

Document downloaded from:

<http://hdl.handle.net/10251/50444>

This paper must be cited as:

Vidal Gimeno, VE.; Flores, LA.; Mayo Nogueira, P.; Ródenas Escribá, FDA.; Verdú Martín, GJ. (2012). Algoritmo Paralelo de Reconstrucción de Imágenes TAC. 38º Reunión Anual Sociedad Nuclear Española. Grupo Senda.



The final publication is available at

<http://www.inscripcioneventos.com/577154/docs/577154-411839.doc>

Copyright Grupo Senda

Algoritmo Paralelo de Reconstrucción de Imágenes TAC*

Vicent Vidal, Liubov Flores, Patricia Mayo, Francisco Rodenas, Gumersindo Verdú
Universidad Politécnica de Valencia
Valencia, España

Resumen—En la tomografía axial computarizada (TAC) los rayos -X se usan para obtener las proyecciones necesarias para generar una imagen de la estructura interior de un objeto. La imagen puede ser reconstruida con diferentes técnicas. Los métodos algebraicos son los más apropiados para la reconstrucción de imágenes de alto contraste y precisión en las condiciones de ruido y por el menor número de proyecciones. Su uso puede ser importante en escáneres portátiles para su funcionalidad en situaciones de urgencia. Sin embargo, en la práctica, estos métodos no son ampliamente usados debido al alto coste computacional de su implementación. En este trabajo se analiza la reconstrucción paralela de imágenes con Portable Extensive Toolkit for Scientific computation (librería PETSc) y se propone el formato binario de datos que lleva a la reducción significativa de tiempo de reconstrucción.

Palabras claves: reconstrucción de imágenes TAC, algoritmos paralelos, librería PETSc, formato binario.

I. INTRODUCCIÓN

En la tomografía axial computarizada (TAC), un conjunto de proyecciones tomadas por un escáner se usa para reconstruir la estructura interna de un objeto. Una proyección k para un ángulo se define como una integral de intensidades de imagen $f(x, y)$ a lo largo de la línea de paso del rayo l y está dada por la formula:

$$P_k(\varphi) = \int_l f(x, y) dl. \quad (1)$$

*La Investigación fue financiada por el ANITRAN Project PROMETEO/2010/039.

L. A. Flores, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, Camino de Vera s/n, 46022, Valencia, España (e-mail: liuflo@posgrado.upv.es).

V. Vidal, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, Camino de Vera s/n, 46022, Valencia, España (e-mail: vvidal@upv.es).

P. Mayo, Titania Servicios Tecnológicos, Grupo Dominguis, Sorolla Center, local 10 Avda. de las Cortes Valencianas, 46015 Valencia, España (e-mail: p.mayo@titaniast.com).

F. Rodenas, Departamento de Matemática Aplicada, Universidad Politécnica de Valencia, Camino de Vera s/n, 46022, Valencia, España (e-mail: frodenas@mat.upv.es).

G. Verdú, Departamento de Ingeniería Química y Nuclear, Universidad Politécnica de Valencia, Camino de Vera s/n, 46022, Valencia, España (e-mail: gverdu@iqn.upv.es).

El problema de reconstrucción consiste en determinar los valores de la función $f(x, y)$ por el conjunto experimental de proyecciones.

Actualmente el proceso de reconstrucción en los escáneres clínicos se basa en algoritmos analíticos basados en la transformada inversa de Fourier. Uno de los algoritmos más estudiados es el algoritmo de retroproyección filtrada (FBP) [1]. Sin embargo, los algoritmos algebraicos representan una opción dominante por dos razones: primero, los métodos analíticos requieren un conjunto de proyecciones completas que no siempre son posibles. Segundo, estos métodos no proporcionan una reconstrucción óptima en condiciones de ruido en la imagen [2].

