Procedia Computer Science
Volume 51, 2015, Pages 703–712
ICCS 2015 International Conference On Computational Science

Iterative Reconstruction from Few-View Projections

Liubov Flores[2*], Vicent Vidal[1] and Gumersindo Verdú[1]
[1]*Universitat Politècnica de València, Valencia, Spain*
[2]*Universidad Peruana Cayetano Heredia, Lima, Perú*
liuflo@posgrado.upv.es, vvidal@dsic.upv.es, gverdu@iqn.upv.es

Abstract
In the medical imaging field, iterative methods have become a hot topic of research due to their capacity to resolve the reconstruction problem from a limited number of projections. This gives a good possibility to reduce radiation exposure on patients during the data acquisition. However, due to the complexity of the data, the reconstruction process is still time consuming, especially for 3D cases, even though implemented on modern computer architecture. Time of the reconstruction and high radiation dose imposed on patients are two major drawbacks in computed tomography. With the aim to resolve them effectively, we adapted Least Square QR method with soft threshold filtering technique for few-view image reconstruction and present its numerical validation. The method is implemented using CUDA programming mode and compared to standard SART algorithm. The numerical simulations and qualitative analysis of the reconstructed images show the reliability of the presented method.

Keywords: CT reconstruction, Iterative algorithms, CUDA C

1 Introduction

In the last three decades, in computed tomography imaging (CT) have been proposed different methods to obtain the internal structure of an object. If analytical methods have been derived from Radon transform (Deans, 2007), in iterative methods, it is optimized an objective function such as a function of maximum likelihood or minimum error (Herman, 2009). All iterative algorithms have in common operations that dominate computational cost.

The reduction of reconstruction time as well as radiation dose without losing the quality of the image are two major problems in CT. 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. Acceleration of iterative reconstruction is an active area of research. Stone *et al.*

[*] Corresponding author: liuflo@posgrado.upv.es

Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2015

doi:10.1016/j.procs.2015.05.188

(Stone, Haldar, Tsao, Hwu, Sutton, & Liang, 2008) 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 (Johnson & Sofer, 1999) 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* (Pratx, Chinn, Olcott, & Levin, 2009) show results of iterative reconstruction using GPU in PET. The required time on a single GPU to reconstruct an image of 160^3 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 (Jang, Kaeli, Do, & Pien, 2009).

In medicine, the diagnosis based on computed tomography is fundamental for the detection of abnormal tissues by different attenuation of X-ray energy, which frequently is not clearly distinguished for radiologists. However, excessive X-ray radiation exposure is not desirable.

Since the development of the large computational capacities in graphical processing units and the ongoing efforts towards lower doses have made in CT, iterative reconstruction has become a hot topic for all major vendors of clinical CT systems. For example, see (Beister, Kolditz, & Kalender, 2012), (Zhao, Hu, & Yang, 2013), (Flores, Vidal, Mayo, Rodenas, & Verdú, 2013), (Flores, Vidal, Mayo, Rodenas, & Verdú, 2013).

In CT, also it is common to find under sampled set of no equally spaced projections. In these cases, iterative methods do not require complete data collection and do provide the optimal reconstruction in noisy conditions in the image. These 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 (Wang G., 2008).

However, for practical use the iterative algorithms must be as efficient as possible. One way to reduce the radiation dose is to lessen the number of rotations during the data acquisition. As a consequence, in the reconstruction appear undesired artifacts. With the development of compressed sensing theory (Donoho, 2006), (Candès, Romberg, & Tao, 2006) compressed sensing based iterative algorithms have drown much attention in medical imaging. Subsequently, many algorithms have been developed and extended in the field of few-view CT image reconstruction. Yu and Wang (Yu & Wang, 2010) adapted a soft-threshold filtering (STF) algorithm for total variation (TV) minimization in image reconstruction. With the aim to eliminate the streak artifacts and preserve the edge structure, Yu and Zeng (Yu & Zeng, 2014) developed iterative reconstruction algorithm based on weighted total difference (WTD) minimization for few-view computed tomography. To solve the model effectively, the soft-threshold filtering method and a fast iterative shrinkage thresholding algorithm have been employed to accelerate the convergence speed.

