern SSDs, and we have implemented and assessed the application of orthogonal transformations previously computed and the resolution of triangular linear systems (problems not included in these previous works).

Figure 9.1 illustrates the process performed by a left-looking Algorithm-By-Blocks for computing the QR factorization of a 9×9 matrix with block size 3. The '$\bullet$' symbol represents a non-modified element by the current task, the '$*$' symbol represents a modified element by the current task, and the '$\circ$' symbol represents a nullified element (either by the current task or by a previous task). The nullified elements are shown because they store information about the Householder transformations that will be later used to apply them. The continuous lines surround the blocks involved in the current task. To reduce the size of this graphic, it only shows the factorization of the first and second block columns.

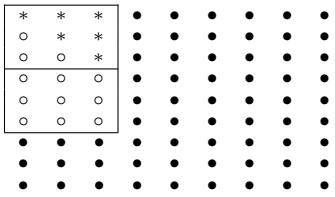
In the processing of the first column, as there are no previous columns, the work to do is just to nullify all the elements below the main diagonal. This process is performed with three tasks (tasks 1, 2, and 3). The first task nullifies elements below the diagonal in A_{00}. The second and third tasks nullify elements in A_{10} and A_{20}, respectively. To nullify those two blocks, these two tasks must also update the A_{00} block. In the processing of the second column, the first work to do is to apply previous transformations to the current block column (tasks 4, 5, and 6). Then, the elements below the diagonal in blocks A_{11} and A_{21} must be nullified (tasks 7 and 8).

Table 9.2 illustrates all the tasks generated and executed by the Algorithm-By-Blocks for computing the QR factorization for the previous case (and also for the general cases $m = n = 3b$, where b is the block size). As can be seen, in this case the full factorization comprises 14 tasks. The effect of the first eight tasks is shown in Figure 9.1. The remaining tasks (not shown in the graphic) proceed in an analogous way on the third block column: First, the transformations obtained when annihilating the elements below the diagonal in the first block column are applied to the third block column. Second, the transformations obtained when annihilating the elements below the diagonal in the second block column are applied to the third block column. Finally, the elements below the diagonal in the third block column are annihilated. The QR factorization of a matrix of any dimension only requires the following four generic tasks:

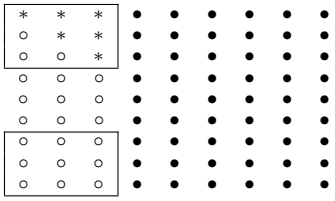
- *Compute_ dense_ QR(A, S)*: This task nullifies all the elements below the diagonal of input/output block A. The output is two-fold: The first is the updated matrix A, and the second is the S factor. The upper triangular part of A contains the updated R triangular factor. The strictly lower triangular part of A contains the Householder reflectors generated



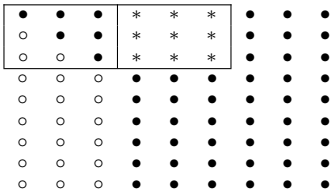
(1) After Compute_QR(A_{00})



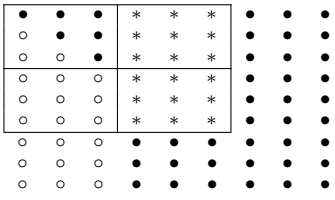
(2) After Compute_TD_QR(A_{00}, A_{10})



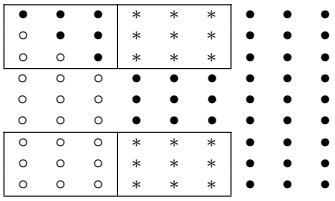
(3) After Compute_TD_QR(A_{00}, A_{20})



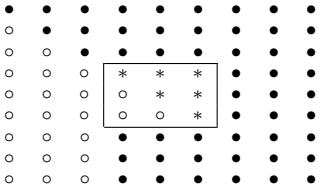
(4) After Apply_left_Qt_of_Dense_QR(A_{00}, A_{01})



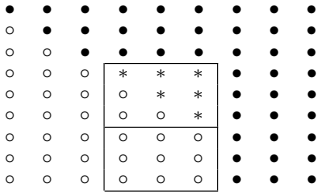
(5) After Apply_left_Qt_of_TD_QR(A_{00}, A_{10}, A_{01}, A_{11})



(6) After Apply_left_Qt_of_TD_QR(A_{00}, A_{20}, A_{01}, A_{21})



(7) After Compute_dense_QR(A_{11}, S_{11})



(8) After Compute_TD_QR(A_{11}, A_{21}, S_{21})

Figura 9.1: An illustration of the first tasks performed by an algorithm-by-blocks for computing the QR factorization. The '•' symbol represents a non-modified element by the current task, '∗' represents a modified element by the current task, and '○' represents a nullified element (by the current task or by a previous task). The continuous lines surround the blocks involved in the current task.

Operation	Operands	
	Out	In
Comp_dense_QR	A_{00} S_{00}	A_{00}
Comp_TD_QR	A_{00} A_{10} S_{10}	A_{00} A_{10}
Comp_TD_QR	A_{00} A_{20} S_{20}	A_{00} A_{20}
Apply_left_Qt_of_dense_QR	A_{01}	A_{00} S_{00} A_{01}
Apply_left_Qt_of_TD_QR	A_{01} A_{11}	A_{10} S_{10} A_{01} A_{11}
Apply_left_Qt_of_TD_QR	A_{01} A_{21}	A_{20} S_{20} A_{01} A_{21}
Comp_dense_QR	A_{11} S_{11}	A_{11}
Comp_TD_QR	A_{11} A_{21} S_{21}	A_{11} A_{21}
Apply_left_Qt_of_dense_QR	A_{02}	A_{00} S_{00} A_{02}
Apply_left_Qt_of_TD_QR	A_{02} A_{12}	A_{10} S_{10} A_{02} A_{12}
Apply_left_Qt_of_TD_QR	A_{02} A_{22}	A_{20} S_{20} A_{02} A_{22}
Apply_left_Qt_of_dense_QR	A_{12}	A_{11} S_{11} A_{12}
Apply_left_Qt_of_TD_QR	A_{12} A_{22}	A_{21} S_{21} A_{12} A_{22}
Comp_dense_QR	A_{22} S_{22}	A_{22}

Tabla 9.2: List of tasks generated by the Algorithm-by-blocks for computing the QR factorization when $m = n = 3b$, where b is the block size.

in this QR factorization. Matrix S contains the S factors, also required to apply the transformations obtained in this task.

- *Apply_ left_ Qt_ of_ dense_ QR(Y, S, C)*: The input data of this task are matrices Y (the Householder reflectors) and S (the S factors), the output of the previous task. Given these two input matrices Y and S, this task applies those transformations to input/output block C.

- *Compute_ TD_ QR(T, D, S)*: The input data of this task are matrices T and D (triangular and dense, respectively, and hence the acronym TD). This task nullifies all the elements in block D and accordingly updates block T. The output is three-fold: The first output is matrix T (containing the updated triangular factor), the second output is matrix D (containing the Householder reflectors), and the third output is matrix S (containing the S factors).

- *Apply_ left_ Qt_ of_ TD_ QR(D, S, F, G)*: The input data of this task are the input matrix D (the Householder reflectors) and S (the S factors). Both of them are the output of the previous task, i.e. the computation of the QR factorization of a triangular-dense factor. This task correspon-

dingly updates input/output matrices F and G with those transformations.

These four generic tasks will be employed when computing the QR factorization ($A = QR$) and when computing the solution of the linear system $X = R^{-1}(Q^T B)$.

System solving

When a linear system of equations $AX = B$ must be solved by using the QR factorization, the first stage is obviously to compute the factorization $A = QR$. The second stage is the following computation: $X = R^{-1}(Q^T B)$, where Q^T is the transpose of Q.

The first sub-step of the second stage ($X = R^{-1}(Q^T B)$) is to compute the product $Q^T B$. As usual in linear algebra, matrix Q (or its transpose) is not explicitly built because of the large cost (in both space and time) of the building operation and the even larger computational cost of the following matrix-matrix multiply. Instead, the transpose of matrix Q will be implicitly applied by using the Householder reflectors and the S factors previously obtained in the QR factorization.

The second sub-step of the second stage ($X = R^{-1}(Q^T B)$) is to multiply the inverse of R and the result of the previous sub-step ($Q^T B$). As usual in linear algebra, to reduce the computational cost, the inverse of R is not explicitly computed, and instead a linear backward substitution is applied. A block row algorithm for the backward substitution has been employed in order to both increase the locality and minimize the number of blocks being written (if a cache of blocks is employed).

Table 9.3 illustrates all the tasks generated and executed by the Algorithm-By-Blocks for computing $X = R^{-1}(Q^T B)$ when the QR factorization has been previously computed for the case $m = n = 3b$, where b is the block size.

9.3 Results

In this section, the precision and the speed of our new implementations are assessed. The first subsection describes the precision study, whereas the second subsection describes the performance study. In all the experiments we used double-precision arithmetic with double-precision real matrices.

Operation	Operands	
	Out	In
Apply_left_Qt_of_dense_QR	B_{00}	A_{00} S_{00} B_{00}
Apply_left_Qt_of_TD_QR	B_{00} B_{10}	A_{10} S_{10} B_{00} B_{10}
Apply_left_Qt_of_TD_QR	B_{00} B_{20}	A_{20} S_{20} B_{00} B_{20}
Apply_left_Qt_of_dense_QR	B_{10}	A_{11} S_{11} B_{10}
Apply_left_Qt_of_TD_QR	B_{10} B_{20}	A_{21} S_{21} B_{10} B_{20}
Apply_left_Qt_of_dense_QR	B_{20}	A_{22} S_{22} B_{20}
Trsm_lunn ($B = \text{upper}(A)^{-1}B$)	B_{20}	A_{22} B_{20}
Gemm_nn_mo ($C = -AB + C$)	B_{10}	B_{10} A_{12} B_{20}
Trsm_lunn ($B = \text{upper}(A)^{-1}B$)	B_{10}	A_{11} B_{10}
Gemm_nn_mo ($C = -AB + C$)	B_{00}	B_{00} A_{01} B_{10}
Gemm_nn_mo ($C = -AB + C$)	B_{00}	B_{00} A_{02} B_{20}
Trsm_lunn ($B = \text{upper}(A)^{-1}B$)	B_{00}	A_{00} B_{00}

Tabla 9.3: List of tasks generated by the Algorithm-by-blocks for solving a linear system using a previously computed QR factorization when $m = n = 3b$, where b is the block size.

	Resolution			
	64^2	256^2	384^2	512^2
Residual	$2.09 \cdot 10^{-13}$	$1.50 \cdot 10^{-12}$	$2.69 \cdot 10^{-12}$	$6.42 \cdot 10^{-12}$

Tabla 9.4: Evolution of the relative residual.

