34th Annual International Conference of the IEEE EMBS
San Diego, California USA, 28 August - 1 September, 2012

Fast Parallel Algorithm for CT Image Reconstruction*

Liubov A. Flores, Vicent Vidal, Patricia Mayo, Francisco Rodenas, Gumersindo Verdú

Abstract— **In X-ray computed tomography (CT) the X rays are used to obtain the projection data needed to generate an image of the inside of an object. The image can be generated with different techniques. Iterative methods are more suitable for the reconstruction of images with high contrast and precision in noisy conditions and from a small number of projections. Their use may be important in portable scanners for their functionality in emergency situations. However, in practice, these methods are not widely used due to the high computational cost of their implementation. In this work we analyze iterative parallel image reconstruction with the Portable Extensive Toolkit for Scientific computation (PETSc).**

I. Introduction

In medicine, the diagnosis based on computed tomography (CT) is fundamental for the detection of abnormal tissues by different attenuation on X-ray energy, which frequently is not clearly distinguished for radiologists. In CT imaging, a set of projections taken with a scanner is used to reconstruct the internal structure of an object. The intensity of a beam of X-ray that passes through some object is observed to decrease. By moving the source and detector, it is possible to obtain a set of projections. A single *k*-th projection at angle *r* can be defined as an integral of image intensities $f(x,y)$ along the line l and is given by the formula:

$$P_{k,r} = \int_l f(x,y)dl \qquad (1)$$

The reconstruction problem consists of determining the values of the function $f(x,y)$ from the set of the experimental projection data. The methods of reconstruction can be divided into analytical, algebraic and statistical methods. Although the first tomography reconstruction from projections had been made using algebraic methods, presently, the reconstruction process in clinical scanners is based on analytical algorithms which use the inverse Fourier transform. Filtered Back Projection algorithm (*FBP*) is one of the widely used algorithms and is well described in literature, for example [1]. On the other hand, the algebraic methods are less used due to their high computational cost.

Nevertheless, the algebraic methods represent a dominant option due to two reasons. Firstly, the analytical methods require complete data collection which is not always possible. Second, they do not provide the optimal reconstruction in noisy conditions in the image [2].

Algebraic methods allow reconstructing images with higher contrast and precision in noisy conditions from a small number of projections than the methods based on the Fourier transform [3]. In CT, it is common to find incomplete set of no equally spaced projections. In these cases, algebraic reconstruction methods provide images with better quality [4], [5], [6].

Nevertheless, the major drawback of the algebraic methods is given by their high computational cost. We propose the usage of Extensive Toolkit for Scientific computation (PETSc) [8] in parallel image reconstruction. As it has been shown in our previous work [9], PETSc facilitates a great deal of the programming task and provides the possibility for the optimal usage of a whole system in the process of reconstruction. In this work, we use different way of presenting input data, which allows us to reduce significantly the reconstruction time and memory usage, and, at the same time, to augment the sizes of the reconstructed images.

II. Mathematical Aspects

Let's suppose that an image $f(x,y)$ can be approximated by the matrix X, whose values represent intensities of the image. We assume that the image fits some square region and is approximated by *n*x*n* matrix *X*. A projection *k* taken at angle *r* can be written as follows:

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

where $W_{ij}(k,r)$ are weight factors that depend on the projection number *k* and the angle r. Since *i* and *j* both range from 1 to *n*, there are n^2 weight factors for a fixed *k* and r; x_{ij} are unknown intensities of the image. In matrix form (2) is given by the formula:

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

where the system matrix A is formed by the elements $W_{ij}(k,r)$, simulates computer tomography functioning and may not be square.

*Research supported by ANITRAN Project PROMETEO/2010/039.

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

V. Vidal is with the 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 is with the Servicios Tecnológicos, Grupo Dominguis, Sorolla Center, local 10 Avda. de las Cortes Valencianas, 46015 Valencia, España (e-mail: p.mayo@titaniast.com).

F. Rodenas is with the Departamento de Matemática Aplicada, Universidad Politécnica de Valencia, Camino de Vera s/n, i46022, Valencia, España (e-mail: frodenas@mat.upv.es).