To update the current reconstruction, both methods use the simultaneous algebraic reconstruction technique (SART) which is a classical reconstruction algorithm in CT imaging (Andersen & Kak, 1984). The high computational cost of the algorithm, especially in 3D reconstruction makes SART difficult for practical uses.

Inspired by these methods and with the aim to reduce the cost and not to lose the quality we propose *Least Square QR* with soft threshold filtering (LSQR-STF) algorithm based on GPUs and evaluate the method in numerical simulation.

The rest of the paper is organized as follows: in section 2, the mathematical aspects of methods used in this work are presented. In the next section, we give the GPU implementation of these algorithms. Then, we describe the methodology used to carry out experiments and present some results. In the final section, we summarize our conclusions.

2 Methods

In literature have been presented various algorithms to resolve the reconstruction problem (see (Herman, 2009) for revision). An algebraic approach to the reconstruction problem is reduced to the lineal system:

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

where the system matrix A simulates computer tomography functioning and its elements depend on the projection number and the angle and may not be square, x is a column matrix whose values represent intensities of the image, and the column matrix P represents projections collected by a scanner.

In this approach, to reconstruct the internal structure of an object is equivalent to solve the system (1) in terms of measured projections.

Many properties of the reconstructed image depend on the approximations when calculating the system matrix in the equation (1). In practice, A is a rectangular no symmetrical sparse matrix and therefore it is recommendable to store only nonzero elements. A could be computed previously, which would accelerate the reconstruction process.

2.1 SART

The simultaneous algebraic reconstruction technique (SART) is considered as a classical reconstruction algorithm in CT imaging (Andersen & Kak, 1984).

The SART-type solution to equation (1) can be written as (Wang & Jiang, 2004) :

$$x^{k+1} = x^k + \lambda^k \frac{1}{\sum_{i=1}^{M} a_{ij}} \sum_{i=1}^{M} \frac{a_{ij}}{\sum_{j=1}^{N} a_{ij}} (p_i - A_i x^k), \tag{2}$$

where $x = \{x_j \mid j = 1, 2, \ldots N\}$ is an image vector of N pixels, $P = \{p_i, \quad i = 1 : M\}$ represents the projection data, $A = \{a_{ij}\}$ is the system matrix of M$x$$N$, whose elements give the length of the segment of the i th X-ray going through the jth image pixel, A_i is the i-th row of A.

The compute operations used in the reconstruction process are pixel-voxel operations. These operations have few dependencies and are executed in large loops. The appropriate platforms for such operations are vector processors or massively parallel architectures and graphical process units (GPUs). Even though implemented on such architecture, the reconstruction time with SART is too big and makes it difficult for practical uses.

2.2 Least Square QR method

CT image reconstruction with few-view projection data is considered as an ill-posed inverse problem. To solve this problem, we have adopted the Least Square QR method (LSQR) (Paige & Saunders, 1982) and compared it to the SART method.

LSQR method solves the system (1) by minimizing $\min\|Ax - P\|_2$. The method is the most reliable algorithm when A is ill-conditioned. Algorithm LSQR is based on bidiagonalization procedure of Golub and Kahan (Golub & Kahan, 1965). It generates a sequence of approximations $\{x_k\}$ such that the residual norm $\|r_k\|_2$ decreases monotonically, where $r_k = P - Ax_k$. The matrix

A is normally large and sparse and is used only to compute products of the form Av and $A^T u$ for various vectors v and u. In this work we use Siddon's algorithm to compute elements of the matrix in a rectangular grid (Siddon, 1985). It has been found that Siddon's algorithm gives a good approximation of the system matrix A (Cibeles Mora Mora, 2008).

