Few-View CT Image reconstruction via Least-Squares Methods: assessment and optimization[*]

Mónica Chillarón,[*,a] Vicente E. Vidal,[a] Gumersindo J. Verdú,[b] and Gregorio Quintana-Ortí[c]

[a]*Depto. de Sistemas Informáticos y Computación, Edificio 1F, Camí de Vera S/n Universitat Politècnica de València, València, 46022, Spain.*

[a]*Instituto de Seguridad Industrial, Radiofísica y Medioambiental Universitat Politècnica de València, València, 46022, Spain.*

[c]*Depto. de Ingeniería y Ciencia de Computadores Universidad Jaime I, Castellón, 12071, Spain.*

[*]Email: mnichipr@inf.upv.es

Number of pages:	23
Number of tables:	7
Number of figures:	6

[*]This paper is intended for consideration in the ICRS 14/RPSD 2022 special issue.

Abstract

The use of iterative algebraic methods applied to the reconstruction of Computed Tomography (CT) Medical Images is proliferating to reconstruct high-quality CT images using far fewer views than through analytical methods. This would imply reducing the dose of X-rays applied to patients who require this medical test. Least-squares methods are a promising approach to reconstruct the images with few projections obtaining high quality. In addition, since these techniques involve a high computational load, it is necessary to develop efficient methods that make use of high-performance computing (HPC) tools to accelerate reconstructions. In this paper, three Least-Squares methods are analyzed: LSMB (Least-Squares Model Based), LSQR (Least Squares QR) and LSMR (Least Squares Minimal Residual), to determine whether the LSMB method provides a faster convergence and thus lower computational times. Moreover, a block version of both the LSQR and LSMR methods was implemented. With them, multiple right-hand sides (multiple slices) can be solved at the same time, taking advantage of the parallelism obtained with the implementation of the methods using the Intel Math Kernel Library (MKL). The two implementations are compared in terms of convergence, time, and quality of the images obtained, reducing the number of projections and combining them with a regularization and acceleration technique. The experiments show how the implementations are scalable and obtain images of good quality from a reduced number of views, being the LSQR method better suited for this application.

Keywords — CT, Reconstruction, Image Quality, Least-Squares, Algebraic Methods

I. INTRODUCTION

Computed Tomography (CT) imaging tests provide enormous benefits in the early detection and early diagnosis of various diseases. However, there are concerns about the possible side effect of a high exposure to ionizing radiation, such as an increased risk of developing cancer, genetic diseases, and other radiation-induced diseases. The minimization of the radiation dose applied to the patient is an important challenge and therefore, low-dose reconstruction methods that provide sufficient image quality to make accurate diagnosis are in continuous development.

There are two different strategies to lower the radiation dose. On the one hand, reducing the energy or the tube current to reduce the X-ray dose, and work with iterative reconstruction methods (IR) based on the Filtered Back-Projection (FBP) [1]. These methods are well-known and can follow advanced strategies such as tube current modulation or automatic exposure control in order to automatically reduce the radiation dose and maintain the image quality taking into account the density of the body part being scanned. On the other hand, reducing the number of projections taken so the total exposure time can be reduced and work with algebraic reconstruction methods.

Algebraic methods can be either direct or iterative, and they solve the equations system that models the problem. Direct methods are more exact, but they require a higher number of projections than iterative methods. For instance, the QR decomposition [2] applied to the reconstruction of CT images [3, 4, 5] is a direct algebraic method that obtains optimum-quality images but requires a minimum number of projections. The number depends on the resolution of the images since it is necessary to have full rank in the system matrix that models the CT scanner.

On the other hand, iterative algebraic methods such as ART [6], SART [7, 8] or LSQR [9, 10, 11], combined with different filtering techniques, can work with a smaller number of views. This would mean a lower dose of ionizing radiation absorbed by the patients, which would potentially improve their health. The problem of CT image reconstruction when making an iterative algebraic approach can be considered as the resolution of a least-squares problem, which can be ill-conditioned if few projections are used and thus its resolution can generate artifacts on the image. It would be interesting to deduce the number of minimum projections needed to obtain a high-quality image for a given resolution and to combine this resolution process with regularization techniques that help eliminate artifacts.