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

For a given angle, we assume that the number of projections ranges from 1 to m. If there are k different angles, then in (3) P is the column matrix with mxk elements, X is the column matrix with n^2 elements:

$$P = \left[p_{11} \ldots p_{m1} \ldots p_{mk}\right]^T, \quad X = \left[x_{11} \ldots x_{12} \ldots x_{nn}\right]^T \qquad (4)$$

and A is the mkxn^2 rectangular matrix:

$$A = \begin{bmatrix} W_{11}(11) & W_{12}(11) & \ldots & W_{nn}(11) \\ \ldots & \ldots & \ldots & \ldots \\ W_{11}(m1) & W_{12}(m1) & \ldots & W_{nn}(m1) \\ \ldots & \ldots & \ldots & \ldots \\ W_{11}(mk) & W_{12}(mk) & \ldots & W_{nn}(mk) \end{bmatrix}. \qquad (5)$$

Many properties of the reconstructed image depend on the approximations when calculating the system matrix. In this work we use Siddon algorithm [9] to calculate elements of the matrix in a rectangular grid. It has been found [10] that Siddon algorithm gives a good approximation of the system matrix. In practice, A is a rectangular nonsymmetrical sparse matrix and therefore it is recommendable to store only nonzero elements. The main characteristics of the matrices used in the experiment are summarized in Table 1. The system (3) may be over determined or undetermined. Over determined systems contain more information on the image and, consequently, the reconstructed image is less noisy.

TABLE I. THE MAIN CHARACTERISTICS OF THE SYSTEM MATRIX

Matrix Size (pixels)	# Nonzero Elements	Generation Time (sec)	Matrix Size (MB)
102400x65536	31490052	51	361
51200x262144	31496952	56	361
102400x262144	62993644	180	722
204800x262144	125986544	540	1500

Fundamentally, the algebraic methods of image reconstruction from projections are schemes for solving a linear system (3). The dimensions of A grow proportionally to the resolution of the image to be reconstructed and the number of projections, increasing therefore the computational cost.

In (3) the input matrix (A) and the right hand side vector (P) are generated previously and can be stored in two formats: as a plain text or in a binary format. The results in this work show that the binary format allows reducing significantly the assembling time performed by PETSc after reading the input data.

Fig. 1 illustrates the following main steps of the reconstruction process:

- CT projections are recollected by a scanner
- The system matrix, that simulates the scanning process, is generated previously by Siddon algorithm
- In binary format these data are used by LSQR solver to find the solution of the system (3) that represents the reconstructed image.

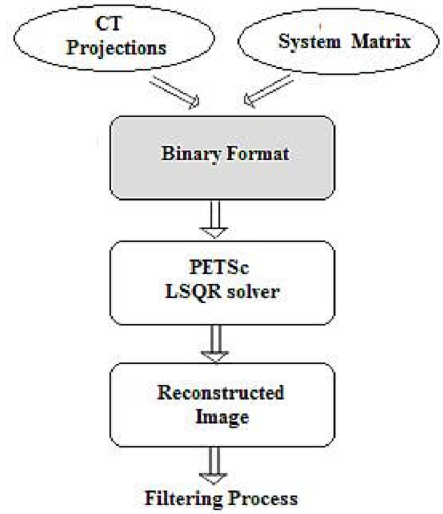


Figure 1. PETSc LSQR solver uses input data in binary format to reconstruct the image

Today's technology provides the possibility to parallelize calculations assigning each independent part to one processor which allows more efficient management of the resources of a system. In this work we employ PETSc library for parallel load of data, assembly of the system matrix and solving system (3) with an iterative method.

PETSc is a set of tools for parallel (as well as serial) numerical solving of large sparse systems of equations. The library includes iterative methods based on Krylov subspaces, as well as several parallel formats for sparse matrices. PETSc is designed to facilitate extensibility. Thus, users can incorporate customized solvers and data structures when using the package. PETSc also provides an interface to several external software packages and runs on most UNIX based-systems.

Furthermore, PETSc enables a great deal of runtime control for the user without any additional coding cost. The runtime options include control over the choice of solvers, preconditioners and problem parameters as well as the generation of performance logs. PETSc is intended for use in large-scale application projects and is in widespread general use throughout the high performance community.

III. METHODOLOGY

For experimental purposes we used the real projections and the original image acquired from the Hospital Clinico Universitario in Valencia. We worked with fan-beam projections collected by the scanner with 512 sensors in the range 0 - 180 with 0.9 degree spacing. To be able to reconstruct the image with the iterative method we complete the given set up to 360 degrees using the symmetry of the system matrix.