9.3.1 Precision and image quality study

In a preliminary test to check the validity of this method, the Forbild Head Phantom (FORBILD Phantom Group 2021) for different resolutions (from 64^2 to 512^2) was projected and reconstructed. Table 9.4 shows the relative residual $r = ||AX - B||_F / ||A||_F$ for those resolutions. As can be seen, the data shows that the method is numerically stable and the solution obtained is very accurate even on the highest resolution. Although the residual grows with the resolution, it is still low, so higher image resolutions could be reached if needed. Table 9.5 shows the quality metrics results, as an average of the quality of every slice of the phantom. It is worth mentioning that for the simulation of the acquisition, the images are always represented taking attenuation coefficients as gray values. The attenuation coefficients are expressed relative to that of water. In the mathematical phantom, the range of the gray values is between 0 (air, CT number -1000) and 1.8 (bone, CT number 800). This is the range of values used for calculating the quality metrics, and the same conversion from CT number

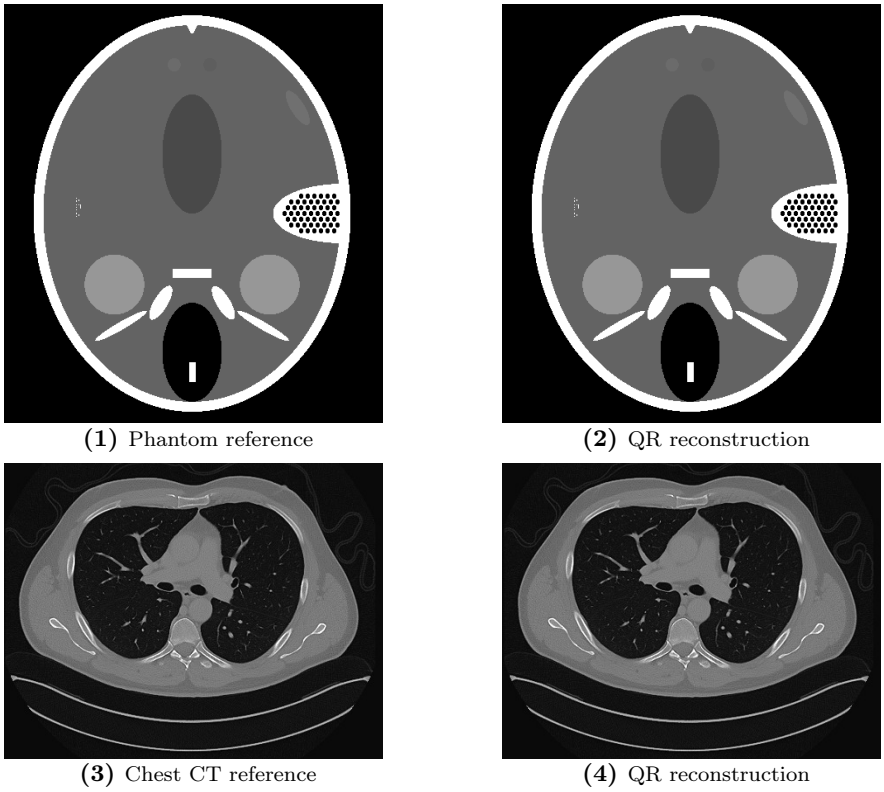
	Resolution			
	64^2	256^2	384^2	512^2
PSNR	258	228	220	204
SSIM	1	1	1	1

Tabla 9.5: Average Reconstruction Image Quality.

to attenuation coefficient is also performed for real CT images. The number of slices for every resolution in these tests is 32 for the 64^2 pixels resolution, 128 for the 256^2 and so on. The SSIM metric is equal to 1 for every image resolution, which indicates we are not losing any internal structure of the images. The PSNR is high for every case, with results always above 200, although it is higher for the smaller resolutions. In other works as (Chillarón y col. 2017) where we worked with iterative methods, we considered reconstructions with a PSNR of around 60 to be high-quality when working with this particular mathematical phantom.

Figures 9.2.1 and 9.2.2 show the central slice of the phantom and our reconstruction for a resolution with 512×512 pixels, the higher resolution we have reconstructed. As can be observed, the images are identical.

A randomly chosen collection of real CT images from the dataset DeepLesion (Yan y col. 2018) was also tested. The selected images, which had 512×512 pixels, were projected with Joseph's method and used as reference. With these images from the dataset, the average PSNR of the reconstructions for 2048 slices corresponding to different studies is 220, and the SSIM is 1. Figures 9.2.3 and 9.2.4 show that our method achieves really high-quality reconstructions, even though these images are much more complex than the phantom.



(1) Phantom reference **(2)** QR reconstruction

(3) Chest CT reference **(4)** QR reconstruction

Figura 9.2: CT images.

Figure 9.3 shows a comparison of the reconstructions obtained with both the QR and LSQR method proposed in (Chillarón, Vidal y Verdú 2020a) using 260 views, as well as with the FBP method with the Ram-Lak filter using different number of views along the 360 degrees of rotation. In this figure, it can be observed how the QR and the LSQR reconstructions are similar, although the latter is smoother since our method includes a regularization technique that smooths the image. The QR reconstruction is identical to the reference image.

As for the FBP reconstructions, it can be seen that using the same 260 views, the resulting image contains artifacts due to an insufficient number of projections. The artifacts diminish when we increment the number of projections taken, until they can't be appreciated anymore (Figure 9.3.6). This is due to the Nyquist-Shannon sampling theorem, that implies the ratio between the number of projections and the number of samples must be in the order of $\pi/2$

(as demonstrated in (Kak y Slaney 2001; Kharfi 2013)). In this particular case, since we have 1025 detectors (samples), the minimum number of projections or views needed is 1601 to get an image without artifacts due to undersampling.

Table 9.6 shows the metrics for this particular image with every method. The results show how the quality improves when we increase the number of views with the FBP method. The best result with the FBP has worse quality than the LSQR reconstruction, but the SSIM is fairly close so the image is similar in terms of internal structures, although it has more noise. The QR reconstructions are much better than the others, since they are almost identical to the reference image, with a MSE of $9.56 \cdot 10^{-24}$.

To conclude, it can be said that to reconstruct the images, having full rank in the sparse weights matrix when employing algebraic methods is not equivalent to having enough projections when using analytical methods, since when employing the same number of views many more artifacts are obtained with the latter.

	PNSR	SSIM	MSE
QR	235.8	1	$9.56 \cdot 10^{-24}$
LSQR	55.1	0.986	$2.22 \cdot 10^{-05}$
FBP 260	35.76	0.827	$9.62 \cdot 10^{-04}$
FBP 360	37.97	0.920	$5.78 \cdot 10^{-04}$
FBP 720	39.25	0.969	$4.30 \cdot 10^{-04}$
FBP 1610	39.31	0.973	$4.24 \cdot 10^{-04}$

Tabla 9.6: Quality metrics results for several reconstruction methods.

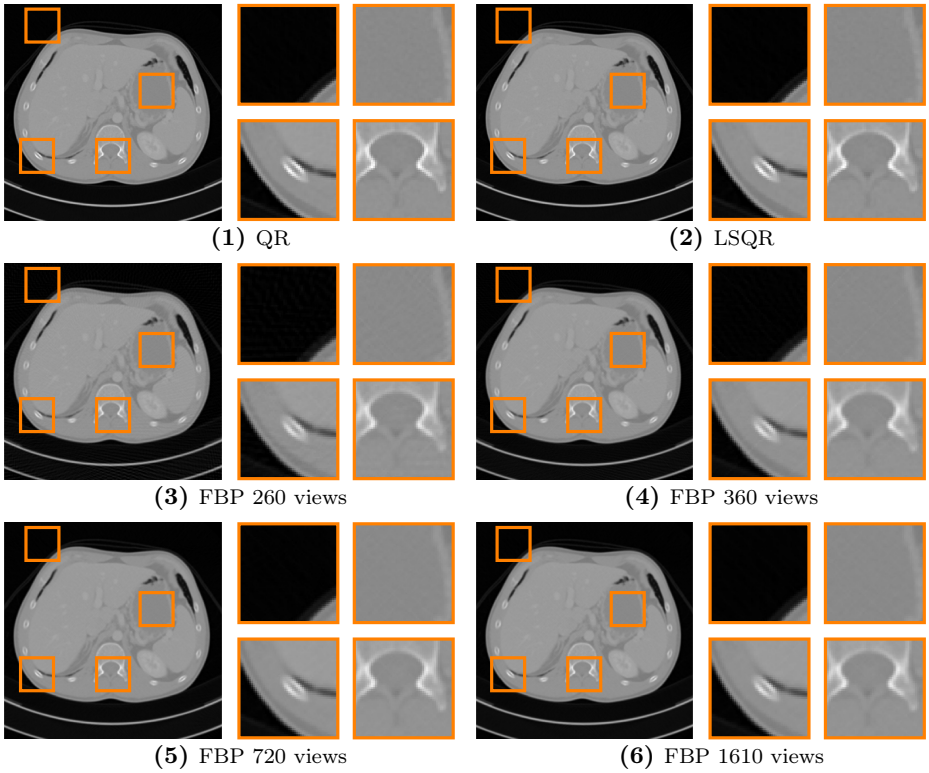


Figura 9.3: Reconstruction using different methods.

9.3.2 Performance study

The computer used in the performance experiments featured one Intel i7-7800X® CPU (6 physical cores) and 128 GiB of RAM in total. The clock frequency of the processor was 3.50 GHz, and the so-called *Max Turbo Frequency* was 4.00 GHz. In addition to one small SSD for storing the operating system and programming tools, the computer had two disks that were employed in the experiments, both with a capacity of 2 TB: One Hard Disk Drive (HDD) and one Solid-State Drive (SSD) with an M.2 connector. The HDD (spinning disk) was a Toshiba DT01ACA200 (Firmware MX4OABB0). The SSD was a Samsung SSD 970 EVO 2TB (Firmware 1B2QEXE7). According to the Linux operating system `hdparm` tool, the read speed of the first one was about 191.43 MB/s, whereas the read speed of the second one was about was 2427.50 MB/s. This is an upper-middle desktop personal computer and its current price is only about a few thousand dollars. Its OS was GNU/Li-

nux (kernel version 3.10.0-862.14.4.el7.x86_64). GCC compiler (version 4.8.5 20150623) was used. Intel(R) Math Kernel Library (MKL) Version 2018.0.2 Product Build 20180127 for Intel(R) 64 architecture was employed for solving some advanced linear algebra problems. Our new implementations were coded with the `libflame` (Release 11104) high-performance library, which employed Intel's MKL for performing the small- and medium-sized basic linear algebra computations.

Because of the variability of the experimental running time on some computers, when solving linear systems three experiments were ran, and the average values were reported. Nevertheless, we must say that the three obtained times were similar on the assessed architecture. All the experiments reported show only the time required by the computation $X = R^{-1}(Q^T B)$, since the QR factorization can be computed only once and then employed for many different images.

