

Optimización en cuanto a la paralelización de la transformada de Radon Inversa

L.A. Flores^{1,6}, V. Vidal Gimeno², P. Mayo Nogueira³, F. Rodenas Escriba⁴, G. Verdú⁵.

^{1,2}*Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, Camino de Vera s/n, 46022, Valencia, España, liuflo@posgrado.upv.es, vvidal@upv.es*

³*TITANIA Servicios Tecnológicos, Grupo Dominguis, Sorolla Center, local 10 Avda. de las Cortes Valencianas, 46015 Valencia, España, p.mayo@titaniast.com*

⁴*Departamento de Matemática Aplicada, Universidad Politécnica de Valencia, Camino de Vera s/n, 46022, Valencia, España, frodenas@mat.upv.es*

⁵*Departamento de Ingeniería Química y Nuclear, Universidad Politécnica de Valencia, Camino de Vera s/n, 46022, Valencia, España, gverdu@iqn.upv.es*

⁶*Departamento de Informática, Física y Matemáticas, Universidad Peruana Cayetano Heredia, Honorio Delgado s/n, 31, Lima, Perú, liuflo@posgrado.upv.es*

Recibido el 11 de setiembre del 2011; aceptado el 16 octubre del 2011

En los escáners clínicos el proceso de reconstrucción de imagen se basa en algoritmos analíticos entre los cuales, el algoritmo de retroproyección filtrada “filtered backprojection” es el más conocido. Este algoritmo se usa para implementar la transformada de Radon inversa que es una herramienta matemática cuya utilización principal en Ingeniería Biomédica es la reconstrucción de imágenes de TAC (Tomografía Axial Computarizada). La transformada de Radon es un conjunto de proyecciones tomadas bajo diferentes ángulos que contienen información sobre el objeto (imagen) escaneado y la inversa de la transformada, representa la imagen de dicho objeto. Desde el comienzo del desarrollo de escanners ha sido importante reducir el tiempo de escaneo, mejorar la calidad de imagen y reducir el tiempo de reconstrucción. La tecnología de hoy ofrece potentes ordenadores con varios procesadores y núcleos que posibilitan reducir el tiempo invertido en la reconstrucción de imágenes. En este trabajo se analiza el algoritmo de reconstrucción de imagen mediante proyecciones paralelas, basado en la transformada de Radon inversa con el objetivo de optimizar su cálculo aprovechando al máximo los recursos de sistemas. Aunque los escanners TAC que están en uso hoy adquieren las proyecciones en geometría fan-beam, en muchas situaciones es preferible reducir dichas proyecciones al algoritmo de retroproyección filtrada basado en proyecciones paralelas que se destaca por su simplicidad y robustez. Esto permite extender los resultados a variedad de situaciones.

Palabras Claves: Tomografía Axial Computarizada, transformada de Radon, reconstrucción paralela.

In clinical scanners, the reconstruction process is based on analytical algorithms. Among them, filtered back projection algorithm (FBP) is the most famous. This algorithm implements the inverse Radon transform which is a mathematical tool used in Biomedical engineering for reconstruction of computer tomography images (TC). The Radon transform is a set of parallel projections taken under different angles and contains the information upon the scanned object (image). The inverse of the transform represents the internal structure of the original object. From the very beginning of the scanners appearance, it was important to reduce the time of scanning, improve the quality of the image and to reduce the time of the reconstruction. Today computers are equipped with multi processors and many core processors that allow reducing the time of reconstruction. In this work, we analyze the parallel implementation of the reconstruction algorithm based on the inverse Radon transform with the aim to optimize the usage of the system resources. Although in modern scanners fan-beam projections are used for the reconstruction, in many situations it is preferable to convert fan-beam projections to parallel and take advantage of the Radon transform for the reconstruction due to its simplicity and robustness. This allows extending results to variety of applications.

Keywords: CT imaging, Radon transform, parallel reconstruction.