We wanted to analyze the capacity of iterative algorithms in parallel reconstruction of images from less number of projections. With this purpose, from the initial set, three sets of equally spaced (with the angle steps 0.9, 1.8, and 3.6 degrees) projections have been derived.

As it has been aforementioned, PETSc facilitates a great deal of programming task. We use *LSQR* solver from PETSc library to find a solution of the least square problem that is equivalent to the ill conditioned system (3). The most part of the computation time is spent to assemble the system matrix which is stored in the Compact Sparse Row format (CSR). To achieve better performance the assembly is made in parallel. The input matrix is divided among the processors, each of which is responsible of the assembly and the storage of one part of the matrix that is used afterwards for solving the system.

Fig. 2 shows the idea of the parallel execution of the algorithm: the input data is read by one processor; the assembly of the system matrix and the right hand side vector, as well as the system solving, are made in parallel; the final solution, again, is stored in the main processor.

Thus, the main steps of the algorithm can be resumed as follows:

- Load input data
- Create LSQR solver
- Solve the system

IV. Results And Discussions

The results have been measured on a cluster system Euler that belongs to the Alicante University in Spain. The cluster is composed of 26 computing nodes. Each of these nodes has two processors Intel XEON X5660 hexacore @2.8GHz and 48 GB RAM. In Euler, it is used Grid Engine function, general purpose Distributed Resource Management (DRM) tool. The scheduler component in Grid Engine supports a wide range of different compute scenarios. Grid Engine is a facility for executing Unix-like batch jobs (shell scripts or binaries) on a pool of cooperating CPUs. Jobs are queued and executed remotely according to defined policies.

For the image of 512x512 pixels the assembling time (***ta***) of the matrix and the system solving time (***ts***) (in seconds) for different number of projections and processors are summarized in Table 2.

In the system matrix, the number of rows is obtained by multiplying the number of used sensors and angles and corresponds to the number of the projections used to reconstruct the image; the number of columns corresponds to the size of the reconstructed image (512x512 pixels).

It can be seen that the system solving time does not vary very much for different number of processors used. The most time is spent to assemble the system matrix and depend noticeably on the number of processors. So, the results show the elements of scalability of the algorithm.

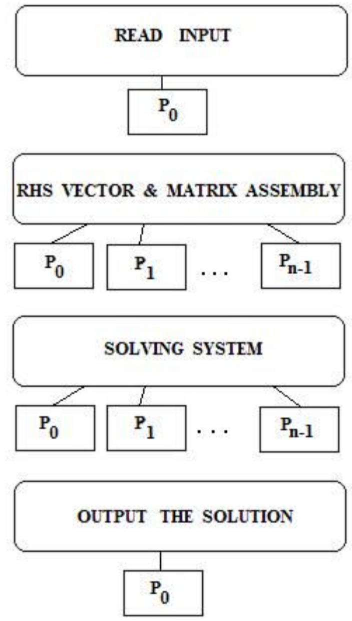


Figure 2. The scheme of the algorithm: to achieve better performance the assembly and the system solving are made in parallel

TABLE II. The Assembling And Execution Time (in seconds) For Different Number Of Projections And Processors On Euler Cluster

System Matrix	Number of processors							
	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
102400x262144	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

The efficiency of the parallel image reconstruction can be appreciated in Fig. 3. The speed up of 1.8 has been achieved to reconstruct the image of 512x512 pixels.

PETSc performance during the parallel reconstruction of an image (512x512 pixels) with the coefficient matrix of the size [204800x262144] is presented in Table 3. It can be seen that the parallel reconstruction allows reducing the memory usage and the number of operations on each node optimizing therefore the whole process. In Table 3 **np** corresponds to the number of used processors.

Finally, Fig. 4 shows the images reconstructed in parallel from different number of equally spaced projections, and the quantitative results of quality comparison between original and reconstructed images are summarized in Table 4.