In this work, three methods are used to solve the least-squares problem associated with the reconstruction for a different number of projections: LSQR [9], LSMR [12] and LSMB [13]. All these methods are based on the Golub-Kahan bidiagonalization process. When few projections are used, they are combined with the STF [14] regularization method and the FISTA [15] acceleration method. A comparison of the three methods is carried out, in order to compare both the convergence rate and the quality obtain. In addition, parallel CPU implementations of the LSQR and LSMR methods for multi-slice reconstructions are presented and analyzed. A set of real CT images of the DeepLesion [16] dataset were used for the experiments, from which the sinograms were simulated, to later be reconstructed using the three methods.

II. MATERIALS AND METHODS

II.A. CT image reconstruction by Least-Squares

The minimization of the radiation applied to the patient is an important challenge and therefore, methods that use the minimum number of projections necessary to reconstruct a CT image with sufficient quality to make an accurate diagnosis are in continuous development. These methods would allow us to reduce the radiation dose to which the patients are exposed, provided we had a sparse-sampling CT scanner. As of yet, no commercial scanners are using this type of acquisition since developing a scanner that can switch on and off the X-ray source at the desired speed is challenging. However, there are prototypes such as the one presented by Muckley et al. [17], which performs the sparse sampling by blocking the X-ray source until a projection has to be taken. Although this approach is not being used yet, multiple studies claim its advantages. In this case, the computational methods needed to perform the reconstructions are algebraic, since the analytical ones do not perform well due to undersampling. In an algebraic approach to the CT image reconstruction problem, the problem to be solved is modelled as:

$$Ax = g \tag{1}$$

where $A = (a_{i,j}) \in R^{M \times N}$ denotes the sparse matrix of dimensions $M \times N$ being $a_{i,j}$ the contribution of the $i - th$ ray on the pixel j. The dimension M is the product of the number of detectors that the CT scanner has multiplied by the number of projections or views taken. N

denotes the resolution of the image (128×128 pixels, 256×256 pixels, etc). Vector g has $1 \times M$ elements (the sinogram). The solution is x, a vector of size $1 \times N$ that can be transformed into a matrix and is the reconstructed image.

The problem in Eq. (1) can be solved with a least-squares minimization approach by minimizing:

$$\min_x \|Ax - g\|^2 \tag{2}$$

or:

$$\min_x \|Ax - g\|^2 + \lambda^2\|x\|^2 \tag{3}$$

The dimensions and rank of matrix A depend, among other parameters, on the number of projections used when performing the CT acquisition. Therefore, if few-projections are used, the rank will be low and the solution to the problem (2) is not unique. Then, the least-squares problem-solving methods calculate the minimum norm solution, generating a succession of approximations $\{x_k\}$ such that $\|(R_k)\|_2$ or $\|(A^T R_k)\|_2$ decrease monotonically, being $R_k = g - Ax_k$ the residue of the k-th iteration.

In this paper, the behavior of the least-squares method LSMB [13] is analyzed and compared with the more classic methods LSQR [9] and LSMR [12]. The solution of the LSMB method is a convex combination of the solutions of the LSQR and LSMR methods. As per the original research, the LSMB method should work at least as well as the LSQR and LSMR algorithms, and never outperform them by more than a small margin. According to the authors, the iterates it produces are convex combinations of the iterates produced by algorithms LSQR and LSMR. Algorithm LSMB is designed to minimize an objective function closely related to the backward error with every iteration, allowing it to terminate sooner than either LSQR or LSMR. This would be interesting for this application since it would mean reducing the time required to obtain the images. The LSMB method has a different stopping criterion than LSQR and LSMR. It stops when the backward error $ERR(X)/NORM(A) < tol(tol = 1.e - 8)$. However, LSMR and LSQR stop when $NORM(g - Ax)/NORM(g) < tol(tol = 1.e - 6)$.

The performance of the methods will also be analyzed when combined with the STF regularization technique and FISTA acceleration, which are necessary when the number of projections used is very low and therefore the matrix A is rank-deficient.