1. Introducción

En muchas aplicaciones se desea determinar (reconstruir) las propiedades estructurales de un objeto utilizando datos experimentales del mismo [1].

El problema de reconstrucción por proyecciones fue resuelto por Johan Radon en 1917. Sucesivos investigadores en diversas áreas no conocían el trabajo de Radon [1] y por eso existe muchos redescubrimientos de los re-

sultados de Radon hasta los años 70 [1].

Los avances tecnológicos y teóricos han promovido un interés continuo de los diferentes métodos de reconstrucción y sus implementaciones. Una de las áreas más amplias de aplicación de la transformada de Radón es la Tomografía Axial Computarizada, donde la información sobre la imagen original se da en forma de conjunto de proyecciones, que se conoce como la transformada de Radon de la imagen, y la inversa de la transformada,

representa la misma imagen [1].

Existen varios métodos de reconstrucción de imagen. Estos métodos se pueden clasificar en: - algoritmos analíticos, - métodos algebraicos, - métodos estadísticos de reconstrucción. Actualmente en la Tomografía Axial Computarizada, la reconstrucción se basa en algoritmos analíticos entre los cuales el algoritmo de retroproyección filtrada es de los más conocidos.

En la implementación de estos algoritmos, se puede plantear como optimizar su ejecución para conseguir tiempos menores. Por ello es importante la computación en paralelo que distribuye procesos de cálculo de forma eficiente.

En este trabajo se analiza el algoritmo de implementación de la transformada de Radon inversa en la reconstrucción de imagen, representada por sus proyecciones paralelas, con el objetivo de determinar el grado de paralelización y escalabilidad del algoritmo para optimizarlo. Aunque en la mayoría de scanners de TAC que están en uso hoy no se adquieren las proyecciones en la geometría paralela, en muchas circunstancias es posible reducir a ellas [2].

2. Aspectos matemáticos de reconstrucción de imagen

Dado la utilidad de la transformada de Radon y su inversa en la reconstrucción de imagen TAC en este apartado vamos a resumir el algoritmo en el que se basa.

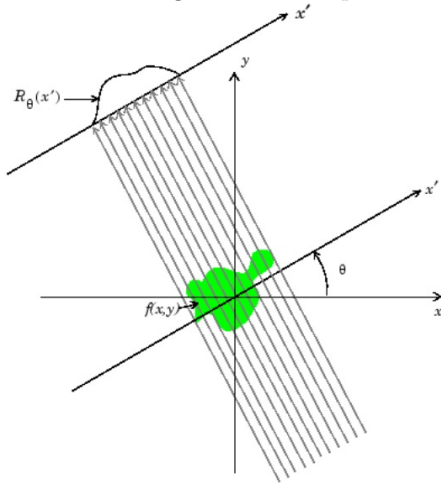


Figura 1. Proyección de un objeto.

2.1. Definición de la transformada de Radon y su inversa

La proyección $R(\theta, s)$ de un objeto inicial se define como el conjunto de integrales de línea de la intensidad de imagen, representada por $f(x, y)$, a lo largo de las líneas l a una distancia s del origen del sistema de coordenadas y formando un ángulo θ con el eje x :

$$R(\theta, s) = \int_l f(x, y) dl \quad (1)$$

Una proyección $R_\theta(x')$ para un ángulo θ determinado, puede verse en la Figura 1.

Un conjunto de proyecciones tomadas para diferentes ángulos, representa la imagen y se conoce como la transformada de Radon. La transformada de Radon en R^2 puede ser expresada en la forma:

$$\mathfrak{R}(f(\theta, s)) = \check{f} = \int_{-\infty}^{\infty} f(x(t), y(t)) dt \quad (2)$$

Donde las coordenadas se han expresado en forma paramétrica mediante la expresión:

$$(x(t), y(t)) = t(\sin \theta, -\cos \theta) + s(\cos \theta, \sin \theta) \quad (3)$$

La función $f(x, y)$ que como se ha comentado, representa la imagen original, puede ser calculada mediante la inversa de la transformada de Radon. En forma simbólica:

$$f(x, y) = \mathfrak{R}^{-1}(\check{f}) \quad (4)$$

Uno de los métodos de calcular la transformada de Radon inversa se conoce como el método directo de Fourier.