The main steps of LSQR are similar to reference (Paige & Saunders, 1982):

- Initialize vectors *u, v, x, w* and scalars η, μ and iterLsqr. The scalars $\alpha_i \geq 0$ and $\beta_i \geq 0$ are chosen so that $\|u_i\|_2 = \|v_i\|_2 = 1$.

$$u_0 = P$$
$$v_0 = A^T u_0$$
$$w_0 = v_0$$
$$x_0 = 0$$
$$\eta_0 = \beta_0$$
$$\mu_0 = \alpha_0$$

- Iteration process:

For i = 1, 2… iterLsqr

  - Bidiagonalization procedure:

$$u_i = Av_{i-1} - \alpha_{i-1} u_{i-1}$$
$$v_i = A^T u_i - \beta_i v_{i-1}$$

  - Update scalars:

$$\rho_i = \sqrt{\mu_{i-1}^2 + \beta_i^2}$$
$$c_i = \mu_{i-1} / \rho_i$$
$$s_i = \beta_i / \rho_i$$
$$\theta_i = s_i \cdot \alpha_i$$
$$\mu_i = -c_i \cdot \alpha_i$$
$$\phi_i = c_i \cdot \eta_{i-1}$$
$$\eta_i = s_i \cdot \eta_{i-1}$$

  - Update vectors

$$x_i = x_{i-1} + (\phi_i / \rho_i) \cdot w_{i-1}$$
$$w_i = v_i - (\theta_i / \rho_i) \cdot w_{i-1}$$

  - Test for stopping criterion

2.3 Soft threshold filtering.

With the aim to eliminate undesired artifacts and preserve the edge structure of the object, we adopted a soft threshold filtering approach for reconstruction from limited number of projections similar to references (Yu & Wang, 2010), (Yu & Zeng, 2014). For an image vector x of N pixels with $N = nxn$, the filtering step for jth pixel in kth iteration is constructed as follows:

$$x_j^k = \frac{1}{4+4\alpha}(q(\omega, x_j^{k-1}, x_{j+n}^{k-1}) + q(\omega, x_j^{k-1}, x_{j+1}^{k-1}) + q(\omega, x_j^{k-1}, x_{j-1}^{k-1}) + q(\omega, x_j^{k-1}, x_{j-n}^{k-1}) + \\ + \alpha((q(\omega, x_j^{k-1}, x_{j+n+1}^{k-1}) + q(\omega, x_j^{k-1}, x_{j+n-1}^{k-1}) + q\left(\omega, x_j^{k-1}, x_{j-n-1}^{k-1}) + q(\omega, x_j^{k-1}, x_{j-n+1}^{k-1})\right), \tag{3}$$

where

$$q(\omega, y, z) = \begin{cases} (y+z), & if \quad |y-z| < \omega \\ y - \omega/2, & if \quad y - z \geq \omega \\ y + \omega/2, & if \quad y - z \leq \omega. \end{cases}$$

The threshold $\omega = \max_i |r_i|, \quad r = A^T(P - Ax)$.

In the filtering step, it is measured horizontal and vertical sparsity as well as diagonal continuity. The parameter α determines the contribution of diagonal pixels to the center pixel and plays the balancing role between sparsity and continuity. In our experiments, the best results are obtained for $\alpha = 1.5$.

In summary, we have a LSQR with soft-threshold filtering (LSQR-STF) algorithm in the following pseudo code:

(1) Initialization:

$$k = 0, \quad x^k = 0, \quad \alpha = 1.5$$

(2) Start iteration process:

- Update the current reconstruction using LSQR
- Perform the filtering step using equation (3)
- Return to step (2) until the stopping criterion is satisfied

In our implementation, the filtering step is applied for every iteration of LSQR. However, various combinations of LSQR and STF steps are possible.