Unless explicitly stated otherwise, all the experiments employed six threads (and therefore six cores) for computation since the computer had six cores, the only exception being the codes with overlapping of computation and I/O. In this case, five threads (and five cores) were employed for computation and one thread (one core) was employed for disk I/O tasks.

We have assessed four configurations, which are obtained as the combinations of two OOC AB methods (non-overlapping or basic OOC AB, and overlapping OOC AB) and two types of disks (HDD and SSD). The assessed four configurations were the following:

- B-OOC + HDD: The basic (or non-overlapping) Out-Of-Core Algorithm-by-Blocks for solving the linear system was employed on the HDD described above. This is also called the initial configuration.
- O-OOC + HDD: The Out-Of-Core Algorithm-by-Blocks with overlapping of computation and I/O for solving the linear system was employed on the HDD described above.
- B-OOC + SSD: The basic (or non-overlapping) Out-Of-Core Algorithm-by-Blocks for solving the linear system was employed on the SSD described above.
- O-OOC + SSD: The Out-Of-Core Algorithm-by-Blocks with overlapping of computation and I/O for solving the linear system was employed on the SSD described above.

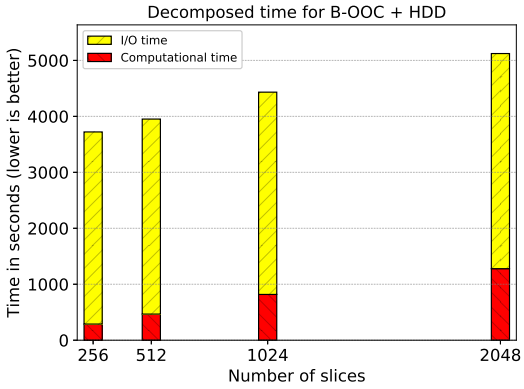


Figura 9.4: Overall times and decomposed times of the initial configuration (B-OOC + HDD) for solving a linear system with A of dimension 266,500 × 262,144, and B of dimension 266,500 × k, where k is the number of slices.

In all our implementations we employed a block size 10240 for the OOC computations (the number of rows and columns of every square block, kept in a different file), since this size usually renders good results on all the assessed code variants (Marqués y col. 2009; Quintana-Ortí y col. 2012; Marqués y col. 2011). In our codes, the block size employed inside every task to process the blocks once they are stored in RAM was 128, since this size usually renders good performances when processing matrices of size 10240. In the rest of the codes not developed by us (matrix-matrix products, etc.), the block size was determined by the library that performed that task (usually Intel's MKL).

Figure 9.4 shows the overall times and the decomposed times of the initial configuration (B-OOC + HDD, that is, the basic or non-overlapping OOC Algorithm-by-Blocks on the HDD) for solving a linear system with A of dimension 266,500 × 262,144, and B of dimension 266,500 × k, where k is the number of slices. The aim of this plot was to assess if the process was feasible, and to determine the main bottleneck of the application. For the system with 2048 slices, 2.50 seconds per slice were needed; for the system with 256 slices, 14.54 seconds per slice were needed. These times showed that the process was feasible, but the times were a bit high in some cases and very high in other cases. Moreover, the decomposition of the time showed that I/O times were very high, but they did not grow too much as the number of slices increased. Therefore, the main bottleneck of this problem was the I/O time, instead of the computational time. Then, adding more cores or several GPUs to the

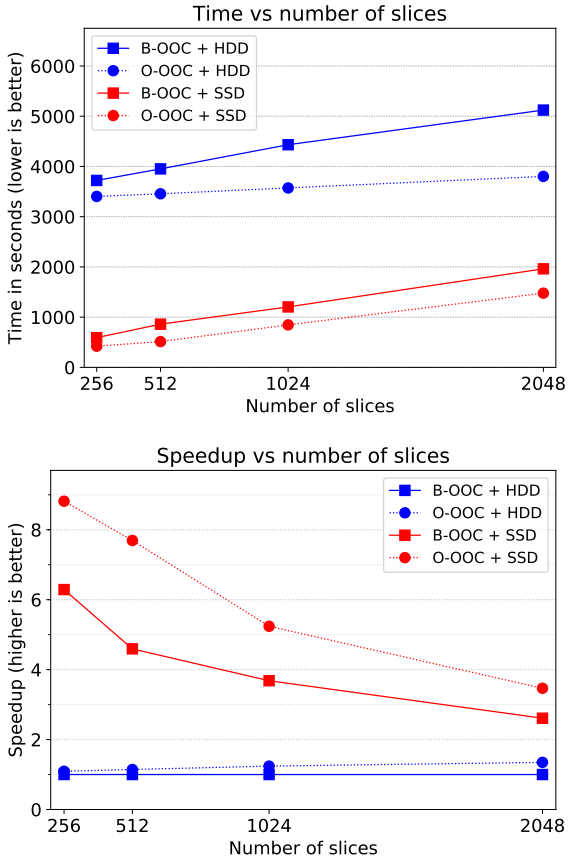


Figura 9.5: Time and speedups for the four configurations.

hardware configuration was not going to help in this case, and the focus should instead be on a fast disk.

Figure 9.5 compares the performances of the four configurations described above: Basic OOC AB on HDD, Overlapping OOC AB on HDD, Basic OOC AB on SSD, and Overlapping OOC AB on SSD. The top subplot shows the times in seconds (lower is better), whereas the bottom subplot shows the speedup (higher is better) with respect to the initial configuration (basic OOC AB on HDD). The speedup is computed as the quotient of the time obtained by the reference configuration and the time obtained by the new configuration. Thus,

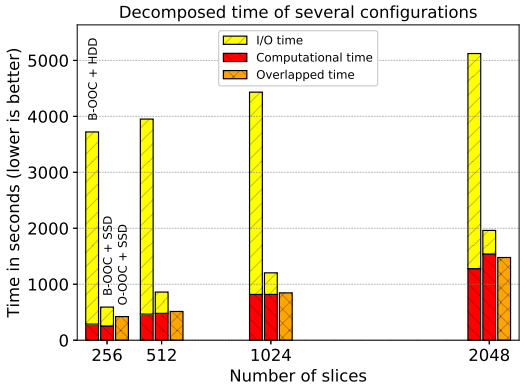


Figura 9.6: Overall times and decomposed times of three configurations for solving a linear system with A of dimension 266,500 × 262,144, and B of dimension 266,500 × k, where k is the number of slices.

this concept means how many times the new configuration is as fast as the reference configuration. Hence, the higher the speedups, the better the performances are. As the reference configuration is the initial one, in the bottom subplot the initial configuration will be shown as ones. As can be seen, the SSD greatly reduced the overall times and increased the speed by more than 6 times for the smallest case (256 slices) with respect to the initial configuration. The overlapping of computation and I/O further increased the speed up to nearly 9 times for the smallest case (256 slices). When the number of slices was high, the improvements were not so great but still very noticeable.

Figure 9.6 shows the overall times and the decomposed times for solving a linear system with A of dimension 266,500 × 262,144, and B of dimension 266,500 × k, where k is the number of slices, on three configurations: B-OOC + HDD, B-OOC + SSD, and O-OOC + SSD. The left bar for each number of slices shows the overall times and the decomposed times of the initial configuration (B-OOC + HDD). As can be seen, its main drawback is the high I/O cost because of using a HDD. The center bar for each number of slices shows the overall times and the decomposed times of a configuration similar to the previous one with an SSD (B-OOC + SSD). As can be seen, the high I/O cost has been greatly reduced. The right bar for each number of slices shows the overall times of the best configuration (O-OOC + SSD). As this configuration overlaps computation and I/O, the time cannot be decomposed. As can be seen, in most cases the I/O cost (the fast SSD) of the previous configuration is completely removed.

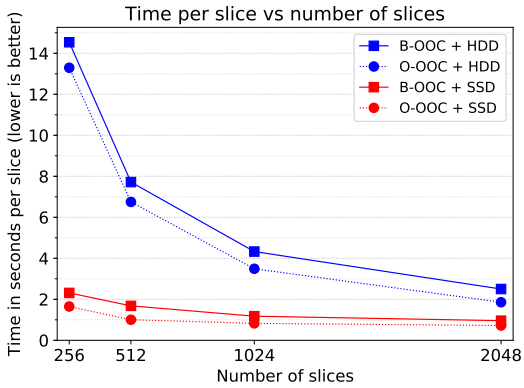


Figura 9.7: Time in seconds per slice for the four configurations.

Figure 9.7 shows the time in seconds required to compute one slice. As it shows, this time was not constant, and it depended somewhat on the number of slices: the more slices to compute, the lower the time per slice. Just consider that, regardless of the number of slices (even for just one slice), the whole factorized matrix A must be read from disk. Thus, this large cost becomes diluted as more slices are being computed. In the initial configuration (basic OOC AB and HDD) the time per slice greatly depended on the number of slices. In the best configuration (overlapping OOC AB on SSD) the time per slice is not so dependent on the number of slices.

Table 9.7 shows the time in seconds required to compute one slice for both the initial configuration and the most performant configuration. As can be seen, the range of the initial configuration is very wide (from 2.50 to 14.54 seconds), whereas the range of the most performant configuration is much narrower (from 0.72 to 1.65 seconds).

The weights matrix for the highest resolution in our experimental study required a storage of about 560 GB ($265{,}500 \times 262{,}144$ double precision elements). Besides the weights matrix, additional space (patient's data, final image, temporary data, application code, operating system, disk buffers and cache, etc.) makes the total size required by this problem even larger. As was told, the computer had 128 GB of RAM. However, only 32 GB were employed as a cache to store blocks of the weights matrix, leaving the rest for other purposes (operating system disk cache and buffers, etc.). We assessed another computer with 48 GB of RAM, and results were similar when using a similar number of cores, but we did not report its results because it was a much more expensive

	Number of slices			
Method	256	512	1024	2048
B-OOC + HDD	14.54	7.72	4.33	2.50
O-OOC + SDD	1.65	1.00	0.83	0.72

Tabla 9.7: Time in seconds per slice versus number of slices.

server. Hence, to obtain good performances, a smaller main memory could be used (thus reducing the total price), but a fast SSD is a strong requirement.

9.4 Discussion

In this paper, we present a direct algebraic method based on the QR factorization for reconstructing CT images efficiently on affordable computers. As we have shown, this method is numerically stable even for high resolutions provided the weights matrices have full rank. For this reason, our method employs more X-ray projections than the algebraic iterative methods, but fewer than the analytical methods. Although we have not measured the radiation dose, literature shows how introducing sparse-sampling CT scanners in the clinical practice could reduce the dose by reducing the exposure time.

With our proposed method, which uses a number of projections that guarantee the full rank of the weights matrix, high-quality images are obtained without requiring an a-priori knowledge or interaction with the patient. This method guarantees the non-creation of artifacts except those produced by problems on detectors, dispersion, movement, intensity of the source, etc., which can be corrected by filtering and segmentation techniques. In addition, our reconstructions achieve remarkable quality even for complex real CT images. It is worth mentioning we have not considered or removed the possible noise on the projections, which we consider unnecessary for this work since most modern CT scanners have their own algorithms to improve the projections and remove the artifacts on the sinograms when a scan is performed, so the data we get is already clean.