2.2. Relación con la Transformada de Fourier para el cálculo de la transformada de Radon inversa

La transformada de Radon está relacionada con la transformada de Fourier por medio del Teorema de Slices [1], que obtiene la transformada de Fourier 2D de la imagen por medio de las proyecciones.

De esta forma, se conoce la transformada de Fourier 2D de la imagen (al menos en algunos puntos) y tomando su inversa se puede obtener la imagen:

$$f(x, y) = \frac{1}{4\pi^2} \int \int G(\theta, \omega) e^{j\omega(x \sin \theta - y \cos \theta)} |\omega| d\omega d\theta \quad (5)$$

donde $G(\theta, \omega)$ es la transformada de Fourier 1D de las proyecciones, $|\omega|$ es el determinante del Jacobiano al pasar de sistema de coordenadas rectangulares a polares. El producto $G(\theta, \omega)|\omega|$ se llama retroproyección filtrada y se obtiene multiplicando las proyecciones por $|\omega|$ en el espacio de Fourier.

La fórmula (4) es la base del cálculo de la transformada de Radon inversa en la reconstrucción de TAC y representa la relación entre las proyecciones y la imagen inicial. El algoritmo de retroproyección filtrada es uno de los algoritmos más aceptados en la reconstrucción de la imagen original en Tomografía Axial Computarizada.

3. Paralelización

En un ordenador la unidad central donde se llevan a cabo los cálculos, es el procesador. Los ordenadores modernos vienen equipados con varios procesadores o, incluso, con varios núcleos (cores) de procesamiento de datos, incluidos en el mismo procesador.

Esta tecnología permite realizar el cálculo de forma secuencial, utilizando un solo proceso, o en forma paralela asignando una porción independiente del cálculo a cada proceso. Esto conlleva una mayor eficacia en la gestión de

recursos de un ordenador. Por otra parte, esta mayor eficacia es muy útil en la reconstrucción de imágenes, pues es un proceso que consume bastante tiempo. Aprovechando los sistemas modernos, se puede reducir el tiempo total de reconstrucción.

4. Resultados del trabajo realizado

El algoritmo representado por la formula (4) es paralelizable porque la reconstrucción se realiza mediante las proyecciones que son independientes. En este trabajo se trata de analizar el grado de paralelización para aprovechar al máximo los recursos de sistema.

El algoritmo fue testeado en dos sistemas diferentes:

- En QUADCLUSTER del departamento DSIC de la UPV. El QUADCLUSTER se compone de cuatro nodos modelo XE310, con procesadores Xeon Quadcore 5365 a 3 Ghz, esto es con cuatro cores cada uno. Los nodos están interconectados mediante una red Gigabit Ethernet.
- En el sistema con el procesador Intel® Core™ 2 Duo CPU P8400 con 2.27 GHz, con dos cores.

Se han adquirido imágenes digitales en formato DICOM, que es el formato que se está implantando en el campo de radiología médica, de tamaños 6.46MB, 13.3MB, 28.7MB. Para la reconstrucción de las imágenes se usaron las proyecciones paralelas sintéticas de estas imágenes generadas en Matlab en el rango de 180 grados.

Se han analizado el algoritmo secuencial y paralelo para cada uno de los sistemas descritos antes. Se han obtenido los tiempos de ejecución, variando el número de procesos Np relacionado con los cores y procesadores de cada sistema, y el tamaño del problema N que está relacionado con la dimensión de imagen.

Los resultados de ejecución se resumen en las Tablas 1 y 2 donde se presenta el tiempo (en segundos) de reconstrucción de imágenes de diferentes tamaños N con diferentes números de procesos Np en los sistemas estudiados.