II.B. Block Implementations of LSQR and LSMR

The most traditional strategy to solve Eq. (1) with Least-Squares methods is to solve one right-hand side (or b) at a time. They could be solved in parallel, but matrix A should be either shared by all processes (which implies a bottleneck in the memory accesses) or replicated for each process (which means more memory usage). A more elegant solution is to model the problem as the resolution of multiple right-hand sides simultaneously. In this case, the problem to solve is the one shown in Eq. (4), where $X = (X^j)$ and $G = (G^j)$ now are matrices with dimensions $N \times S$ and $M \times S$ respectively, and j denotes the $j - th$ column (image or sinogram respectively).

$$AX = G \tag{4}$$

F. Toutounian et al. propose two implementations of LSQR (from now on Block LSQR or BlLSQR) [18] and LSMR (Block LSMR or BlLSMR) [19] for several right-hand sides. The algorithms used are shown in Alg. 1 and 2. This approach takes advantage of matrix-matrix operations to obtain higher efficiency.

In this paper, a high-performance implementation of these algorithms has been developed. The implementation uses Intel's Math Kernel Library (MKL), which employs BLAS and LAPACK routines. It is worth mentioning that three modifications to the original algorithms proposed by the authors have been made. First, we take $Xini$ as the initial solution, and use the residual matrix $G - AXini$ to compute its QR factorization in the initialization. The original algorithms do not take an initial solution. Then, we iterate $iter$ times and return the solution X. We ignore the stopping criteria proposed by the authors in [18, 19]. The number of iterations is fixed because the methods will be combined with the filtering and regularization techniques. Last, we have adapted the algorithms to work with rectangular matrices. In the original algorithms, they work with only square matrices. However, having square matrices for high-resolution images would imply adjusting the number of projections so the system matrix is square. For instance, for a reconstruction with image resolution 256×256 and a scanner with 1024 detectors, we could only solve the problem with 64 projections if the matrix can only be square. By adapting the method

Algorithm 1 Block LSQR algorithm.

```
Input: $A_{MxN}, G_{MxS}, Xini_{NxS}, iter$
Output: $X_{NxS}$
1: $X = Xini$
2: $[U, B] = econQR(G - AX)$
3: $[V, A_1] = econQR(A^T U)$
4: $W_1 = V, \Phi = B, \rho = A_1^T$
5: for $i = 1, \ldots, iter$ do
6:     $W = AV - UA_1^T$
7:     $[U, B] = econQR(W)$
8:     $S = A^T U - VB^T$
9:     $[V, A_1] = econQR(S)$
10:    $[Q, R] = QR(\begin{bmatrix} \rho \\ B \end{bmatrix})$
11:    $\begin{bmatrix} a & b \\ c & d \end{bmatrix} = Q$
12:    $\theta = c^T A_1^T$
13:    $\rho = d^T A_1^T$
14:    $\phi = a^T \phi_1$
15:    $\phi_1 = b^T \phi_1$
16:    $P = W_1 R^{-1}$
17:    $X = X + P\phi$
18:    $W_1 = V - P\theta$
19: end for
```

Algorithm 2 Block LSMR algorithm.

```
Input: $A_{MxN}, G_{MxS}, Xini_{NxS}, iter$
Output: $X_{NxS}$
1: $X = Xini$
2: $[U, B] = econQR(G - AX)$
3: $[V, A_1] = econQR(A^T U)$
4: $B_1 = A_1 B$
5: $\sigma_0 = -B_1$
6: $\sigma_1 = [0]_{s \times s}$
7: $a, b_0, c, d_0 = [0]_{s \times s}$
8: $b, d = [I]_{s \times s}$
9: $P_1, P_0 = [0]_{n \times s}$
10: for $i = 1, \ldots, iter$ do
11:    $W = AV - UA_1^T$
12:    $[U, B] = econQR(W)$
13:    $A_2 = A_1 A_1^T + B^T B$
14:    $S = A^T U - VB^T$
15:    $AUX = V$
16:    $[V, A_1] = econQR(S)$
17:    $\beta = d_0 B_1^T$
18:    $\alpha = c\beta + dA_2$
19:    $\beta_1 = a\beta + bA_2$
20:    $\theta = b_0 B_1^T$
21:    $B_1 = A_1 B$
22:    $[Q, R] = QR(\begin{bmatrix} \alpha \\ B_1 \end{bmatrix})$
23:    $d_0 = d$
24:    $b_0 = b$
25:    $\begin{bmatrix} a & b \\ c & d \end{bmatrix} = Q^T$
26:    $\sigma = -(R^{-T}((\theta^T \sigma_1) + (\beta_1 \sigma_0)))$
27:    $P = (AUX - (P_1\theta) - (P_0\beta_1))(R^{-1})$
28:    $P_1 = P_0$
29:    $P_0 = P$
30:    $X = X + P\sigma$
31:    $\sigma_1 = \sigma_0$
32:    $\sigma_0 = \sigma$
33: end for
```