3 GPU implementation

Computer graphic cards, such as NVIDIA Tesla K20c have been used to carry out the experiment. Such a GPU card has a total number of 2496 cuda cores with 5GB memory, shared by all processor cores. Utilizing such a GPU card with tremendous parallel computing ability considerably elevate the computation efficiency of our algorithms.

NVIDIA also introduced CUDA^T (NVIDIA CUDA Zone), 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 and overcome the challenge to develop application software that transparently scales its parallelism to leverage the increasing number of processor cores.

As it was mentioned in section 2.2, the system matrix A is normally large and sparse and is used only to compute products of the form Av and A^Tu for various vectors v and u. One optimization technique considered in the implementation of LSQR is utilization of functions of CUBLAS and CUSPARSE libraries (NVIDIA CUDA Zone) to handle matrix-vector operations. CUBLAS 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 download 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.

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.

The most effective and important optimization opportunities are presented in exploration and effective use of the device memory. We used global memory of the device to allocate input data. However, the read only data have been allocated in constant memory and the fastest shared memory was used for temporary results whenever it was possible.

Selecting the correct size for a thread block is particularly key for performance since it determines the number of threads that can be run simultaneously. We used 512 threads per block and chose to generate the number of blocks on a pixel bases.

4 Results and discussions

In the experiment, we have analyzed two iterative algorithms (LSQR-STF and SART) implemented on one GPU card. The results have been obtained on the system *gpu.dsic.upv.es with* CPU of 2.6 GHz and NVIDIA TESLA K20c GPU. The system belongs to the Department of Systems and Computation of Polytechnic University of Valencia.

We worked with real projections and reference images acquired from the Hospital Clinico Universitario in Valencia. The experimental data have been collected by the scanner with 512 sensors in the range 0 - 180 with 0.9 degree spacing. The sensors in this scanner are equally spaced along a line of 50 cm. To be able to reconstruct an image with the iterative methods, we have completed the given set up to 360 degrees using the symmetry structure of the system matrix. Due to the physical conditions, real data contain noise and is not similar to ideal projections. We wanted to analyze the capacity of our algorithm in parallel reconstruction of images from a limited 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.

We tested the LSQR-STF algorithm and compared the results to SART. The reconstruction time (in seconds) using different number of projections for images of 256x256 pixels is given in Table 1. The stopping criterion was defined as reaching the maximum iteration number 10. In the system matrix, the number of rows is obtained by multiplying the number of used sensors and angles; the number of columns corresponds to the size of the reconstructed image. The results show that the high computational cost of SART makes it unpractical for clinical use. At the same time, to reconstruct an image even of 50 slices (512x512x50 voxels) with LSQR take only **4.6** seconds.

Number of projections	SART	LSQR	LSQR-STF
50	260 sec	0.07 sec	0.09 sec
100	520 sec	0.14 sec	0.17 sec
200	1031 sec	0.28 sec	0.30sec

Table 1: Matrix size reconstruction time dependence for image 256x256 by SART, LSQR and LSQR-STF after 10 iterations

Figure 1 shows images reconstructed by LSQR-STF and SART from different number of projections after 10 iterations. In Figure 2, it is presented SART reconstruction of the same image after 400 iterations. Comparing these results, we conclude that within much shorter time LSQR obtains visually the same result as SART after 400 iterations (which requires 1h 30 min for reconstruction of an image of 256x256 pixels from 100 projections.)

To evaluate quality of the reconstructions, the following functions have been used:

- Mean square error:

$$MSE = \sum_{i=1}^{n}\sum_{j=1}^{n}[I_1(i,j) - I_2(i,j)]^2,$$

- Peak signal-to-noise ratio:

$$PSNR = \frac{1}{nn}\log_{10}\frac{MAX_I^2}{MSE},$$

where n corresponds to the resolution (nxn pixels) of the reconstructed image, I_1 and I_2 are reference and reconstructed images, MAX_I is the maximum possible pixel value of the image. The results are summarized in Table 2 and Figure 3.