Los métodos algebraicos proporcionan reconstrucción de imágenes con alto contraste y precisión en condiciones de ruido y por el menor número de proyecciones [3]. En TAC es común encontrarse con proyecciones incompletas y no igualmente espaciadas. En estas situaciones los métodos algebraicos proporcionan imágenes de mejor calidad [4], [5], [6].

Sin embargo, como hemos mencionado, la mayor desventaja de los métodos algebraicos está dada por su alto coste computacional. Nosotros proponemos el uso de Extensive Toolkit for Scientific computation (PETSc) [8] para la reconstrucción paralela de imágenes. En nuestra previa investigación [9] hemos mostrado que PETSc facilita el trabajo de programación y posibilita el uso óptimo del sistema en el proceso de reconstrucción. En este trabajo, proponemos un formato diferente de presentación de datos de entrada, lo que permite reducir significativamente el tiempo de reconstrucción, uso de memoria y al mismo tiempo aumentar el tamaño de imágenes reconstruidas.

II. ASPECTOS MATEMÁTICOS

Supongamos que una imagen $f(x, y)$ puede ser aproximada por una matriz X , cuyos valores representan intensidades de la imagen. Asumimos que la imagen se encaja en una región cuadrada y está aproximada por la matriz del tamaño $n \times n$. Una proyección k para un ángulo φ puede ser escrita en forma discreta por la siguiente fórmula:

$$\sum_{i=1}^n \sum_{j=1}^n W_{ij}(k, \varphi) x_{ij} = P_{k, \varphi}, \quad (2)$$

donde los valores $W_{ij}(k, \varphi)$ representan la contribución de cada pixel de la imagen a cada proyección y dependen de la proyección k y el ángulo φ ; x_{ij} representan las intensidades desconocidas de la imagen; $P_{k,\varphi}$ corresponden a las proyecciones tomadas por el escáner. En forma matricial la ecuación (2) puede ser escrita como:

$$AX = P, \quad (3)$$

donde la matriz de coeficientes A está formada por los elementos $W_{ij}(k, \varphi)$, simula el proceso de recolección de proyecciones y puede ser no cuadrada.

Para el ángulo dado asumimos que el número de proyecciones varía de 1 a m . Entonces para k diferentes ángulos en (3) P es una matriz columna con mxk elementos, X es una matriz columna con n^2 elementos:

$$P = [p_{11} \dots p_{m1} \dots p_{mk}], \quad X = [x_{11} \dots x_{12} \dots x_{mn}] \quad (4)$$

A es una matriz rectangular del tamaño mxn^2 :

$$A = \begin{bmatrix} W_{11}(11) & W_{12}(11) & \dots & W_{mn}(11) \\ \dots & \dots & \dots & \dots \\ W_{11}(m1) & W_{12}(m1) & \dots & W_{mn}(m1) \\ \dots & \dots & \dots & \dots \\ W_{11}(mk) & W_{12}(mk) & \dots & W_{mn}(mk) \end{bmatrix}. \quad (5)$$

Muchas propiedades de la imagen reconstruida dependen de las aproximaciones cuando se calcula la matriz del sistema A . En la práctica, la matriz A es una matriz dispersa no cuadrada. Hemos usado el algoritmo de Siddon [9] para calcular los elementos de la matriz. Como se muestra en [10], este algoritmo da buenos resultados en la aproximación de la matriz. Las características principales de las matrices usadas en el experimento se presentan en la Tabla 1. El sistema (3) puede ser subdeterminado o sobredeterminado. Los sistemas sobredeterminados contienen más información sobre la imagen, y consecutivamente la imagen reconstruida tiene menos artefactos.

TABLA 1. LAS CARACTERÍSTICAS PRINCIPALES DE MATRIZ DE SISTEMA

Tamaño de Matriz (píxeles)	Tiempo de Generación (sec)	Tamaño de Matriz (MB)	
		ASCII format	Binary format
102400x65536	45.3	954	361
51200x262144	72.5	973	361
102400x262144	203.2	2047	721
204800x262144	446.4	4298	1500

Las dimensiones de la matriz A crecen proporcionalmente a la resolución de la imagen a reconstruir y al número de

proyecciones, aumentando de esta forma el coste computacional.