to rectangular matrices, the number of projections can be selected without restrictions. All in all, with these modifications, the algorithms adapt perfectly to the problem of CT image reconstruction for multiple slices.

In Algorithms 1 and 2, the $QR()$ function is a traditional QR factorization, and $econQR()$ is the economic version of the factorization, where the resulting matrices Q and R only have N rows if $M > N$. The matrices a, b, c, and d have a dimension of $S \times S$ elements. Note that dimension S is the number of right-hand sides to solve. Table I contains the list of the MKL calls performed to solve all the types of matrix operations present in the algorithms.

TABLE I
List of Intel's MKL calls to BLAS or LAPACK functions.

Operation (example)	Description	MKL Call
$G - AX$ (Algs. 1 and 2, line 2) $A^T U$ (Algs. 1 and 2, line 3)	Computes the product of a sparse matrix and a dense matrix. $Y := \alpha AX + \beta X$ where α and β are scalars, A is a sparse matrix, and X and Y are dense matrices.	mkl_sparse_d_mm(...)
$QR(Y)$ (Alg. 1, line 10 and Alg. 2, line 22) $econQR(Y)$ (Algs. 1 and 2, line 2)	Computes the QR factorization of a general m-by-n matrix Y.	LAPACKE_dgeqrf(...)
	Generates the real orthogonal matrix Q of the QR factorization formed by dgeqrf.	LAPACKE_dorgqr(...)
UA_1^T (Alg. 1, line 6 and Alg. 2, line 11)	Computes a matrix-matrix product with general dense matrices.	cblas_dgemm(...)
R^{-1} (Alg. 1, line 16 and Alg. 2, line 27)	Computes the inverse of a triangular matrix.	LAPACKE_dtrtri(...)

The complete reconstruction method proposed in this section consists in the combination of the block LSQR or block LSMR method, combined with the STF and FISTA methods every *iter* iterations. The complete algorithm is shown in Alg. 3. Here, the BlLSQR and BlLSMR function calls correspond to Algorithms 1 and 2, α is a parameter used to balance the gradient sparsity and the gradient continuity usually set to 1, and the function q is the one presented in Eq. (5).

At the beginning of the algorithm, the variable X is the initial solution. If it is not known, $X = 0_{N\times S}$. At the end of the algorithm, X contains the solution obtained; the j-th column contains the j-th reconstructed image with resolution $\sqrt{N} \times \sqrt{N}$. Normally the parameters $iter_{WTDSTF}$ and $iter_{AFISTA}$ are 1 or 2. The value of $iter_{BL}$ usually varies depending on the resolution of the image, as we studied in [20]. The program stops when it reaches the maximum number of iterations given or when the relative residual $r\_residual = \|G - AX\|_2 / \|G\|_2$ is lower than the tolerance. The number of total iterations ($maxiter$) is usually set to 10000 and the desired tolerance to 10^{-6}.