5 Conclusions

To solve the reconstruction problem in the field of CT imaging, two iterative methods have been evaluated. The results show that the Least Square QR method combined with soft threshold filtering is capable to reconstruct images using under sampled set of projections with comparatively acceptable quality. Soft threshold filtering technique, considering horizontal and vertical sparsity and diagonal continuity, helps to preserve better edge structure information.

The GPU based implementation of LSQR-STF lowers the computational cost significantly, allowing reducing a data acquisition process as well as radiation dose for patients.

Iterative algorithms are playing increasing role in such important imaging modality as computer tomography due to their capacity to reconstruct images from a limited data set which results in reduction of reconstruction time. In the near future, we will investigate LSQR method in 3D reconstruction where a huge amount of computing is involved.

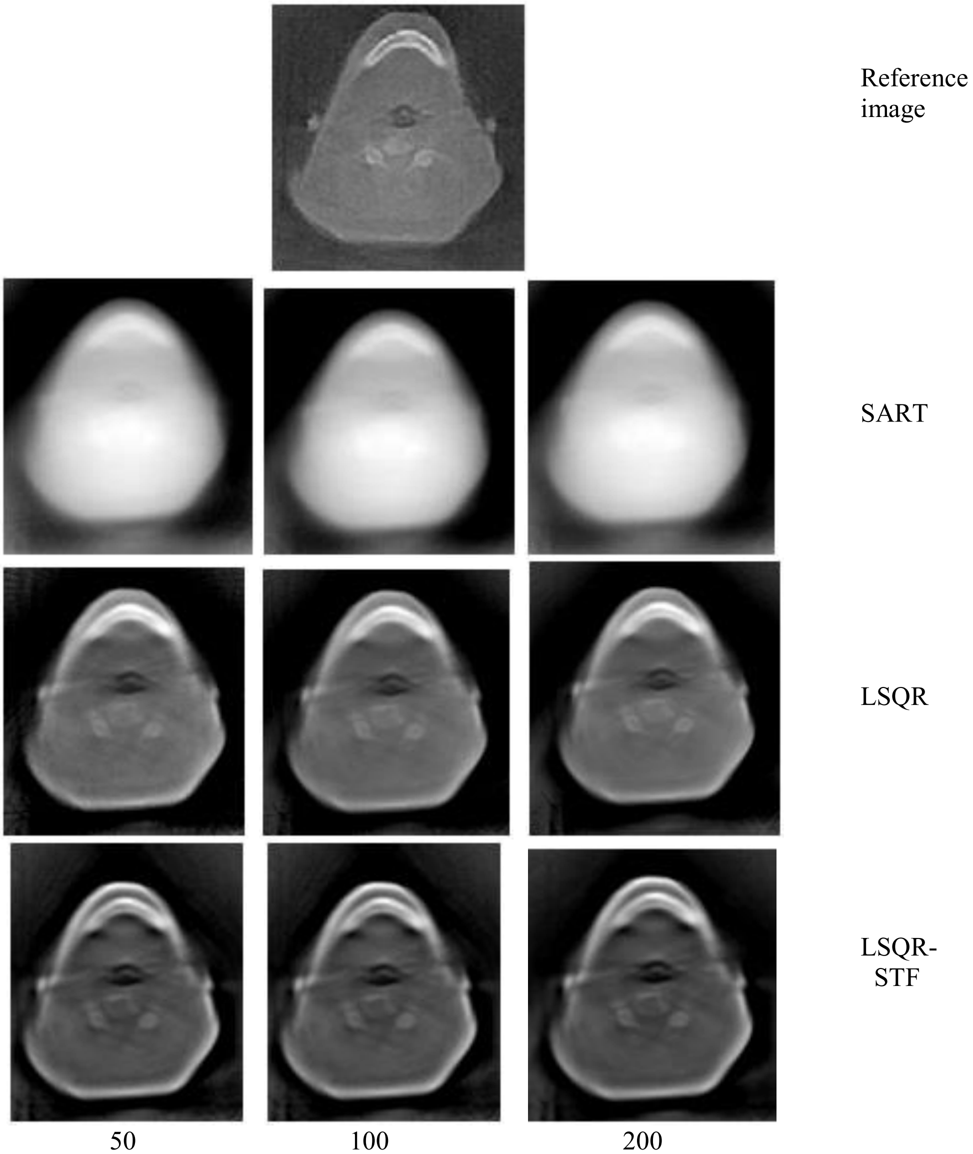