En (3) la matriz A y el vector P se generan previamente y pueden ser almacenadas en dos formatos: en formato de texto plano (ASCII) o en formato binario. A diferencia de nuestro previo trabajo [9], en este experimento los datos de entrada se presentan en forma binaria. Esto nos permitió aumentar la resolución de imágenes reconstruidas (hasta 512x512 píxeles) y al mismo tiempo reducir significativamente el tiempo de reconstrucción.

En la Figura 1 se ilustra los siguientes pasos principales del proceso de reconstrucción:

- Proyecciones TAC se colectan por un escáner
- La matriz del sistema que simula el proceso de escaneo se genera previamente por el algoritmo de Siddon
- En formato binario estos datos se usan por PETSc para encontrar la solución del sistema (3). La solución representa la imagen reconstruida.

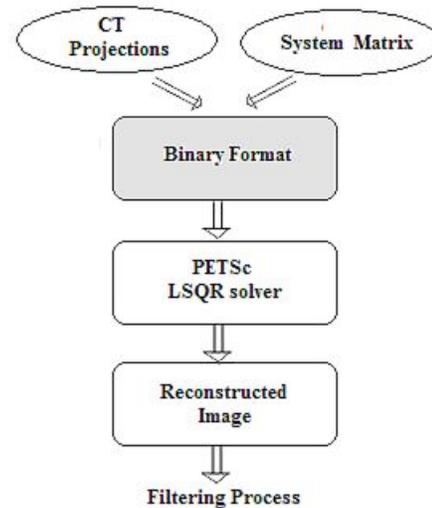


Figura 1. PETSc LSQR solver usa datos de entrada en formato binario para reconstruir la imagen

La tecnología de hoy posibilita la paralelización de cómputo asignando cada parte independiente del cálculo a un proceso, lo que proporciona un manejo más eficiente de los recursos de un sistema.

En este trabajo nosotros usamos la librería PETSc para leído paralelo de datos de entrada, ensamblaje de la matriz A y resolución del sistema (3) con el método iterativo.

PETSc (Portable Extensive Toolkit para cálculos científicos) consta de un conjunto de herramientas para la solución paralela numérica (así como secuencial) de sistemas de ecuaciones dispersas de gran tamaño. La librería incluye métodos iterativos basados en subespacios de Krylov y también varios formatos compactos de almacenamiento de matrices dispersas.

PETSc está diseñada para facilitar la extensión. De esta forma, al usar el paquete, los usuarios pueden incorporar métodos de solución y estructuras de datos particulares. Además, PETSc permite el uso de varios paquetes externos y puede ser usada en la mayoría de sistemas UNIX.

También, PETSc proporciona varias opciones de control a la hora de ejecución (*runtime control*) sin algún coste adicional. Las opciones incluyen control sobre el método de solución, preconditionadores, parámetros del problema, así como logos de ejecución.

PETSc está diseñada para ser usada en aplicaciones de gran tamaño y es común en la comunidad computacional de altas prestaciones.

III. METODOLOGÍA

En nuestro experimento hemos usado proyecciones reales y imágenes originales adquiridos en el Hospital Clínico Universitario en Valencia. Hemos trabajado con fan-beam proyecciones colectadas por el escáner con 512 sensores en el rango 0 - 180 con paso angular de 0.9 grados. Para poder reconstruir la imagen con el método iterativo hemos completado el conjunto dado de proyecciones hasta 360 grados usando la simetría de la matriz de coeficientes.

Con el propósito de analizar la capacidad de algoritmos iterativos en la reconstrucción paralela por el menor número de proyecciones, del conjunto inicial dado hemos generado tres conjuntos de proyecciones espaciados igualmente (con los pasos angulares de 0.9, 1.8 y 3.6 grados).