We have shown that an efficient reconstruction of CT images can be achieved using Out-Of-Core and Algorithm-By-Blocks techniques. By using our techniques, affordable computers with a price of about one order of magnitude lower can be successfully employed, because a large main memory (which is quite expensive) is not required, just a fast hard drive. For this reason, the

equipment needed to reconstruct the images is affordable and thus more accessible to the public. The type of hard drive can improve our reconstruction times drastically. When using a HDD, the performance is dominated by the I/O time, whereas when using an SSD the I/O time is greatly reduced and the performance is dominated by the computational time again.

Furthermore, the method that overlaps computation and I/O can further reduce the reconstructing time, thus making our method more competitive. We could perform an standard CT study with resolution 512^2 and 256 slices in about 7 minutes. We have also shown that the cost per slice is lower as the number of simultaneous slices to reconstruct is higher, which would be beneficial for full-body CT scans. Our proposed method is not as fast as the state-of-the-art fast backprojection techniques Miqueles, Koshev y Helou 2017; Koshev, Helou y Miqueles 2016, which can obtain a high-resolution 1024^2 pixels image in 0.565 seconds on an Intel Xeon 3.4 GHz processor when reconstructing only one slice. However, we believe that this performance difference is much smaller when reconstructing multiple slices (see Table 9.7). On the other hand, our proposed method provides exact solutions that avoid noise or artifacts, which can be a very interesting approach even if the computational complexity is slightly higher.

9.5 Conclusions

With the proposed QR factorization method and system solving using Out-Of-Core techniques we were able to reconstruct high-quality CT images using the minimum number of projections to have full rank. Our results show that efficiently computing high-quality reconstructions with direct algebraic methods on affordable equipment can be achieved since our approach relies on the cheaper hard drives, instead of the more expensive main memory. Using SSD storage we can further boost the performance of the method, reducing the I/O time significantly. Because of the stability of our method, we could increase the resolution of the images provided we had enough storage space, and solve larger systems getting valid results.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be constructed as a potential conflict of interest.

Acknowledgments

This research has been supported by "Universitat Politècnica de València", "Generalitat Valenciana" under PROMETEO/2018/035 and ACIF/2017/075, co-financed by FEDER and FSE funds, and the "Spanish Ministry of Science, Innovation and Universities" under Grant RTI2018-098156-B-C54 co-financed by FEDER funds.

Capítulo 10

Conclusiones y Trabajo Futuro

10.1 Conclusión

Durante el desarrollo de esta tesis se han explorado varios métodos algebraicos para el problema de resolución de imagen TC. Para ello, se ha simulado un escáner de tipo *fanbeam* con un solo array de detectores, reconstruyendo volúmenes corte a corte. Además, las proyecciones simuladas se han obtenido de forma dispersa, empleando una selección no equiangular con el fin de obtener la máxima información posible al disminuir el número de proyecciones.

Los métodos analizados corresponden a las variantes iterativas y directas, teniendo así varias posibilidades a la hora de resolver el sistema de ecuaciones que modela el problema de reconstrucción.

El método iterativo seleccionado ha sido el Least Squares QR (LSQR), que combinado con el filtro STF y la técnica de aceleración FISTA permite aproximar la solución de forma iterativa obteniendo una buena calidad de imagen incluso cuando el número de proyecciones es reducido. Con el fin de determinar los parámetros óptimos de reconstrucción se ha llevado a cabo un estudio multiparamétrico realizando tantas ejecuciones como combinaciones resultantes de variar los parámetros que influyen en la calidad y en el convergencia.

Puesto que el número de ejecuciones necesarias es elevado, el estudio ha sido diseñado para su ejecución de forma distribuida en una plataforma Grid.

Como se ha expuesto en el capítulo 4, el diseño de este tipo de análisis multiparamétrico en plataformas de cómputo distribuido puede ser muy útil ya que ahorra tiempo total de cómputo, a la vez que emplea recursos de diversas entidades puestos a disposición de la comunidad científica, lo cual permite ahorrar en coste. Sin embargo, no está libre de complicaciones, en especial cuando el software que se necesita emplear no es estándar, como sucede en este caso. Dichas complicaciones se han solucionado empleado dos aproximaciones distintas: realizando una instalación en espacio de usuario del Runtime de Matlab, o desplegando un contenedor Docker a través de udocker que ya contiene el Runtime. Como se ha mostrado en las conclusiones del capítulo 4, es ligeramente más eficiente la primera estrategia porque se aprovecha mejor la capacidad de cómputo de la máquina en cuestión. Sin embargo, la segunda puede ser la mejor opción en la mayoría de casos, cuando no se pueda instalar el software necesario directamente. Con todo ello, el estudio en Grid diseñado puede ser de gran utilidad para explorar y analizar el comportamiento de nuevos métodos en el futuro.

Para mejorar el comportamiento del método LSQR+STF+FISTA se ha realizado un estudio de filtros de imagen en el capítulo 5, donde se ha analizado el comportamiento de cuatro filtros tanto en sinogramas con reducción de proyecciones, como en las propias imágenes reconstruidas. Tal y como se ha mostrado en los resultados, el filtro que obtiene mejor calidad de imagen es el filtro Bilateral, por lo que se ha seleccionado para emplearlo como paso extra en el proceso de reconstrucción. Del análisis realizado con una imagen abdominal de TC real, se concluye que la introducción de este filtro en el proceso mejora la calidad de las imágenes obtenidas independientemente del número de proyecciones empleadas, por lo que es altamente recomendable emplearlo.

Por otra parte, se ha analizado el uso de métodos algebraicos directos para la resolución de los sistemas de ecuaciones lineales asociados al problema de reconstrucción de imagen TC. Los dos métodos estudiados han sido la factorización QR y la descomposición en valores singulares (SVD), tal y como se ha expuesto en el capítulo 6. En este capítulo se ha estudiado la factibilidad de emplear este tipo de métodos, llegando a la conclusión que el uso de memoria RAM para obtener las factorizaciones puede ser tan elevado que no permita llegar a las resoluciones más altas incluso cuando se emplean 250GB de memoria principal. Además, se ha comprobado que el método SVD requiere más memoria y tiempo que el QR. En cuanto a la calidad de las imágenes obtenidas se ha comprobado que es extremadamente alta (se podría decir que las

reconstrucciones son perfectas con respecto a la imagen de referencia) siempre y cuando la matriz del sistema sea de rango completo. Cuando tiene rango deficiente, se puede obtener una solución del sistema pero la imagen resultado es de poca calidad, siendo peor en este caso las obtenidas mediante el método QR. Por ello, se ha determinado que este tipo de métodos sólo se empleará cuando el número de proyecciones sea lo suficientemente alto para garantizar el rango completo. Como se ha estudiado, este número será 90 para resolución 256×256 y 260 para 512×512, lo que todavía deja un margen de reducción de dosis con respecto a los métodos analíticos.

Por los resultados obtenidos en dicho estudio preliminar, se han descartado el método SVD en favor del método QR, cuyo comportamiento se ha seguido analizando en el capítulo 7. En él, se ha valorado la posibilidad de emplear las reflexiones de Householder con el fin de no tener que formar la matriz Q de manera explícita, lo que permite ahorrar memoria. La implementación utilizada de este método es la incluida en la librería SuiteSparse, que aporta paralelismo multihilo a través de BLAS.

De los resultados extraídos se puede concluir que emplear las reflexiones de Householder es altamente beneficioso en el tiempo de factorización, obteniendo tiempos aproximadamente reducidos en un factor 3 con respecto a la factorización con la Q explícita. Además, se muestra cómo el aumento del número de hilos aporta un *SpeedUp* a las factorizaciones nada despreciable.

Por otra parte, se han comparado las reconstrucciones obtenidas por ambos métodos, resolviendo un solo corte, o 128 cortes (un volumen) en la misma ejecución. Como se ha demostrado, cuando se resuelve un solo lado derecho, la resolución con la Q explícita es ligeramente más rápida. Sin embargo, al aumentar el número de cortes emplear el método de reflexiones de Householder permite acelerar las reconstrucciones en un factor de entre 2.8 y 1.75. Además, se ha mostrado cómo aumentar el número de hilos empleados en esta etapa no obtiene mejores prestaciones.

En el capítulo 8 se ha realizado un estudio comparativo de la reconstrucción iterativa mediante LSQR+STF+FISTA y la resolución directa mediante QR. Dicho estudio ha sido realizado con una colección de imágenes de TC reales con el fin de determinar qué método tiene mejor comportamiento. Para el caso analizado, se han comparado todas las reconstrucciones obtenidas, correspondientes a imágenes de lesiones en diferentes zonas (hueso, riñón, pulmón, etc.), midiendo la calidad de imagen con las métricas de calidad así como analizando las lesiones de forma visual.

Se ha llegado a la conclusión que para problemas de rango completo, el método LSQR también obtiene reconstrucciones de alta calidad. Sin embargo, los niveles de ruido son más altos, y se pueden formar artefactos o un efecto de *oversmoothing* que empeora la imagen, lo cual no sucede con el método QR. Además, debido al ruido, puede haber estructuras que no sean visibles en ciertas ventanas por diferencias en sus valores HU. Con todo ello, se concluye que en el supuesto de tener rango completo, sigue siendo preferible el uso del método QR ya que es más directo, acotable y obtiene imágenes exactas con respecto a la de referencia. Sin embargo, el método LSQR podría permitir una mayor disminución de la dosis, con una calidad de imagen ligeramente inferior pero aún así aceptable.

Por último, en el capítulo 9 se ha presentado una implementación optimizada del método QR, mediante un algoritmo a bloques que emplea la técnica Out-Of-Core para reducir el uso de memoria RAM. Se han desarrollado dos variantes del método: una variante tradicional en la cual se leen los datos de disco a medida que se necesitan para un cálculo, y una en la cual se solapa el cálculo con la lectura/escritura de los datos. Esta versión con solapamiento se ha obtenido mediante el uso de un *runtime* que analiza las dependencias de datos y planifica las lecturas y escrituras para que los datos estén listos cuando se necesiten para el cálculo.

Puesto que el cuello de botella de los métodos Out-Of-Core es el tiempo de entrada/salida, esto resulta altamente beneficioso de cara el tiempo total. Por el mismo motivo, el uso de discos de estado sólido SSD en lugar de los discos tradicionales también puede aportar un aumento de las prestaciones, puesto que sus velocidades de lectura y escritura son mucho más altas. En los resultados se ha demostrado que solapar cálculos con operaciones de entrada y salida puede llegar a enmascarar el tiempo de lectura y escritura de los datos, obteniendo tiempos totales muy similares a los tiempos correspondientes al cálculo.