Tabla 1. Tiempo de ejecución en un nodo de QUADCLUSTER

Np \ N	367	1452	2531	3795	5545
1	1.2	20.3	61.7	138.4	295.5
2	0.9	10.5	32.1	72.3	154.5
4	0.6	6.0	18.7	42.0	91.0
6	0.9	6.8	20.3	45.3	93.7
16	0.9	6.9	21.2	44.0	94.7

Tabla 2. Tiempo de ejecución en el sistema con Core 2 Duo.

Np \ N	367	1452	2531	3795	5545
1	3.8	60.2	177.8	403.1	857.0
2	2.2	35.9	108.1	241.9	519.1
4	2.3	34.8	107.5	240.1	521.4
6	2.2	35.2	105.1	242.1	518.2
16	2.0	34.2	105.5	234.9	506.1

Los gráficos siguientes permiten visualizar la dependencia del tiempo de ejecución en función del tamaño de imagen N y número de procesos Np.

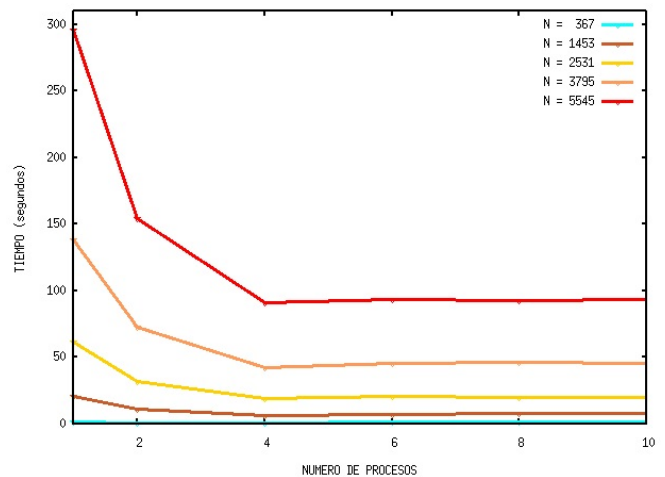


Figura 2. Variación de tiempo de ejecución en un nodo de QUADCLUSTER con Quadcore de 3Ghz.

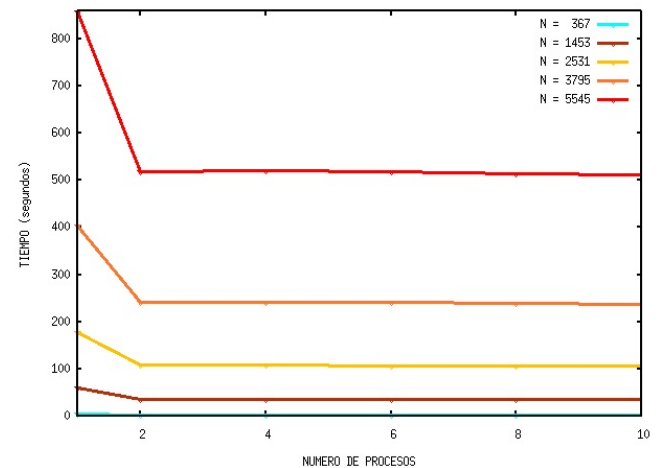


Figura 3. Variación de tiempo de ejecución en sistema con el procesador Core 2 Duo de 2.27Ghz.

Analizando los resultados obtenidos acerca del grado de paralelismo y eficiencia del algoritmo en estos sistemas, se observa que:

- En QUADCLUSTER, el tiempo óptimo se logra con cuatro procesos. El tiempo de reconstrucción de imagen con cuatro procesos disminuye en 3 veces comparado con el tiempo necesario para el mismo cálculo con un solo proceso lo que lleva a una eficiencia del 80% del sistema.
- En el sistema Core 2 Duo, las condiciones óptimas

se logran con dos procesos con una eficiencia del sistema de 83%. La eficiencia de los sistemas en ambos casos es aproximadamente de 80% y se obtiene con el número de procesos iguales al número de procesadores del sistema.