Como era mencionado previamente, PETSc facilita bastante el trabajo de programación. Hemos usado *LSQR* solver de la librería PETSc para encontrar la solución del problema de mínimos cuadrados. La mayor parte de tiempo se gasta para el ensamblaje de la matriz del sistema (3). Para lograr mejores resultados de ejecución, la matriz se almacena en *Compact Sparse Row format (CSR)* y el ensamblaje se realiza en paralelo. La matriz se divide entre los procesadores, cada uno de ellos es responsable por el ensamblaje y almacenamiento de una parte de la matriz que se usa posteriormente para resolver el sistema (3).

En la Figura 2 se muestra la idea de la ejecución del algoritmo paralelo: los datos de entrada se leen por un procesador; el ensamblaje de la matriz A y del vector P del sistema (3), así como la resolución del sistema, llevan a cabo en paralelo; la solución final, de nuevo, se guarda en el procesador principal. Los pasos principales del algoritmo se resumen como sigue:

- Leer datos de entrada
- Crear LSQR solver
- Resolver sistema

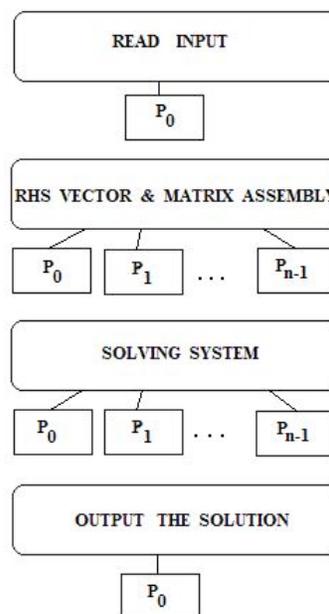


Figura 2. El esquema del algoritmo: para lograr la mejor performance el ensamblaje y resolución del sistema se hacen en paralelo

IV. RESULTADOS Y DISCUSIONES

El experimento fue llevado a cabo en sistema cluster Euler que pertenece a la Universidad Alicante en España. El cluster se compone de 26 nodos de computo. Cada nodo tiene dos procesadores Intel XEON X5660 hexacore @2.8GHz and 48 GB RAM. En Euler la función Grid Engine representa una herramienta de distribución de recursos de proposito general (Distributed Resource Management - DRM). Grid Engine facilita la ejecución de trabajos mediante colas y requerimientos definidos para cada trabajo.

Para la imagen de 512x512 píxeles el tiempo de ensamblaje (ta) de la matriz y el tiempo de resolución del sistema (ts) en segundos para diferente número de proyecciones y procesadores se resumen en la Tabla 2.

TABLA 2. EL TIEMPO DE ENSAMBLAJE Y RESOLUCIÓN DEL SISTEMA (EN SEGUNDOS) PARA DIFERENTE NÚMERO DE ROYECCIONES Y PROCESADORES EN CLUSTER EULER

Matriz de Sistema	Número de procesadores							
	$np = 2$		$np = 4$		$np = 8$		$np = 16$	
	ta	ts	ta	ts	ta	ts	ta	ts
51200x262144	5.6	0.6	3.1	0.5	2.1	0.3	1.9	0.8
10240x262144	11	1.4	6.2	0.9	4.3	0.8	3.7	1.2
204800x262144	22	4.3	12	2.4	8.7	2.3	7.6	2.5

En la matriz del sistema, el número de filas se obtiene multiplicando el número de sensores y ángulos usados y corresponde al número de proyecciones usadas para reconstruir la imagen; el número de columnas corresponde a la resolución de imagen reconstruida (512x512 píxeles).

Se puede notar que el tiempo de resolución (t_s) no varía mucho para diferente número de procesadores usados. La mayor parte de tiempo se gasta en ensamblaje (t_a) de la matriz del sistema y depende notablemente del número de procesadores. De esta forma, se nota el elemento de escalabilidad del algoritmo y la importancia de la paralelización del proceso.

