35th Annual International Conference of the IEEE EMBS
Osaka, Japan, 3 - 7 July, 2013

Iterative Reconstruction of CT Images on GPUs*

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

Abstract— **Although widely used in nuclear medicine (gamma-camera, single photon emission computed tomography (SPECT), positron emission tomography (PET)), iterative reconstruction has not yet penetrated in CT. The main reason for this is that data sets in CT are much larger than in nuclear medicine and iterative reconstruction then becomes computationally very intensive. Graphical Processing Units (GPUs) provide the possibility to reduce effectively the high computational cost of their implementation. It is the goal of this work to develop a GPU-based algorithm to reconstruct high quality images from under sampled and noisy projection data.**

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 reconstruction problem has been resolved by Johan Radon in 1917 [1]. Since then, technological and theorical advances have been the moving force for constant interest in different reconstruction methods and their implementation. In the implementation of an algorithm, it is possible to plan how to optimize its execution and achieve better performance. That is why parallel computing that distributes calculation processes efficiently is important. It is has been recognized that the graphic processing unit (GPU) can be exploited for improving computational efficiency [2] and using the graphic processing unit to improve algorithm performance has become increasingly popular.

The filtered backprojection method is one of the analytical methods and it is used in most of today's cone-beam CT scanners as the standard reconstruction approach. Interestingly, the GPU implementation of the filtered backprojection algorithm has been more widely investigated in the computed tomography literature [3].

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

L. A. Flores is with the Departamento de Sistemas Informáticos y Computación, Universitat Politècnica de València, 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, Universitat Politècnica de València, 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, Universitat Politècnica de València, Camino de Vera s/n, 46022, Valencia, España (e-mail: frodenas@mat.upv.es).

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

On the other hand, iterative methods provide the optimal reconstruction in noisy conditions in the image. In CT, it is common to find incomplete set of no equally spaced projections. In these cases, according to the research ([6], [7], [8]), iterative reconstruction techniques provide images with better quality.

Acceleration of iterative reconstruction is an active area of research. Stone *et al*. [9] describe the accelerated reconstruction algorithm on graphical processing units (GPUs) for advanced magnetic resonance imaging (MRI). They reconstruct images of 128^3 voxels in over one minute. Johnson and Sofer [10] propose a parallel computational method for emission tomography applications that is capable of exploiting the sparsity and symmetries of the model and demonstrate that such a parallelization scheme is applicable to the majority of iterative reconstruction algorithms. The time needed for the reconstruction of thick-slices images (128x128x23 in voxels) is over 3 minutes. Pratx *et al* [11] show results of iterative reconstruction using GPU in PET. The required time on a single GPU to reconstruct an image of 1603 voxels is 8.8 second. Multi GPU implementation of tomography reconstruction accelerates reconstruction of images 350x350x9 up to 67 seconds on a single GPU and 32 seconds on four GPUs [12].

In our previous work we have analyzede the usage of Extensive Toolkit for Scientific computation (PETSc) [13] in parallel image reconstruction. It has been shown that 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 present the GPU based implementation of the iterative algorithm for the image reconstruction.

The outline of this paper is as follows. In section 2, we present briefly mathematical aspects of the problem and a GPU implementation of the algorithm.The test results are presented in section 3 and section 4 summarizes our conclusions.

II. Methodology

A. Mathematical aspects

It is possible to consider the problem of image reconstruction from projections as a system of linear equations of the form:

$$Ax = P, \tag{1}$$

where the system matrix $\boldsymbol{A}$ simulates computer tomography functioning and its elements depend on the projection number and the angle at which the projections have been taken and may not be square, $\boldsymbol{x}$ is a column matrix whose

values represent intensities of the image, and the column matrix $\boldsymbol{P}$ represents projections collected by a scanner.

For a given angle, we assume that the number of projections ranges from 1 to m. If there are k different angles, then in (1) $\boldsymbol{P}$ is a column matrix with mxk elements, $\boldsymbol{x}$ is a column matrix with n^2 elements and $\boldsymbol{A}$ is an $mkxn^2$ rectangular matrix. Many properties of the reconstructed image depend on the approximations when calculating the system matrix. We used Siddon algorithm to compute the elements of the matrix. It has been shown [14] that this method gives good results in approximating the system matrix in a rectangular grid.