Algorithm 3 Complete iterative reconstruction method

```
Input: $A_{MxN}$, $G_{MxS}$, $X_{NxS}$, $tolerance$, $maxiter$, $iter_{BL}$, $iter_{WTDSTF}$, $iter_{AFISTA}$, $\alpha$
Output: $X$
1: Initialization: $r\_residual = 0, totaliter = 0, X_0 = 0_{Mxs}, r = \sqrt{M}, t_0 = 1$
2: while ($r\_residual > tolerance$) and ($maxiter > totaliter$) do
3:   Step 1: Data constraint
4:   $X = BlLSQR(A, G, X, iter_{BL})$ or $X = BlLSMR(A, G, X, iter_{BL})$
5:   $R = G - AX$
6:   Step 2: Regularization WTD-STF
7:   for $j = 1, \ldots, S$ do
8:     $w_j = \max_i \mid R_i^j \mid$
9:   end for
10:  for $k = 1, \ldots, iter_{WTDSTF}$ do
11:    for $j = 1, \ldots, S$ do
12:      for $i = 1, \ldots, M - (r + 1)$ do
13:        $X^j_{i_{WTDSTF}} = \frac{1}{4+4\alpha}(q(w_j, X_i^j, X_{i+r}^j) + q(w_j, X_i^j, X_{i+1}^j) + q(w_j, X_i^j, X_{i-1}^j) + q(w_j, X_i^j, X_{i-r}^j) +$
           $+\alpha(q(w_j, X_i^j, X_{i+1+r}^j) + q(w_j, X_i^j, X_{i-1+r}^j) + q(w_j, X_i^j, X_{i-1-r}^j) + q(w_j, X_i^j, X_{i+1-r}^j)))$
14:      end for
15:    end for
16:    $X = X_{WTDSTF}$
17:  end for
18:  Step 3: Acceleration AFISTA
19:  for $k = 1, \ldots, iter_{AFISTA}$ do
20:    for $j = 1, \ldots, S$ do
21:      for $i = 1, \ldots, M$ do
22:        $t_1 = \frac{1+\sqrt{1+4t_0^2}}{2}$
23:        $X^j_{AFISTA} = X^j + (\frac{t_0-1}{t_1})(X^j - X_0^j)$
24:      end for
25:    end for
26:    $t_0 = t_1$
27:    $X_0 = X$
28:    $X = X_{AFISTA}$
29:  end for
30:  $r\_residual = \|R\|_2 / \|G\|_2$
31:  $totaliter = totaliter + 1$
32: end while
33: return $X$
```

$$q(w,y,z) = \begin{cases} (y+z)/2 \;\; if \;\; \mid y-z \mid < w \\ y - w/2 \;\; if \;\; y-z \geq w \\ y + w/2 \;\; if \;\; y-z \leq -w \end{cases} \tag{5}$$

II.C. Image Quality Metrics

To measure the quality of the reconstructed images, the metrics SSIM (Structural Similarity Index) and PSNR (Peak Signal-To-Noise Ratio) [21] are used. With SSIM (6), the internal structures (shapes) of the images can be compared with the reference image. With PSNR (7), which is calculated from the Mean Square Error (MSE) (8), the differences of the pixels values compared to the reference image are measured, getting the signal-to-noise ratio. SSIM provides a perceptual metric, more similar to how the human eye perceives the images and in particular, how well the structures are preserved. However, with PSNR the image noise is measured, and it focuses on the pixel values.

$$SSIM = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{x,y} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{6}$$

$$PSNR = 10 * log_{10} \frac{MAX_{I_0}^2}{MSE} \tag{7}$$

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (I_0(i,j) - u(i,j))^2 \tag{8}$$

III. RESULTS

III.A. LSMB, LSQR and LSMR comparison

In order to test the proposed methods, the system matrix and projections data (sinograms) were simulated for image resolution of 512x512 pixels and 30, 45, 60, 90,120, 150 and 180 views. The scanner simulated has 1025 detectors. A set of CT images selected from the dataset DeepLesion [16] were used as reference, re-projecting them with Joseph [22] method to obtain the sinograms.

Table II shows the results of the comparison of the three methods without being combined with regularization techniques. The image quality has been measured with the PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index) metrics [21]. It can be seen that

TABLE II
Performance results of the three methods varying the number of views.

	Iterations			**SSIM**			**PSNR**		
Views	**LSMB**	**LSQR**	**LSMR**	**LSMB**	**LSQR**	**LSMR**	**LSMB**	**LSQR**	**LSMR**
30	468	707	781	0.484	0.484	0.484	26.47	26.47	26.47
45	558	989	1109	0.500	0.500	0.500	28.12	28.12	28.12
60	622	1137	1276	0.595	0.594	0.594	29.40	29.40	29.40
90	664	1662	1893	0.637	0.634	0.634	31.30	31.31	31.31
120	776	2026	2317	0.655	0.648	0.659	32.87	32.87	32.87
150	759	2015	2283	0.703	0.705	0.705	34.37	34.41	34.41
180	708	2367	2680	0.802	0.796	0.797	35.91	36.00	36.00

TABLE III
Performance results of the three methods with STF and FISTA.