La eficiencia de la reconstrucción paralela se aprecia en la Figura 3. Se logra el speed up de 1.8 para reconstruir la imagen de 512x512 píxeles.

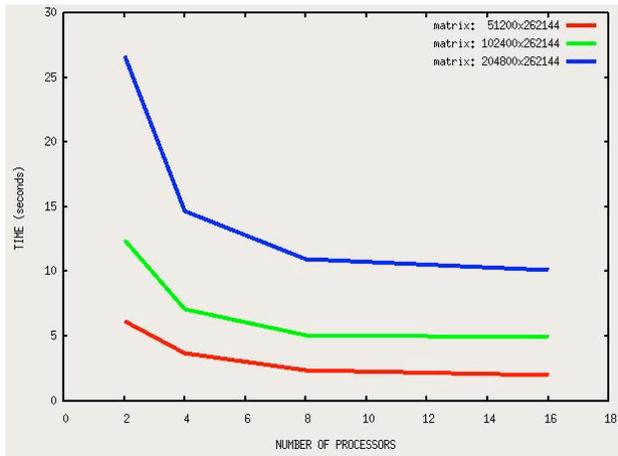


Figura 3. Tiempo de reconstrucción (en segundos) de la imagen de 512x512 píxeles por diferente número de proyecciones y procesadores; las dimensiones de la matriz corresponden: filas – al número de proyecciones, columnas – al tamaño (512x512 píxeles) de la imagen reconstruida

El rendimiento de PETSc en el proceso paralelo de reconstrucción de una imagen de 512x512 píxeles con la matriz de sistema del tamaño [204800x262144] se presenta en la Tabla 3.

TABLA 3. RENDIMIENTO DE PETSC EN LA RECONSTRUCCIÓN PARALELA DE IMÁGENES EN CADA NODO DE CLUSTER EULER

	np = 2	np = 4	np = 8	np = 16
Flops	2.90e+09	1.46e+08	7.30e+08	3.64e+08
Flops/sec	1.18e+08	9.20e+07	6.10e+07	3.16e+07
Uso de Memoria (Kbytes)	759727.9	380391.5	190271.4	89649.4
Mensajes Mpi	2.70e+01	8.20e+01	1.80e+02	3.80e+02

En la Tabla 3, Flops y Flops/sec corresponden al número de operaciones total y por segundo realizadas en cada nodo de cómputo, np corresponde al número de procesadores usados.

Se puede notar que la reconstrucción paralela reduce el uso de memoria y el número de operaciones en cada nodo optimizando de esta forma todo el proceso.

Finalmente, en la Figura 4 se muestran imágenes reconstruidas en forma paralela por diferente número de proyecciones igualmente espaciadas y en la Tabla 4 se resumen los resultados de la comparación cuantitativa entre la imagen original y reconstruida.

Para la comparación de imágenes se usaron funciones Mean Square Error (MSE) y Peak Signal to Noise Ratio (PSNR) [9]. Los resultados muestran que el algoritmo iterativo es capaz de reconstruir imágenes de buena calidad por menor número de proyecciones.

TABLA 4. COMPARACIÓN CUANTITATIVA ENTRE LA IMAGEN ORIGINAL Y RECONSTRUIDA

N proyecciones	MSE	PSNR
512x100	0.0132	66.9394
512x200	0.0108	67.8119
512x400	0.0095	68.3459

Generalmente, para mejorar la calidad de una imagen reconstruida se realiza postprocesado (e.g. filtrado) de la imagen. En este trabajo se compara las imágenes después de la reconstrucción sin ningún filtrado.

En la Tabla 4 el número de proyecciones se calcula multiplicando en número de sensores y ángulos usados para la colección de las proyecciones. Los resultados muestran que algoritmos iterativos son capaces de reconstruir imágenes de buena calidad por menor número de proyecciones.