Además, se ha analizado la eficiencia de resolver distinto número de cortes o lados derechos, empleando desde 256 sinogramas a 2048. Pese a que el *SpeedUp* de la mejor variante (SSD con solapamiento) con respecto a la variante básica (HDD sin solapamiento) baja al aumentar el número de cortes, se observa que el tiempo por corte disminuye de 1.65 segundos al resolver 256 cortes, a 0.72 segundos al resolver 2048. Esto podría ser altamente beneficioso de cara a estudios TC de cuerpo completo.

Con todo ello, este estudio presenta una implementación optimizada del método QR para emplear técnicas Out-Of-Core que reducen el uso de memoria

RAM, y por tanto permiten resolver problemas de altas dimensiones en máquinas *multicore* de coste medio.

Mediante los estudios presentados en los diferentes capítulos se han analizado diferentes estrategias para resolver el problema de reconstrucción de TC de manera algebraica. Se ha demostrado cómo los métodos directos obtienen imágenes de mayor calidad, pero la cantidad de memoria RAM que requieren puede llegar a ser prohibitiva. Esto no sucede con los métodos iterativos, que pese a obtener aproximaciones de menor calidad, pueden trabajar con tamaño de problemas más grandes.

Además, los métodos iterativos permiten una mayor reducción del número de proyecciones ya que no requieren que la matriz que modela el escáner físico tenga rango completo. Según un estudio preliminar presentado al congreso nacional 46ª Reunión Anual de la Sociedad Nuclear Española que tendrá lugar en Octubre de 2021, con título "Estudio de estimación de dosis de radiación para TC de pocas vistas: simulación de Monte-Carlo", en el cual se ha modelado un escáner TC que realiza adquisiciones dispersas mediante métodos de MonteCarlo, la reducción de proyecciones empleadas se traduce literalmente en el mismo porcentaje de dosis reducida si las proyecciones son seleccionadas de forma equiangular. De esta forma, si mediante un método analítico se emplean 1610 proyecciones como en el capítulo 9, y 260 mediante uno algebraico, esto supone un 83% menos de dosis depositada en el paciente. Empleando pasos no equidistantes, se podría reducir la dosis depositada en los distintos órganos, para conseguir proteger aquellos más vulnerables o que no se quieren estudiar. Por tanto, ambas estrategias analizadas lograrían una gran reducción de dosis con respecto a los métodos tradicionales, siendo el LSQR el que menos vistas necesita, y el QR el que mejor calidad obtiene y más consistente es por no ser iterativo y, por tanto, no tener que lograr convergencia.

Con respecto a los tiempos de reconstrucción, se ven notablemente reducidos con la implementación Out-Of-Core del método QR, pudiéndose obtener un estudio de 256 imágenes en aproximadamente 7 minutos. En otro estudio presentado en el congreso nacional 46ª Reunión Anual de la Sociedad Nuclear Española con título "Reconstrucción de imagen TC con pocas vistas por bloques usando GPU: rendimiento y calidad" , en el que se realiza una implementación paralela a bloques de método LSQR tanto para CPU como para GPU se están obteniendo tiempos por corte de 43 segundos para CPU y 5 segundos empleando GPU para resolver 64 cortes de manera simultánea. Esta implementación no está tan optimizada como la anterior, pero aún así los tiempos obtenidos mediante GPU son reducidos. Aunque los tiempos todavía no son

comparables a los métodos analíticos, consideramos que ya están en un rango aceptable para considerar su aplicación en la práctica clínica.

10.2 Trabajo Futuro

EL trabajo futuro de esta línea de investigación consiste por una parte en la optimización de los métodos estudiados, realizando una implementación para GPU del método QR Out-Of-Core, en la que se está trabajando y podría reducir los tiempos por corte a 0.1 segundos, lo cual permitiría obtener un volumen de 256 cortes en 25 segundos. Además, se pretende mejorar la implementación del método LSQR en GPU para obtener mejores prestaciones para que pueda ser comparable con el método directo.

Además, se pretenden adaptar todas las técnicas anteriores al modelo de escáner *cone-beam*, que emplea un haz cónico para proyectar volúmenes sin necesidad de desplazamientos como los escáneres que emplean haz en abanico. Este tipo de escáner se emplea en mamografías y en TC dental, y nuestras técnicas podrían ayudar a reducir su dosis.

Por otra parte, se pretende colaborar con hospitales para verificar el correcto funcionamiento de los métodos con sinogramas reales, ya que hasta el momento hemos trabajado con sinogramas simulados, tanto de fantomas como de imágenes reales. Para ello, se ha solicitado un proyecto de colaboración, cuyo entregable final sería un programa con interfaz gráfica que integrara nuestras técnicas de reconstrucción con un menú de visualización y análisis de calidad para poder poner a disposición del personal clínico una versión beta de la aplicación de métodos algebraicos para reducción de dosis en imagen TC.

Por último, puesto que la introducción de la Inteligencia Artificial al campo de tratamiento de imagen médica está demostrando tener mucho potencial, se quieren incorporar técnicas de IA para mejorar la calidad de las reconstrucciones y así poder lograr mayores reducciones de dosis. Para ello, es necesario contar con un gran volumen de datos, puesto que es esencial para poder entrenar las redes neuronales. Es por eso que también se ha solicitado un proyecto de colaboración con un hospital para poder obtener una base de datos de sinogramas e imágenes TC asociadas, obtenidos de forma tradicional y mediante diferentes estrategias de reducción de dosis, que conformen una base de aprendizaje que pueda ser utilizada para investigar sobre métodos basados en IA para maximizar la calidad de imagen y minimizar la dosis empleada.

Bibliografía

Aarle, Wim van y col. (2016). "Fast and flexible X-ray tomography using the ASTRA toolbox". En: *Opt. Express* 24.22, págs. 25129-25147. DOI: 10.1364/OE.24.025129 (vid. pág. 19).

Agulleiro, JI y col. (2010). "Vectorization with SIMD extensions speeds up reconstruction in electron tomography". En: *Journal of structural biology* 170.3, págs. 570-575 (vid. pág. 19).

Andersen, A. H. (1989). "Algebraic reconstruction in CT from limited views". En: *IEEE Transactions on Medical Imaging* 8.1, págs. 50-55 (vid. págs. 70, 109, 141, 160).

Andersen, A. H. y A. C. Kak (1984). "Simultaneous algebraic reconstruction technique (SART): a superior implementation of the ART algorithm". En: *Ultrasonic Imaging* 6.1, págs. 81-94 (vid. págs. 70, 109, 125, 141, 160).

Anderson, E. y col. (1992). *LAPACK Users' Guide*. Philadelphia: SIAM (vid. pág. 60).

Balay, Satish y col. (2021). *PETSc Users Manual*. Inf. téc. ANL-95/11 - Revision 3.15. Argonne National Laboratory (vid. pág. 61).

Barrett, Julia F y Nicholas Keat (2004). "Artifacts in CT: recognition and avoidance". En: *Radiographics* 24.6, págs. 1679-1691 (vid. págs. 91, 111).

Beck, A. y M. Teboulle (2009). "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". En: *SIAM Journal on Imaging Sciences* 2.1, págs. 183-202 (vid. págs. 42, 71, 92, 99, 144).

Berry, Michael W, Shakhina A Pulatova y GW Stewart (2005). "Algorithm 844: Computing sparse reduced-rank approximations to sparse matrices". En: *ACM Transactions on Mathematical Software (TOMS)* 31.2, págs. 252-269 (vid. pág. 113).

Bientinesi, Paolo, Enrique S. Quintana-Ortí y Robert A. van de Geijn (2005). "Representing Linear Algebra Algorithms in Code: The FLAME Application Programming Interfaces". En: *ACM Trans. Math. Soft.* 31.1, págs. 27-59 (vid. pág. 63).

Blackford, L Susan y col. (2002). "An updated set of basic linear algebra subprograms (BLAS)". En: *ACM Transactions on Mathematical Software* 28.2, págs. 135-151 (vid. pág. 58).

Bray, Freddie y col. "Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries". En: *CA: A Cancer Journal for Clinicians* 68.6 (), págs. 394-424. DOI: 10.3322/caac.21492. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.3322/caac.21492 (vid. pág. 141).

Brooks, R.A y G. Di Chiro (1976). "Principles of computer assisted tomography (CAT) in radiographic and radioisotopic imaging". En: *Physics in Medicine and Biology* 21.5, págs. 689-732 (vid. págs. 70, 110, 124).

Brown, Robert W y col. (2014). *Magnetic resonance imaging: physical principles and sequence design*. John Wiley & Sons (vid. pág. 124).

Buzzi, AE y MV Suárez (2013). "Tomografía lineal: nacimiento, gloria y ocaso de un método". En: *Revista argentina de radiología* 77.3, págs. 0-0 (vid. pág. 26).

Chillarón, Mónica, Vicente Vidal y Gumersindo Verdú (2019). "Parallel CT Reconstruction for Multiple Slices Studies with SuiteSparseQR Factorization Package". En: *Computational Science – ICCS 2019*. Cham: Springer International Publishing, págs. 160-169 (vid. pág. 121).

Chillarón, Mónica y col. (2018). "CT Medical Imaging Reconstruction Using Direct Algebraic Methods with Few Projections". En: *Computational Science – ICCS 2018*. Cham: Springer International Publishing, págs. 334-346 (vid. págs. 91, 105, 125, 132, 142, 153, 161).

Chillarón, Mónica y col. (2020). "Computed tomography medical image reconstruction on affordable equipment by using Out-Of-Core techniques". En: *Computer methods and programs in biomedicine* 193, pág. 105488 (vid. pág. 155).

Chillarón, Mónica, Vicente Vidal y Gumersindo Verdú (2020a). "CT image reconstruction with SuiteSparseQR factorization package". En: *Radiation Physics and Chemistry* 167, pág. 108289. DOI: https://doi.org/10.1016/j.radphyschem.2019.04.039 (vid. págs. 137, 161, 173).

Chillarón, Mónica, Vicente Vidal y Gumersindo Verdú (2020b). "Evaluation of image filters for their integration with LSQR computerized tomography reconstruction method". En: *PLOS ONE* 15.3, págs. 1-14. DOI: 10.1371/journal.pone.0229113 (vid. pág. 85).

Chillarón, Mónica y col. (2017). "Combining Grid Computing and Docker Containers for the Study and Parametrization of CT Image Reconstruction Methods". En: *Procedia Computer Science* 108. International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland, págs. 1195 -1204. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2017.05.065 (vid. págs. 65, 89, 91, 101, 111, 125, 127, 144, 148, 153, 160, 172).

Cho, Eun-Suk y col. (2012). "Cerebral computed tomography angiography using a low tube voltage (80 kVp) and a moderate concentration of iodine contrast material: a quantitative and qualitative comparison with conventional computed tomography angiography". En: *Investigative radiology* 47.2, págs. 142-147 (vid. pág. 9).