We implemented the iterative Least Square QR method (LSQR) to solve the system (1) by minimizing: $\min\|Ax-P\|_2$. The matrix $\boldsymbol{A}$ is normally large and sparse and is used only to compute products of the form $\boldsymbol{Av}$ and $\boldsymbol{A^Tu}$ for various vectors $\boldsymbol{v}$ and $\boldsymbol{u}$.

In practice, $\boldsymbol{A}$ is a rectangular nonsymmetrical sparse matrix and therefore it is recommendable to use compact storage format as Compact Sparse Row (CSR) or Compact Sparse Column (CSC), that allow to store only nonzero elements. The dimensions of $\boldsymbol{A}$ grow proportionally to the resolution of the image to be reconstructed and the number of projections, increasing therefore the computational cost.

In this paper we attempt to develop an algorithm suitable for GPU parallelization.

B. GPU implementation of the algorithm

The main steps of the reconstruction process are shown in Fig. 1.

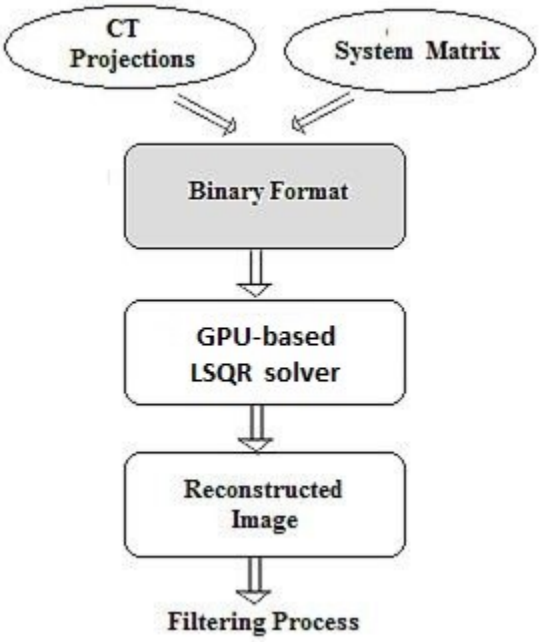

Figure 1. Reconstruction process with LSQR solver in CUDA parallel programming model

The system matrix and the projections are generated previously and stored in binary format.

Special GPUs card dedicated for scientific computing, like the NVIDIA Tesla M2050 card is used in this paper to carry out the experiment. Such a GPU card has a total number of 448 cuda cores with 3GB ECC memory, shared by all processor cores. Utilizing such a GPU card with tremendous parallel computing ability can considerably elevate the computation efficiency of our algorithm.

NVIDIA also introduced CUDA™ [15], a general purpose parallel computing architecture – with a new parallel programming model and instruction set architecture – that leverages the parallel compute engine in NVIDIA GPUs to solve many complex computational problems in a more efficient way than on a CPU. CUDA comes with a software environment that allows developers to use C or C++ as high-level programming languages.

We also use CUBLAS [16] and CUSPARSE [17] libraries that allow the user to access the computational resources of NVIDIA Graphical Processing Unit (GPU). The CUBLAS library is an implementation of BLAS (Basic Linear Algebra Subprograms) on top of the NVIDIA®CUDA™ runtime. To use the CUBLAS library, the application must allocate the required matrices and vectors in the GPU memory space, fill them with data, call the sequence of desired CUBLAS functions, and then upload the results from the GPU memory space back to the host. The CUBLAS library also provides helper functions for writing and retrieving data from the GPU.

The NVIDIA® CUDA™ CUSPARSE library contains a set of basic linear algebra subroutines used for handling sparse matrices and is designed to be called from C or C++. These subroutines include operations between vector and matrices in sparse and dense format, as well as conversion routines that allow conversion between different matrix formats.