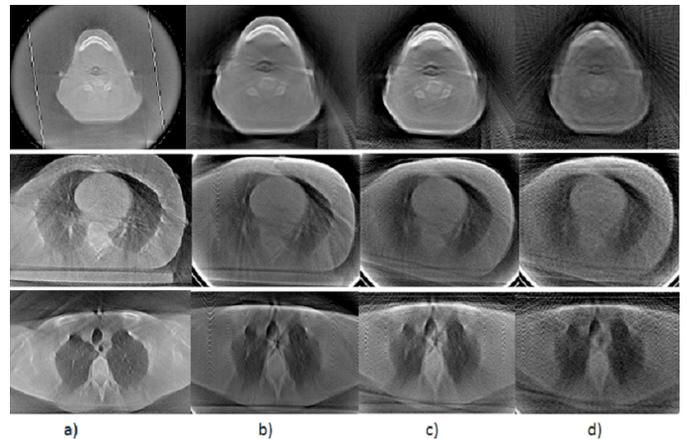


Figura 4. Imágenes reconstruidas: a) imágenes originales; b), c), d) reconstrucción iterativa por 400, 200 y 100 ángulos en la iteración 12 cuando se logra la tolerancia indicada

V. CONCLUSIONES

En este trabajo se analizó la posibilidad de reconstrucción de imagen con el método iterativo LSQR por el menor número de proyecciones y con el empleo de la librería PETSc para la optimización de todo el proceso de reconstrucción. Los resultados obtenidos muestran:

- Que el formato binario de datos permite significativamente reducción el uso de memoria y tiempo de reconstrucción lo que posibilita aumento de resolución de imágenes reconstruidas.
- Los métodos algebraicos son capaces de reconstruir imágenes de buena calidad sin necesidad de un conjunto completo de proyecciones.
- Nosotros esperamos resultados más significativos en la reconstrucción en 3D con el gran número de cómputo.

AGRADECIMIENTOS

Queremos agradecer Sergio Díez, Jefe de Servicio de Protección Radiológica y Radio física de la hospital Clínico Universitario, por la colaboración de llevar a cabo este trabajo.

También queremos mencionar la Universidad Alicante por posibilitar la ejecución de nuestros algoritmos en el sistema cluster Euler.

REFERENCIAS

- [1] R. C. Gonzales, R. E. Woods. Digital image processing. 3rd edition, Prentice Hall, 2008
- [2] G. Wang, H. Yu, and B. De Man. An outlook on X-ray CT research and development. *Medical Physics*, 35(3):1051-1064, Mar. 2008.
- [3] R. S. Deans. *The Radon Transform and Some of Its Applications*. Dover Publications, INC. Mineola, New York, 2007.
- [4] B. M. Crawford and G. T. Herman. Low-dose, large-angled cone-beam helical CT data reconstruction using algebraic reconstruction techniques. *Image and Vision Comp.*, 25:78-94, 2007.
- [5] J. Nuyts, B. De Man, P. Dupont, M. Defrise, P. Suetens, and L. Mortelmans. Iterative reconstruction for helical CT : A simulation study. *Phys. Med. Biol.*, 43:729-737, 1998.
- [6] R. G. Wells, M. A. King, P. H. Simkin, P. F. Judy, A. B. Brill, H. C. Giord, R. Licho, P. H. Pretorius, P. B. Schneider, and D. W. Seldin. Comparing Filtered backprojection and ordered-subsets expectation maximization for small-lesion detection and localization in 67Ga SPECT. *J. Nucl. Med.*, 41:1391-1399, 2000.
- [7] N. Sinha and J. T. W. Yeow. Carbon nanotubes for biomedical applications. *IEEE Trans. Nano.*, 4(2):180-196, 2005.
- [8] <http://acts.nersc.gov/petsc/index.html>
- [9] L. Flores, V. Vidal, P. Mayo, F. Rodenas, G. Verdú, “ Iterative reconstruction of CT images with PETSc”. *BMEI*, vol 1, 343-346, 2011