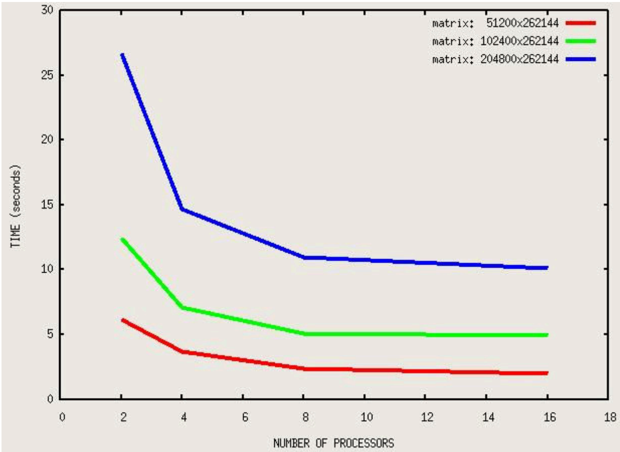


Figure 3. Reconstruction time (in seconds) for the image of 512x512 pixels from different number of projections and processors; the matrix dimensions corresponds: rows - to the number of the projections; columns - to the size (512x512 pixels) of the reconstructed image

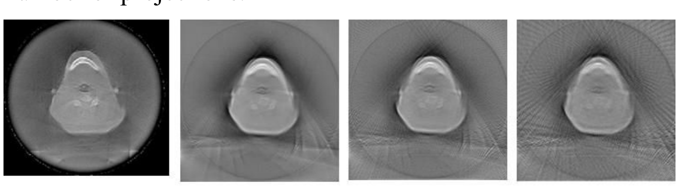
To compare the images we use Mean Square Error (MSE) and Peak Signal to Nose Ratio (PSNR) functions [9].

TABLE III. PETSC PERFORMANCE ON EACH NODE IN THE PARALLEL RECONSTRUCTION OF IMAGES ON EULER CLUSTER

	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
Memory usage (Kbytes)	759727.9	380391.5	190271.4	89649.4
Mpi Messages	2.70e+01	8.20e+01	1.80e+02	3.80e+02

It is needed to mention that usually post processing procedure (as filtering) is applied to the reconstructed image in order to improve the quality. In this work we compare images right after the reconstruction stage without any filtering.

In Table 4 the number of projections is calculated by multiplying the number of sensors and angles used to collect the projections. The results show that the iterative algorithms are capable to reconstruct images of a good quality from less number of projections.



a) b) c) d)

Figure 4. Reconstructed images: a) original image; b), c), d) iterative reconstruction from 400, 200 and 100 angles at the iteration 12 when the given tolerance is achieved.

TABLE IV. QUALITY COMPARISON BETWEEN ORIGINAL AND RECONSTRUCTED IMAGES

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

V. CONCLUSIONS

In this work the possibility of the image reconstruction with iterative algebraic methods from projections and the usage of the PETSc library for the optimization of the whole process have been analyzed. The obtained results show the capacity of the algebraic methods to reconstruct images with low computational cost.

The usage of the PETSc library facilitates a great deal and optimizes the whole work in the parallel reconstruction of images. We expect more significant results in undergoing work of 3D image reconstruction when a huge amount of computing is involved.

ACKNOWLEDGMENT

We wish to thank Dr. Sergio Díez, Head of the Radiology and Radiophysics Protection Service of the hospital Clinico Universitario, for the collaboration in carrying out this work.

We also grateful to the Alicante University for allowing to test our algorithms on Euler cluster system.

REFERENCES

[1] R. C. Gonzales, R. E. Woods, *Digital Image Processing*, 3rd ed., Prentice Hall, 2008

[2] G. Wang, H.Yu, and B. De Man, "An outlook on X-ray CT research and development". Medical Physics, vol. 35(3), pp. 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., vol. 25, pp. 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., vol. 43, pp. 729-737, 1998.

[6] R. G. Wells, M. A. King, P. H. Simkin, P. F. Judy, A. B. Brill, H. C. Gi®ord, 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, vol. 41, pp. 1391-1399, 2000.

[7] N. Sinha and J. T. W. Yeow, "Carbon nanotubes for biomedical applications". IEEE Trans. Nano., vol. 4(2), pp. 180-196, 2005.

[8] http://acts.nersc.gov/petsc/index.html. Last access Jan. 2012.

[9] L. Flores, V. Vidal, P. Mayo, F. Rodenas, G. Verdú, " Iterative reconstruction of CT images with PETSc". BMEI, vol 1, pp. 343-346, 2011.