Figure 1: Reconstructed images by SART and LSQR_STF from 50, 100 and 200 projections after 10 iterations

Acknowledgments

This work has been supported by Universitat Politècnica de València and partially funded by ANITRAN PROMETEOII/2014/008 of the Generalitat Valenciana of Spain.

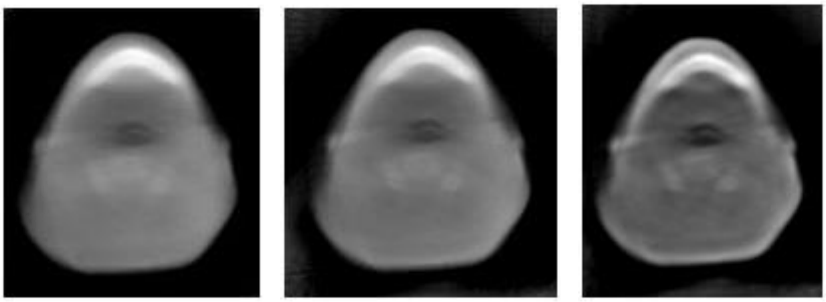

Figure 2: Comparison of relatively acceptable reconstructions by SART and LSQR-STF from 50 projections; the reconstruction time by SART after 200 iterations is 1h 30 min; LSQR reconstructs the same image after 10 iterations in 0.3 seconds

Image 256x256	MSE	PSNR
SART	0.0436	61.7334
LSQR	0.0886	58.6587
LSQR-STF	0.0307	63.2576

Table 2: Quality comparison of reconstructed images from 50 projections

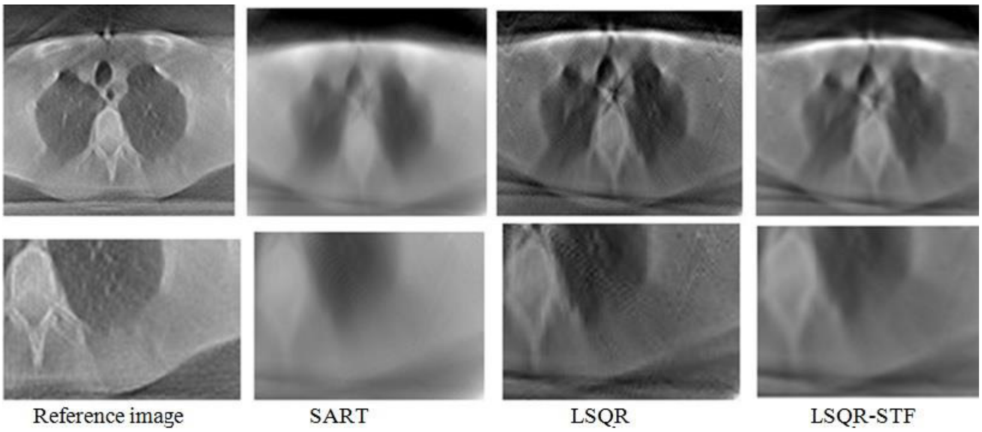

Figure 3: Reconstructions by SART, LSQR and LSQR-STF and zoom-in views of them. By LSQR-STF, the sharp and clear edges can be preserved better.

References

Andersen, A., & Kak, A. (1984). Simultaneous algebraic reconstruction technique (SART): A superior implementation of the ART algorithm. *Ultrasonic Imaging* , 81-94.