A continuación se presenta a modo de ejemplo una imagen sintética reconstruida a partir de 180 proyecciones mediante el algoritmo secuencial y el paralelo en uno de los 2 sistemas.

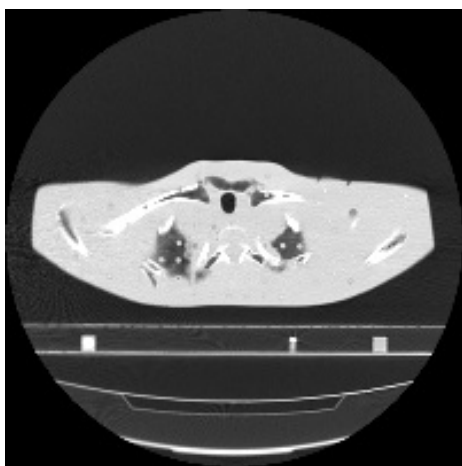


Figura 4. *Imagen original.*

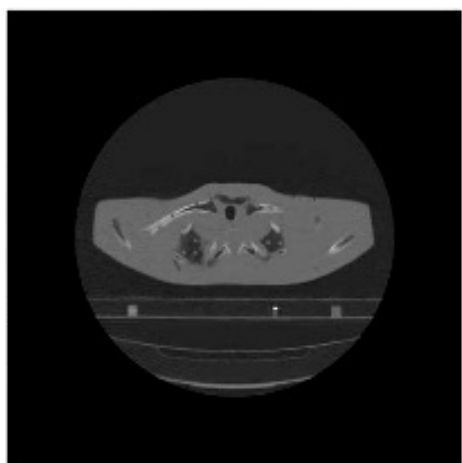


Figura 5. *Imagen reconstruida mediante el algoritmo secuencial.*

En la implementación de este algoritmo analítico, visualmente ambos métodos secuencial y paralelo obtienen una reconstrucción similar a la imagen original, teniendo la imagen obtenida por el secuencial menos artefactos.

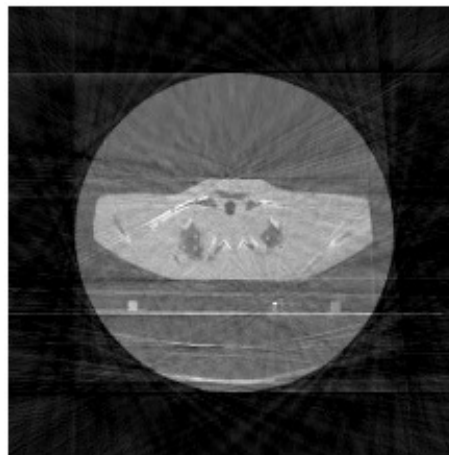


Figura 6. *Imagen reconstruida mediante el algoritmo paralelo.*

5. Conclusiones

Los resultados muestran que el algoritmo de retroproyección filtrada es paralelizable y el menor tiempo de ejecución se consigue cuando se ejecuta en forma paralela, utilizando el número de procesos igual al número de procesadores de sistema.

Por otro lado, la calidad de imagen es mejor cuando un solo proceso realiza la reconstrucción, por lo cual como trabajo futuro se va a realizar una comparación del algoritmo de retroproyección filtrada con otros algoritmos de reconstrucción. Con esto se va a optimizar su implementación aprovechando los recursos del sistema y es de esperar obtener mejores resultados en el proceso de reconstrucción de imágenes médicas.

Agredecimientos

Queremos agradecer a Sergio Díez, Jefe del Servicio de Protección Radiológica y Radiofísica del Hospital Clínico Universitario de Valencia, su colaboración para llevar a cabo este trabajo.

1. Stanley R Deans. The Radon Transform and Some of Its Applications. Dover Publications, INC, 2007.
2. Willi A. Kalender. Computed Tomography. Publicis Corporate Publishing, 2005.