Clarke, RH y Jack Valentin (2009). "The History of ICRP and the Evolution of its Policies: Invited by the Commission in October 2008". En: *Annals of the ICRP* 39.1, págs. 75-110 (vid. pág. 25).

Coban, S.B. y W.R.B. Lionheart (2014). "Regularised GMRES-type Methods for X-Ray Computed Tomography". En: *he Third International Conference*

on Image Formation in X-Ray Computed Tomography. Salt Lake City, Utah, USA (vid. pág. 17).

Cools, Siegfried y col. (2015). "A multi-level preconditioned Krylov method for the efficient solution of algebraic tomographic reconstruction problems". En: *Journal of Computational and Applied Mathematics* 283, págs. 1-16. ISSN: 0377-0427. DOI: https://doi.org/10.1016/j.cam.2014.12.044 (vid. pág. 17).

Davis, Timothy A. (dic. de 2011). "Algorithm 915, SuiteSparseQR: Multifrontal Multithreaded Rank-revealing Sparse QR Factorization". En: *ACM Trans. Math. Softw.* 38.1, 8:1-8:22. ISSN: 0098-3500. DOI: 10.1145/2049662.2049670 (vid. págs. 60, 109, 111, 126, 129, 142).

Davis, Timothy A., Patrick R. Amestoy y Iain S. Duff (2021). *SuiteSparse: A suite of sparse matrix packages* (vid. pág. 60).

D'Azevedo, Ed y Jack Dongarra (dic. de 2000). "Design and implementation of the parallel out-of-core ScaLAPACK LU, QR, and Cholesky factorization routines". En: *Concurrency - Practice and Experience* 12, págs. 1481-1493 (vid. pág. 165).

De González, Amy Berrington y col. (2009). "Projected cancer risks from computed tomographic scans performed in the United States in 2007". En: *Archives of internal medicine* 169.22, págs. 2071-2077 (vid. pág. 160).

De Man, Quinten y col. (2019). "A two-dimensional feasibility study of deep learning-based feature detection and characterization directly from CT sinograms". En: *Medical physics* 46.12, e790-e800 (vid. pág. 18).

Dong, Jian y col. (2019). "Evolution from total variation to nonlinear sparsifying transform for sparse-view CT image reconstruction". En: *bioRxiv*. DOI: 10.1101/785261. eprint: https://www.biorxiv.org/content/early/2019/09/27/785261.full.pdf (vid. pág. 10).

Dongarra, Jack J. y col. (1988). "An Extended Set of FORTRAN Basic Linear Algebra Subprograms". En: *ACM Trans. Math. Soft.* 14.1, págs. 1-17 (vid. pág. 59).

Dongarra, Jack J. y col. (1990). "A Set of Level 3 Basic Linear Algebra Subprograms". En: *ACM Trans. Math. Soft.* 16.1, págs. 1-17 (vid. pág. 59).

Egi.eu (2021). *EGI Fundation Web Site*. `http://www.egi.eu/` (vid. pág. 54).

Feldkamp, Lee A, LC Davis y James W Kress (1984). "Practical cone-beam algorithm". En: *J Opt Soc Am* 1.6, págs. 612-619 (vid. pág. 141).

Flohr, Thomas (2013). "CT systems". En: *Current Radiology Reports* 1.1, págs. 52-63 (vid. pág. 28).

Flores, L., V. Vidal y G. Verdú (2015). "Iterative reconstruction from few-view projections". En: *Procedia Computer Science* 51, págs. 703-712 (vid. págs. 39, 71, 80, 89, 91, 111, 125, 127, 141, 144, 160).

Flores, Liubov A y col. (2014). "Parallel CT image reconstruction based on GPUs". En: *Radiation Physics and Chemistry* 95, págs. 247-250 (vid. págs. 89, 91, 111, 125, 141, 144, 160).

Flynn, Michael J (1972). "Some computer organizations and their effectiveness". En: *IEEE transactions on computers* 100.9, págs. 948-960 (vid. pág. 48).

FORBILD Phantom Group (2021). *FORBILD Head Phantom*. `http://www.imp.uni-erlangen.de/forbild/english/results/index.htm` (vid. págs. 72, 91, 110, 171).

Funama, Yoshinori y col. (2011). "Combination of a low tube voltage technique with the hybrid iterative reconstruction (iDose) algorithm at coronary CT angiography". En: *Journal of computer assisted tomography* 35.4, pág. 480 (vid. pág. 15).

Ge, Yongshuai y col. (2020). "ADAPTIVE-NET: Deep computed tomography reconstruction network with analytical domain transformation knowledge". En: *Quantitative imaging in medicine and surgery* 10.2, pág. 415 (vid. pág. 18).

Gervaise, Alban y col. (2012). "CT image quality improvement using adaptive iterative dose reduction with wide-volume acquisition on 320-detector CT". En: *European radiology* 22.2, págs. 295-301 (vid. pág. 15).

Golub, Gene y William Kahan (1965). "Calculating the singular values and pseudo-inverse of a matrix". En: *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis* 2.2, págs. 205-224 (vid. págs. 39, 113, 144).

Golub, Gene H, Per Christian Hansen y Dianne P O'Leary (1999). "Tikhonov regularization and total least squares". En: *SIAM journal on matrix analysis and applications* 21.1, págs. 185-194 (vid. pág. 18).

Golub, Gene H. y James M. Ortega (1993). *Scientific Computing: An Introduction with Parallel Computing*, pág. 442. ISBN: 0122892534 (vid. págs. 109, 127, 143).

Golub, Gene H y Charles F Van Loan (2013). *Matrix Computations*, pág. 780. ISBN: 9781421407944 (vid. págs. 110, 111, 126).

Gomes, Jorge y col. (2018). "Enabling rootless Linux Containers in multi-user environments: the udocker tool". En: *Computer Physics Communications* 232, págs. 84-97 (vid. págs. 58, 71, 74).

Gordon, Richard, Robert Bender y Gabor T Herman (1970). "Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography". En: *Journal of theoretical Biology* 29.3, págs. 471-481 (vid. pág. 17).

Grant, Eric J y col. (2017). "Solid cancer incidence among the life span study of atomic bomb survivors: 1958–2009". En: *Radiation research* 187.5, págs. 513-537 (vid. pág. 5).

Gregor, Jens y Thomas Benson (2008). "Computational analysis and improvement of SIRT". En: *IEEE transactions on medical imaging* 27.7, págs. 918-924 (vid. pág. 17).

Gullberg, G.T., Yu-Lung Hsieh y G.L. Zeng (1996). "An SVD reconstruction algorithm using a natural pixel representation of the attenuated Radon

transform". En: *IEEE Transactions on Nuclear Science* 43.1, págs. 295-303. DOI: 10.1109/23.485969 (vid. pág. 16).

Gunnels, John A. y col. (2001). "FLAME: Formal Linear Algebra Methods Environment". En: *ACM Trans. Math. Soft.* 27.4, págs. 422-455 (vid. pág. 63).

Gunter, Brian C. y Robert A. van de Geijn (2005). "Parallel Out-of-Core Computation and Updating the QR Factorization". En: *ACM Transactions on Mathematical Software* 31.1, págs. 60-78 (vid. pág. 165).

Gunter, Brian C., Wesley C. Reiley y Robert A. van de Geijn (2001). "Parallel Out-of-Core Cholesky and QR Factorizations with POOCLAPACK". En: *Proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE Computer Society (vid. págs. 155, 165).

Hall, EJ y DJ Brenner (2008). "Cancer risks from diagnostic radiology". En: *The British journal of radiology* 81.965, págs. 362-378 (vid. pág. 160).

Han, Yoseob y Jong Chul Ye (2018). "Framing U-Net via deep convolutional framelets: Application to sparse-view CT". En: *IEEE transactions on medical imaging* 37.6, págs. 1418-1429 (vid. pág. 18).

Hansen, Per Christian (2000). "The L-curve and its use in the numerical treatment of inverse problems". En: *InviteComputational Inverse Problems in Electrocardiology* (vid. pág. 128).

Hernandez, Vicente, Jose E Roman y Vicente Vidal (2005). "SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems". En: *ACM Transactions on Mathematical Software (TOMS)* 31.3, págs. 351-362 (vid. págs. 62, 109).

Hetmaniak, Y y col. (2003). "Pulmonary nodules: dosimetric and clinical studies at low dose multidetector CT". En: *Journal de radiologie* 84.4 Pt 1, págs. 399-404 (vid. pág. 8).

Hong, Jae-Young y col. (2019). "Association of exposure to diagnostic low-dose ionizing radiation with risk of cancer among youths in South Korea". En: *JAMA network open* 2.9, e1910584-e1910584 (vid. pág. 5).

Hore, Alain y Djemel Ziou (2010). "Image Quality Metrics: PSNR vs. SSIM". En: *2010 20th International Conference on Pattern Recognition*. IEEE, págs. 2366-2369. ISBN: 978-1-4244-7542-1. DOI: 10.1109/ICPR.2010.579 (vid. págs. 96, 115, 133, 146, 147, 163).

Hounsfield, G.N. (1973). "Computerized transverse axial scanning (tomography):Part I. Description of system". En: *British Journal of Radiology* 46, págs. 1016-1022 (vid. págs. 27, 70, 110).

Ichimaru, Michito, Toranosuke Ishimaru y Joseph L Belsky (1978). "Incidence of leukemia in atomic bomb survivors belonging to a fixed cohort in Hiroshima and Nagasaki, 1950-71: Radiation dose, years after exposure, age at exposure, and type of leukemia". En: *Journal of radiation research* 19.3, págs. 262-282 (vid. pág. 5).

Jain, Anil K (1989). *Fundamentals of digital image processing*. Prentice-Hall, Inc. (vid. pág. 93).

Jiang, Ming y Ge Wang (2003). "Convergence of the simultaneous algebraic reconstruction technique (SART)". En: *IEEE Transactions on image processing* 12.8, págs. 957-961 (vid. pág. 17).

Joffrain, Thierry, Enrique S. Quintana-Ortí y Robert A. van de Geijn (2005). "Rapid Development of High-Performance Out-of-Core Solvers". En: *Proceedings of PARA 2004*. LNCS 3732. Springer-Verlag Berlin Heidelberg, págs. 413-422 (vid. págs. 155, 165).

Joseph, P. (1982). "An improved algorithm for reprojecting rays through pixel images". En: *IEEE Transactions on Medical Imaging* 1.3, págs. 192-196 (vid. págs. 37, 72, 91, 110, 127, 143, 147, 162).

Kak, Avinash C. y Malcolm Slaney (2001). *Principles of Computerized Tomographic Imaging*. Society for Industrial y Applied Mathematics. ISBN: 978-0-89871-494-4. DOI: 10.1137/1.9780898719277 (vid. págs. 108, 140, 160, 174).