CUBLAS and CUSPARSE are written using the CUDA parallel programming model and help to overcome the challenge to develop application software that transparently scales its parallelism to leverage the increasing number of processor cores.

III. RESULTS AND DISCUSSIONS

For experimental purposes we used the real projections and the reference images acquired from the Hospital Clinico Universitario in Valencia. We worked with fan-beam projections collected by a 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 symmetric structure of the system matrix. In order to analyze the capacity of iterative algorithms to reconstruct images from less number of projections, 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.

The results have been measured on a one GPU card of the cluster system Euler that belongs to the Alicante University in Spain. The GPU computing node consists of 2 x CPU Intel Xeon X5660, each with 6 cores of 2,80 GHz and 3 x GPU NVIDIA TESLA M2050 with 448 cores and 3GB memory each of them.

For the images of 256x256 and 512x512 pixels the solving time of the system (1) on a one CPU and a one GPU card is given in Table 1. The GPU time corresponds to the execution time of operations on a device not taking into account time spent in queues. The standard deviation of the results after running the application ten times is 2.9e-004. 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 (256x256 and 512x512 pixels).

TABLE 1. THE RECONSTRUCTION TIME OF IMAGES ON CPU AND GPU ON EULER CLUSTER

System Matrix (rows x columns)	CPU (seconds)	One GPU card (seconds)
M1 = (256x100) x (256x256)	2.7	0.1569
M2 = (256x200) x (256x256)	5.3	0.3056
M3 = (256x400) x (256x256)	10.5	0.6127
M4 = (512x100) x (512x512)	12.3	0.6584
M5 = (512x200) x (512x512)	24.4	1.2741

The results show the efficiency of the algorithm based on a GPU parallel computing ability. SpeedUp of 19.2 has been achieved to reconstruct images of 512x512 pixels. Comparing with the best results presented in [12] (reconstruction of 350x350x9 images requires 67 seconds on a single GPU), we see that our implementation (considering 2D case) allows to reconstruct images with higher resolution and in much less time.

TABLE II. QUALITY COMPARISON BETWEEN REFERENCE AND RECONSTRUCTED IMAGES OF 512X512 PIXELS

N of Angles	MSE	PSNR
100	0.0143	66.9300
200	0.0110	67.8019
400	0.0100	68.3378

Also the quality comparison between reference images and images reconstructed from different number of angles has been made and the quantitative results are summarized in Table 2. To compare the images Mean Square Error (MSE) and Peak Signal to Nose Ratio (PSNR) functions have been used. The results show that iterative LSQR algorithm allows the reconstruction of images of a good quality from less number of angles or, consequently, projections. This might be useful in situations when the complete set of projections is not physically possible. For example, in scanners that might be used to realize urgent examination at any place. They do not provide equally spaced data, so, the algebraic reconstruction is more suitable for these devices.

Fig. 2 shows the images reconstructed in parallel from different number of equally spaced projections. It is needed to be mentioned that usually post processing procedure (as filtering) is applied to the reconstructed image in order to improve the quality. In this work we present the images right after the reconstruction stage without any filtering.

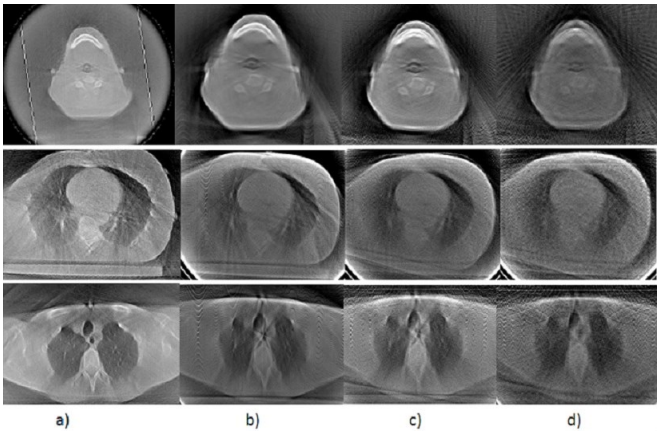

Figure 2. Reconstructed images (512x512 pixels): a) reference images; b), c), d) iterative reconstruction from 400, 200 and 100 angles at the iteration 12 when the given tolerance is achieved

Finally, Fig. 3 illustrates the capacity of the iterative algorithm to reconstruct images from incomplete and unequally spaced data while FBP fails to do that.