	Iterations			**SSIM**			**PSNR**		
Views	**LSMB**	**LSQR**	**LSMR**	**LSMB**	**LSQR**	**LSMR**	**LSMB**	**LSQR**	**LSMR**
30	1560	1385	1555	0.648	0.635	0.657	27.81	27.68	27.79
45	1710	1660	1680	0.702	0.735	0.699	31.08	31.78	31.04
60	3935	3840	4010	0.780	0.714	0.780	34.64	30.69	34.61
90	10000	10000	10000	0.974	0.905	0.979	43.72	41.29	44.77
120	1930	1880	1890	0.873	0.900	0.880	38.22	41.15	38.47
150	1760	2005	1680	0.968	0.969	0.973	42.80	42.15	43.41
180	1480	2010	1390	0.991	0.991	0.992	48.74	51.92	49.49

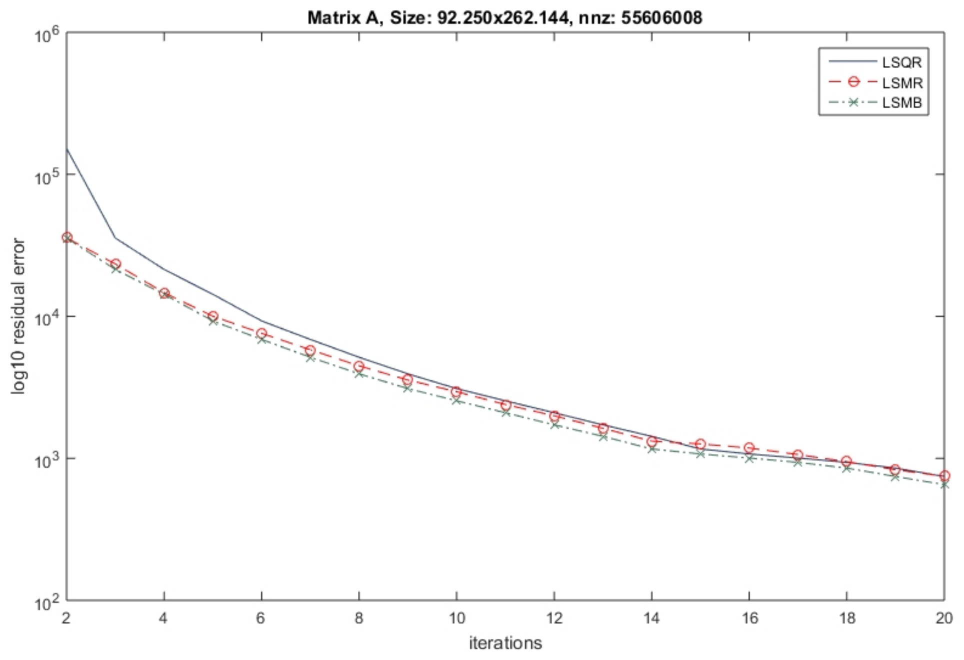
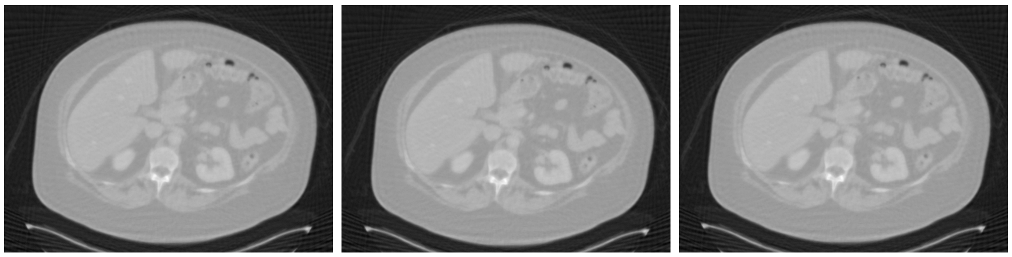


Fig. 1. Example of the residual error evolution.

all methods have a very similar behavior in terms of the quality of the image obtained. However, the LSMB method greatly reduces the number of iterations necessary to converge with respect to the other two. This difference becomes more noticeable when the size of the problem increases, needing only a third of the number of iterations that the LSQR needs for 180 views. Figure 1 shows the evolution of the residual error of the three methods for the case with 90 projections and the first 20 iterations. As can be observed, the LSMB method consistently obtains a lower error than LSQR and LSMR, so it needs fewer iterations to reach convergence.