Katsikis, Vasilios N, Dimitrios Pappas y Athanassios Petralias (2011). "An improved method for the computation of the Moore-Penrose inverse matrix". En: *Applied Mathematics and Computation* 217.23, págs. 9828-9834 (vid. págs. 110, 113).

Khan, Atif N y col. (2013). "Effect of tube voltage (100 vs. 120 kVp) on radiation dose and image quality using prospective gating 320 row multidetector computed tomography angiography". En: *Journal of clinical imaging science* 3 (vid. pág. 9).

Kharfi, Faycal (2013). "Mathematics and Physics of Computed Tomography (CT): Demonstrations and Practical Examples". En: *Imaging and Radioanalytical Techniques in Interdisciplinary Research*. Ed. por Faycal Kharfi. Rijeka: IntechOpen. Cap. 4 (vid. pág. 174).

Kopp, Felix K. y col. (2018). "Diagnostic value of sparse sampling computed tomography for radiation dose reduction: initial results". En: *Medical Imaging 2018: Physics of Medical Imaging*. Ed. por Joseph Y. Lo, Taly Gilat Schmidt y Guang-Hong Chen. Vol. 10573. International Society for Optics y Photonics. SPIE, págs. 1027 -1032 (vid. pág. 160).

Korn, A y col. (2012). "Iterative reconstruction in head CT: image quality of routine and low-dose protocols in comparison with standard filtered backprojection". En: *American journal of neuroradiology* 33.2, págs. 218-224 (vid. pág. 15).

Koshev, Nikolay, Elias S Helou y Eduardo X Miqueles (2016). "Fast Backprojection Techniques for High Resolution Tomography". En: *arXiv preprint arXiv:1608.03589* (vid. pág. 182).

Krille, L y col. (2015). "Risk of cancer incidence before the age of 15 years after exposure to ionising radiation from computed tomography: results from a German cohort study". En: *Radiation and environmental biophysics* 54.1, págs. 1-12 (vid. pág. 5).

Kubo, Takeshi y col. (2016). "Low dose chest CT protocol (50 mAs) as a routine protocol for comprehensive assessment of intrathoracic abnormality". En: *European journal of radiology open* 3, págs. 86-94 (vid. págs. 7, 8).

Kumar, Indrajeet y col. (2014). "Reduction of speckle noise from medical images using principal component analysis image fusion". En: *2014 9th International Conference on Industrial and Information Systems (ICIIS)*. IEEE, págs. 1-6 (vid. pág. 90).

Lawson, C. L. y col. (1979). "Basic Linear Algebra Subprograms for Fortran Usage". En: *ACM Trans. Math. Soft.* 5.3, págs. 308-323 (vid. pág. 59).

Lee, D y col. (2017). "Quantitative evaluation of anatomical noise in chest digital tomosynthesis, digital radiography, and computed tomography". En: *Journal of Instrumentation* 12.04, T04006 (vid. pág. 8).

Leswick, David A y col. (2008). "Thyroid shields versus z-axis automatic tube current modulation for dose reduction at neck CT". En: *Radiology* 249.2, págs. 572-580 (vid. pág. 8).

Li, Yinsheng y col. (2019). "Learning to reconstruct computed tomography images directly from sinogram data under a variety of data acquisition conditions". En: *IEEE transactions on medical imaging* 38.10, págs. 2469-2481 (vid. pág. 18).

Lim, Jae S (1990). "Two-dimensional signal and image processing". En: *Englewood Cliffs* (vid. págs. 93, 94).

Maier, Andreas y col. (2013). "CONRAD—A software framework for cone-beam imaging in radiology". En: *Medical physics* 40.11 (vid. págs. 72, 110).

Maldjian, Pierre D y Alice R Goldman (2013). "Reducing radiation dose in body CT: a primer on dose metrics and key CT technical parameters". En: *American journal of roentgenology* 200.4, págs. 741-747 (vid. pág. 6).

Marqués, M. y col. (2011). "Using desktop computers to solve large-scale dense linear algebra problems". En: *The Journal of Supercomputing* 58.2, págs. 145-150. ISSN: 1573-0484 (vid. págs. 155, 166, 177).

Marqués, Mercedes y col. (2009). "Out-of-Core Computation of the QR Factorization on Multi-core Processors". En: *Lecture Notes in Computer Science 5704, Euro-Par'2009, (Eds. H. Sips, D. Epema, H. -X. Lin)*, págs. 809-820. ISBN: 978-3-642-03868-6 (vid. págs. 156, 166, 177).

Marquez, J. y col. (2020). "Image Quality Evaluation in Phase Contrast Computed Tomography from Synchrotron-Beams Reconstructed by Iterative

Methods". En: *Revista Cubana de Física* 37.2, págs. 88-94. ISSN: 2224-7939 (vid. pág. 17).

May, Matthias S y col. (2011). "Dose reduction in abdominal computed tomography: intraindividual comparison of image quality of full-dose standard and half-dose iterative reconstructions with dual-source computed tomography". En: *Investigative radiology* 46.7, págs. 465-470 (vid. pág. 15).

Merelli, Ivan y col. (2011). "Grid computing for sensitivity analysis of stochastic biological models". En: *International Conference on Parallel Computing Technologies*. Springer, págs. 62-73 (vid. pág. 71).

Merkel, Dirk (2014). "Docker: lightweight linux containers for consistent development and deployment". En: *Linux journal* 2014.239, pág. 2 (vid. págs. 57, 74).

Mettler, FA y col. (2019). *NCRP report no. 184: Medical radiation exposure of patients in the United States (184)* (vid. pág. 4).

Meulepas, Johanna M y col. (2019). "Radiation exposure from pediatric CT scans and subsequent cancer risk in the Netherlands". En: *JNCI: Journal of the National Cancer Institute* 111.3, págs. 256-263 (vid. pág. 5).

Miqueles, Eduardo, Nikolay Koshev y Elias S Helou (2017). "A backprojection slice theorem for tomographic reconstruction". En: *IEEE Transactions on Image Processing* 27.2, págs. 894-906 (vid. pág. 182).

Mori, Shinichiro y col. (2006). "A combination-weighted Feldkamp-based reconstruction algorithm for cone-beam CT". En: *Physics in Medicine & Biology* 51.16, pág. 3953 (vid. pág. 160).

Mosca, Ettore y col. (2009). "Stochastic simulations on a grid framework for parameter sweep applications in biological models". En: *2009 International Workshop on High Performance Computational Systems Biology*. IEEE, págs. 33-42 (vid. pág. 71).

Moscariello, Antonio y col. (2011). "Coronary CT angiography: image quality, diagnostic accuracy, and potential for radiation dose reduction using a novel iterative image reconstruction technique—comparison with traditional

filtered back projection". En: *European radiology* 21.10, págs. 2130-2138 (vid. pág. 15).

Muckley, Matthew J y col. (2019). "Image reconstruction for interrupted-beam x-ray CT on diagnostic clinical scanners". En: *Physics in Medicine & Biology* 64.15, pág. 155007 (vid. pág. 11).

Notohamiprodjo, S y col. (2015). "Image quality of iterative reconstruction in cranial CT imaging: comparison of model-based iterative reconstruction (MBIR) and adaptive statistical iterative reconstruction (ASiR)". En: *European radiology* 25.1, págs. 140-146 (vid. pág. 16).

Nyquist, Harry (1928). "Certain topics in telegraph transmission theory". En: *Transactions of the American Institute of Electrical Engineers* 47.2, págs. 617-644 (vid. pág. 111).

Padole, Atul y col. (2015). "CT radiation dose and iterative reconstruction techniques". En: *American Journal of Roentgenology* 204.4, W384-W392 (vid. pág. 125).

Paige, C. C. y M. A. Saunders (1982). "LSQR: An algorithm for sparse linear equations and sparse least squares". En: *ACM Transactions on Mathematical Software* 8.1, págs. 43-71 (vid. págs. 39, 40, 71, 92, 109, 144).

Papadakis, Antonios E y John Damilakis (2019). "Automatic tube current modulation and tube voltage selection in pediatric computed tomography: a phantom study on radiation dose and image quality". En: *Investigative radiology* 54.5, pág. 265 (vid. pág. 8).

Parcero, E y col. (2017). "Impact of view reduction in CT on radiation dose for patients". En: *Radiation Physics and Chemistry* 137, págs. 173-175 (vid. págs. 17, 39, 89, 91, 99, 111, 125, 141, 144, 153).

Pearce, Mark S y col. (2012). "Radiation exposure from CT scans in childhood and subsequent risk of leukaemia and brain tumours: a retrospective cohort study". En: *The Lancet* 380.9840, págs. 499-505 (vid. pág. 5).

Qi, Hongliang, Zijia Chen y Linghong Zhou (2015). "CT Image Reconstruction from Sparse Projections Using Adaptive TpV Regularization." En:

Computational and mathematical methods in medicine 2015, pág. 354869. ISSN: 1748-6718 (vid. pág. 161).

Quintana, Enrique S. y col. (enviado). "Gauss-Jordan Based Matrix Inversion and its Parallelization". En: *SJSC* (vid. pág. 128).

Quintana-Ortí, Gregorio y col. (2012). "A runtime system for programming out-of-core matrix algorithms-by-tiles on multithreaded architectures". En: *ACM Transactions on Mathematical Software (TOMS)* 38.4, pág. 25 (vid. págs. 166, 177).

Qurashi, A y col. (2018). "Optimal abdominal CT protocol for obese patients". En: *Radiography* 24.1, e1-e12 (vid. pág. 7).

Radon, J. (1986). "On the determination of functions from their integral values along certain manifolds". En: *IEEE Transactions on Medical Imaging* 5.4, págs. 170-176 (vid. págs. 70, 108).

Rahib, Lola y col. (2014). "Projecting Cancer Incidence and Deaths to 2030: The Unexpected Burden of Thyroid, Liver, and Pancreas Cancers in the United States". En: *Cancer Research* 74.11, págs. 2913-2921. ISSN: 0008-5472. DOI: 10.1158/0008-5472.CAN-14-0155. eprint: http://cancerres.aacrjournals.org/content/74/11/2913.full.pdf (vid. págs. 4, 141).

Ramírez Giraldo, Juan Carlos, Carolina Arboleda Clavijo y Cynthia H. McCollough (dic. de 2008). "TOMOGRAFÍA COMPUTARIZADA POR RAYOS X: FUNDAMENTOS Y ACTUALIDAD". es. En: *Revista Ingeniería Biomédica* 2, págs. 54 -66. ISSN: 1909-9762 (vid. pág. 36).

Rehani, Madan M y col. (2020). "Patients undergoing recurrent CT scans: assessing the magnitude". En: *European radiology* 30.4, págs. 1828-1836 (vid. pág. 5).

Rehani, MM y col. (2004). "Managing patient dose in computed tomography (CT)". En: *ICRP publication* 87 (vid. pág. 89).

Reiley, Wesley C. y Robert A. van de Geijn (1999). *POOCLAPACK: Parallel Out-of-Core Linear Algebra Package*. Inf. téc. CS-TR-99-33. Department of Computer Sciences, The University of Texas at Austin (vid. pág. 165).