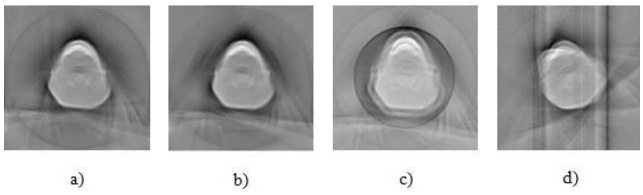

Figure 3. Reconstruction from incomplete data: (a) LSQR and (c) FBP-reconstruction from the set with removed angles (256 sensors x 170 angles); (b) LSQR and (d) FBP - reconstruction from the set with removed sensors (226 sensors x 200 angles). Removed angles and sensors have been chosen arbitrary.

IV. CONCLUSIONS

The GPU-based iterative algorithm of image reconstruction presented in this paper shows that the iterative methods are capable to reconstruct images with low computational cost.

CUDA parallel programming model with CUBLAS and CUSPARSE libraries allow overcoming the challenge to solve complex computational problems and take advantage of the computational resources of the NVIDIA graphics processor (GPU).

We believe that more significant results could be achieved in 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. S. Deans, *The Radon transform and some of its applications*. Dover Publications, INC. Mineola, New York, 2007.

[2] K. Mueller, F. Xu, and N. Neophytou, "Why do GPUs work so well for acceleration of CT?," in SPIE Electronic Imaging '07 (Keynote, Computational Imaging V), San Jose, CA, 2007.

[3] F. Xu and K. Mueller, "Accelerating popular tomographic reconstruction algorithms on commodity PC graphics hardware," IEEE Transaction of Nuclear Science, 2005.

[4] 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.

[5] 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.

[6] 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.

[7] R. G. Wells, M. A. King, P. H. Simkin, P. F. Judy, A. B. Brill, H. C. Gifford, 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.

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

[9] Stone S. S., Haldar J. P., Tsao S.C., Hwu W.-m W., Sutton B. P., Liang Z. P., 2008. Accelerating advanced MRI reconstructions on GPUs. Journal of Parallel and Distributed Computing, vol. 68, issue 10, 1307-1318.

[10] Johnson C.A., Sofer. A., 1999. A data-parallel algorithm for iterative tomographic image reconstruction. Frontiers of Massively Parallel Computation, pp. 126-137.

[11] Pratx G., Chinn G., Olcott P.D., Levin C. S., 2009. Fast, Accurate and Shift-Varying Line Projections for Iterative Reconstruction Using the GPU. IEEE Transactions on Medical Imaging, 28(3), pp. 435-445.

[12] Jang B, Kaeli D., Do S., Pien H., 2009. Multi GPU implementation of iterative tomographic reconstruction algorithms. Biomedical Imaging: From Nano to Macro, pp. 185-188.

[13] L. Flores, V. Vidal, P. Mayo, F. Rodenas, G. Verdú, "Fast Parallel Algorithm for CT Image Reconstruction,'' Proceedings of 34th Annual International Conference of the IEEE Engieneering in Medicine & Biology Society. August 28-September 1, 2012 San Diego, p. 4374-4377. ISBN: 978-1-4244-4120-4

[14] M. T. Cibeles Mora, "Metodos de reconstruccion volumetrica algebraica de imágenes tomograficas." PhD thesis, UPV, Valencia, Spain, 2008.

[15] http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Programming_Guide.pdf. Last access 11.2012.

[16] http://developer.download.nvidia.com/compute/DevZone/docs/html/CUDALibraries/doc/CUBLAS_Library.pdf. Last access 11.2012.

[17] http://developer.download.nvidia.com/compute/DevZone/docs/html/CUDALibraries/doc/CUSPARSE_Library.pdf. Last access 11.2012.