Beister, M., Kolditz, & Kalender, W. (2012). Iterative reconstruction methods in X-ray CT. *Physica Medica-European Journal of Medical Physics , 28*, 94-108.

Candès, E., Romberg, J., & Tao, T. (2006). Stable signal recovery from incomplete and inaccurate measurements. *Communication on Pure and Applied Mathematics , 59*, 1207-1223.

Cibeles Mora Mora, T. (2008). Métodos de reconstrucción volumétrica algebraica de imágenes tomográficas. *Tesis PhD* .

Deans, R. (2007). *The radon transform and some of its applications.* New York: Dover Publisher.

Donoho, D. (2006). Compressed sensing. *IEEE Transactions on Information Theory , 52*, 1289-1306.

Flores, L., Vidal, V., Mayo, P., Rodenas, F., & Verdú, G. (2013). CT image reconstruction based on GPUs. *Procedia Computer Science , 18*, 1412-1420.

Flores, L., Vidal, V., Mayo, P., Rodenas, F., & Verdú, G. (2013). Parallel CT Image Reconstruction based on GPUs. *Radiation Physics and Chemistry* , 247-250.

Golub, G., & Kahan, W. (1965). Calculating the singular values and pseudoinverse of a matrix. *SIAM Journal. Numerical Analisis , 2*, 205-224.

Herman, G. (2009). *Fundamentals of computerized tomography: Image reconstruction from projections.* 2nd ed. Springer.

Jang, B., Kaeli, D., Do, S., & Pien, H. (2009). Multi GPU implementation of iterative tomographic reconstruction algorithms. *Biomedical Imaging: From Nano to Macro* , 185-188.

Johnson, C., & Sofer, A. (1999). A data-parallel algorithm for iterative tomographic image reconstruction. *Frontiers of Massively Parallel Computation* , 126-137.

NVIDIA CUDA Zone. (n.d.). Retrieved October 2014, from Cuda C Programming Guide: https://developer.nvidia.com/cuda-downloads

NVIDIA CUDA Zone. (n.d.). Retrieved October 2014, from GPU Accelerated Libraries. cuBLAS: https://developer.nvidia.com/cublas

Paige, C., & Saunders, M. (1982). An algorithm for Sparse Linear Equations and Sparse Linear Squares. *ACM Trans. Math. Software , 8* (1), 43-47.

Pratx, G., Chinn, G., Olcott, P., & Levin, C. (2009). Fast, Accurate and Shift-Varying Line projections for Iterative Reconstruction Using the GPU. *IEEE Transactions on Medical Imaging , 28* (3), 435-445.

Siddon, R. (1985). Fast calculation of exact radiological path length for a three dimentional CT array. *Medical Physics , 12*, 252-255.

Stone, S., Haldar, J., Tsao, S., Hwu, W., Sutton, B., & Liang, Z. (2008). Accelerating advanced MRI reconstructions on GPUs. *Journal of Parallel and Distributed Computing , 68* (10), 1307-1318.

Wang G., Y. H. (2008). An outlook on X-ray CT research and development. *Medical Physics , 35* (3), 1051-1064.

Wang, G., & Jiang, M. (2004). Ordered-subset simultaneous algebraic reconstruction techniques (OS-SART). *J. X-ray Sci. Technol. , 12*, 169-77.

Yu, H., & Wang, G. (2010). A soft-threshold filtering approach for reconstruction from a limited number of projections. *Physics in Medicine and Biology , 55*, 3905-3916.

Yu, W., & Zeng, L. (2014). A novel weighted total diference based image reconstruction algorithm for few-view computed tomography. *PLos ONE , 9* (10), 1-10.

Zhao, X., Hu, J., & Yang, Y. (2013). GPU based iterative cone-beam CT reconstruction using empty space skipping technique. *Journal of X-Ray science and Technology , 21*, 53-69.