Rodríguez-Alvarez, María José y col. (2018). "QR-factorization algorithm for computed tomography (CT): comparison with FDK and conjugate gradient (CG) algorithms". En: *IEEE Transactions on Radiation and Plasma Medical Sciences* 2.5, págs. 459-469 (vid. págs. 16, 125, 142, 161).

Roman, Jose E y col. (2015). "SLEPc users manual". En: *D. Sistemes Informàtics i Computació Universitat Politècnica de València, Valencia, Spain, Report No. DSIC-II/24/02* (vid. pág. 62).

Röntgen, W. C. (1896). "On a New Kind of Rays". En: *Science* 3.59, págs. 227-231. ISSN: 00368075, 10959203 (vid. pág. 23).

Sagara, Yoshiko y col. (2010). "Abdominal CT: comparison of low-dose CT with adaptive statistical iterative reconstruction and routine-dose CT with filtered back projection in 53 patients". En: *American Journal of Roentgenology* 195.3, págs. 713-719 (vid. pág. 15).

Sauter, Andreas P y col. (2019). "Sparse sampling computed tomography (SpSCT) for detection of pulmonary embolism: a feasibility study". En: *European radiology* 29.11, págs. 5950-5960 (vid. pág. 11).

Schultz, Carl H. y col. (2020). "The Risk of Cancer from CT Scans and Other Sources of Low-Dose Radiation: A Critical Appraisal of Methodologic Quality". En: *Prehospital and Disaster Medicine* 35.1, 3–16. DOI: 10.1017/S1049023X1900520X (vid. pág. 5).

Sechopoulos, Ioannis (2013). "A review of breast tomosynthesis. Part II. Image reconstruction, processing and analysis, and advanced applications". En: *Medical physics* 40.1 (vid. pág. 161).

Senthilraja, S, P Suresh y M Suganthi (2014). "Noise reduction in computed tomography image using WB filter". En: *International Journal of Scientific and Engineering Research* 5.3, págs. 243-247 (vid. págs. 90, 92).

Sollmann, N y col. (2018). "Effects of virtual tube current reduction and sparse sampling on MDCT-based femoral BMD measurements". En: *Osteoporosis International* 29.12, págs. 2685-2692 (vid. pág. 160).

Sollmann, Nico y col. (2019). "Multi-detector CT imaging: impact of virtual tube current reduction and sparse sampling on detection of vertebral fractures". En: *European radiology* 29.7, págs. 3606-3616 (vid. pág. 11).

Sørensen, Hans Henrik B y Per Christian Hansen (2014). "Multicore performance of block algebraic iterative reconstruction methods". En: *SIAM Journal on Scientific Computing* 36.5, págs. C524-C546 (vid. pág. 19).

Sung, Hyuna y col. (2021). "Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries". En: *CA: a cancer journal for clinicians* 71.3, págs. 209-249 (vid. pág. 4).

Szucs-Farkas, Zsolt y col. (2008). "Effect of X-ray tube parameters, iodine concentration, and patient size on image quality in pulmonary computed tomography angiography: a chest-phantom-study". En: *Investigative radiology* 43.6, págs. 374-381 (vid. pág. 9).

Tack, Denis y col. (2005). "Multi–detector row CT pulmonary angiography: comparison of standard-dose and simulated low-dose techniques". En: *Radiology* 236.1, págs. 318-325 (vid. pág. 8).

Tang, Jie, Brian E Nett y Guang-Hong Chen (2009). "Performance comparison between total variation (TV)-based compressed sensing and statistical iterative reconstruction algorithms". En: *Physics in Medicine and Biology* 54.19, págs. 5781-5804. ISSN: 0031-9155 (vid. págs. 160, 161).

Tang, Xiangyang y col. (2006). "A three-dimensional-weighted cone beam filtered backprojection (CB-FBP) algorithm for image reconstruction in volumetric CT helical scanning". En: *Physics in Medicine & Biology* 51.4, pág. 855 (vid. págs. 141, 160).

The MathWorks, Inc. (2021a). *MATLAB - MathWorks*. `http://es.mathworks.com/products/matlab/` (vid. pág. 57).

The MathWorks, Inc. (2021b). *MATLAB Compiler*. `https://es.mathworks.com/products/compiler/` (vid. pág. 74).

Toledo, Sivan y Fred Gustavson (ago. de 1998). "The Design and Implementation of SOLAR, a Portable Library for Scalable Out-of-Core Linear Al-

gebra Computations". En: *Proceedings of the Annual Workshop on I/O in Parallel and Distributed Systems, IOPADS.* (vid. pág. 165).

Tomasi, Carlo y Roberto Manduchi (1998). "Bilateral filtering for gray and color images". En: *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271).* IEEE, págs. 839-846 (vid. pág. 94).

Van Zee, Field G y Robert A Van De Geijn (2015). "BLIS: A framework for rapidly instantiating BLAS functionality". En: *ACM Transactions on Mathematical Software (TOMS)* 41.3, págs. 1-33 (vid. pág. 58).

Vandeghinste, Bert y col. (2013). "Low-Dose Micro-CT Imaging for Vascular Segmentation and Analysis Using Sparse-View Acquisitions". En: *PLoS ONE* 8.7. Ed. por Arrate Muñoz-Barrutia, e68449. ISSN: 1932-6203 (vid. págs. 160, 161).

Wang, Endong y col. (2014). "Intel math kernel library". En: *High-Performance Computing on the Intel® Xeon Phi™*. Springer, págs. 167-188 (vid. págs. 58, 62).

Wang, Ge y Ming Jiang (2004). "Ordered-subset simultaneous algebraic reconstruction techniques (OS-SART)". En: *Journal of X-ray Science and Technology* 12.3, págs. 169-177 (vid. pág. 17).

Whaley, R. Clint (2011). "ATLAS (Automatically Tuned Linear Algebra Software)". En: *Encyclopedia of Parallel Computing*. Ed. por David Padua. Boston, MA: Springer US, págs. 95-101. ISBN: 978-0-387-09766-4. DOI: 10.1007/978-0-387-09766-4_85 (vid. pág. 59).

Willemink, Martin J y col. (2013a). "Iterative reconstruction techniques for computed tomography Part 1: technical principles". En: *European radiology* 23.6, págs. 1623-1631 (vid. pág. 160).

Willemink, Martin J y col. (2013b). "Iterative reconstruction techniques for computed tomography part 2: initial results in dose reduction and image quality". En: *European radiology* 23.6, págs. 1632-1642 (vid. pág. 160).

Wolterink, Jelmer M y col. (2017). "Generative adversarial networks for noise reduction in low-dose CT". En: *IEEE transactions on medical imaging* 36.12, págs. 2536-2545 (vid. pág. 18).

Wu, Weiwen y col. (2018a). "Low-dose spectral CT reconstruction using image gradient $\ell 0$–norm and tensor dictionary". En: *Applied Mathematical Modelling* 63, págs. 538 -557. ISSN: 0307-904X (vid. pág. 160).

Wu, Weiwen y col. (2018b). "Non-local Low-rank Cube-based Tensor Factorization for Spectral CT Reconstruction". En: *IEEE transactions on medical imaging* 38.4, págs. 1079-1093 (vid. pág. 160).

Wu, Weiwen y col. (2019). "Improved Material Decomposition with a Two-step Regularization for spectral CT". En: *IEEE Access* 7, págs. 158770-158781 (vid. pág. 161).

Wysoczański, D, J Mroczka y AG Polak (2013). "Performance analysis of regularization algorithms used for image reconstruction in computed tomography". En: *Bulletin of the Polish Academy of Sciences. Technical Sciences* 61.2 (vid. pág. 18).

Xianyi, Zhang, Wang Qian y Werner Saar (2021). *OpenBLAS: An optimized BLAS library* (vid. pág. 58).

Xie, Shipeng y col. (2018). "Artifact removal using improved GoogLeNet for sparse-view CT reconstruction". En: *Scientific reports* 8.1, págs. 1-9 (vid. pág. 18).

Yan, Ke y col. (2018). "DeepLesion: automated mining of large-scale lesion annotations and universal lesion detection with deep learning". En: *Journal of Medical Imaging* 5.3, pág. 036501 (vid. págs. 92, 99, 127, 137, 147, 156, 172).

Yan Liu y col. (2014). "Total Variation-Stokes Strategy for Sparse-View X-ray CT Image Reconstruction". En: *IEEE Transactions on Medical Imaging* 33.3, págs. 749-763. ISSN: 0278-0062 (vid. pág. 160).

Yanagawa, Masahiro y col. (2012). "Pulmonary nodules: effect of adaptive statistical iterative reconstruction (ASIR) technique on performance of

a computer-aided detection (CAD) system—comparison of performance between different-dose CT scans". En: *European journal of radiology* 81.10, págs. 2877-2886 (vid. pág. 15).

Yang, Qingsong y col. (2018). "Low-dose CT image denoising using a generative adversarial network with Wasserstein distance and perceptual loss". En: *IEEE transactions on medical imaging* 37.6, págs. 1348-1357 (vid. pág. 18).

Yu, H. y Ge. Wang (2010). "A soft-threshold filtering approach for reconstruction from a limited number of projections". En: *Physics in Medicine and Biology* 55, pág. 3905 (vid. págs. 40, 71, 72, 92, 99, 144).

Yu, W. y L. Zeng (2014). "A Novel Weighted Total Difference Based Image Reconstruction Algorithm for Few-View Computed Tomography". En: *PLoS ONE* 9.10. DOI: e109345 (vid. págs. 18, 40, 70, 92, 99, 125, 144, 160).

Yu, Xiaodong y col. (2019). "GPU-based iterative medical CT image reconstructions". En: *Journal of Signal Processing Systems* 91.3, págs. 321-338 (vid. pág. 19).

Yurt, Ayşegül, İsmail Özsoykal y Funda Obuz (2019). "Effects of the use of automatic tube current modulation on patient dose and image quality in computed tomography". En: *Molecular imaging and radionuclide therapy* 28.3, pág. 96 (vid. pág. 9).

Zee, Field G. Van (2012). libflame*: The Complete Reference*. www.lulu.com (vid. pág. 63).

Zee, Field G. Van y col. (2009). "The libflame Library for Dense Matrix Computations". En: *IEEE Computation in Science & Engineering* 11.6, págs. 56-62 (vid. pág. 63).

Zhu, Zangen y col. (2013). "Improved compressed sensing-based algorithm for sparse-view CT image reconstruction." En: *Computational and mathematical methods in medicine* 2013, pág. 185750. ISSN: 1748-6718 (vid. pág. 160).

Zhuang, Tingliang y col. (2004). "Fan-beam and cone-beam image reconstruction via filtering the backprojection image of differentiated projection data". En: *Physics in Medicine & Biology* 49.24, pág. 5489 (vid. pág. 160).