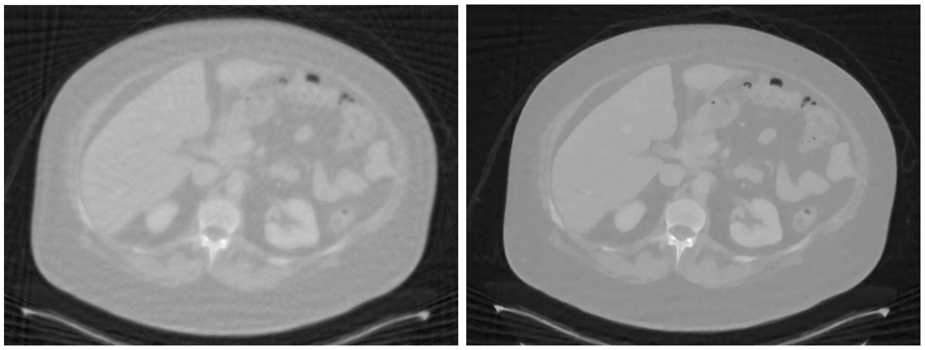
However, the quality is not high for any of the methods, since the SSIM should be closer to 1. This is due to the artifacts produced by the projections reduction. Figure 2 shows the solutions obtained by the three methods with a high number of projections (120). As can be seen, the quality is low, and the images have a lot of artifacts that distort the internal structures. To solve this issue, it is necessary to combine the resolution with the STF regularization and FISTA acceleration.

Table III shows the best results of the combination, varying the views and the number of iterations of LSMB/LSQR/LSMR that are performed before preforming an application of STF and/or FISTA. Generally, the best reconstructions are obtained making 1 application of the STF and FISTA every 5 iterations of the resolution method. As it can be observed, the overall quality increases when combining these techniques with the resolution methods, but the quality obtained by every method is very similar. The number of total iterations is similar in every case too, so the LSMB method is not superior now. Results show that this combination is essential, since the SSIM is now much closer to 1 than in Table II, which means the images have less artifacts. As an example, Figure 3 shows a reconstruction only by LSMB with 90 views, and one combining it with STF and FISTA. The difference is clear, as indicated by the quality metrics, going from a 0.63 SSIM to 0.97, and from a 31.3 PSNR to 43.72. It can be seen that the case with 90 views in Table III did not converge to the desired tolerance, so it reached the maximum number of iterations. This could be due to having an ill-conditioned problem, and the selected projections should probably be changed to improve the convergence by varying the angular step. In any case, this particular results require further analysis. However, as a positive aspect, since the number of iterations is higher, the quality of the final image is also higher than expected. It can be seen that the quality is similar to the 150 projections case (60 more). This implies that the proposed methods could



(a) LSMB (b) LSQR (c) LSMR

Fig. 2. Reconstructions with only the three methods and 120 projections.



(a) LSMB (b) LSMB+STF+FISTA

Fig. 3. Reconstructions with LSMB and 90 projections.

reach higher image quality with fewer projections if the tolerance was lowered.

It must be mentioned that the cost of every method is slightly different. Since LSMB is a convex combination of the solutions obtained by LSQR and LSMR, the computational cost per iteration is higher. For instance, for the case with 120 projections (matrix A of size $123,000 \times 252,144$), the time per iteration using the original Matlab codes were 0.18 seconds for LSMB, 0.1 seconds for LSQR and 0.099 seconds for LSMR. Therefore, the cost of the LSMB is almost the cost of the other two methods combined. All this is added to the fact that when the method is combined with the regularization techniques the number of iterations is higher, it no longer provides the advantage of early stopping and time reduction. For these reasons, the parallel block implementation of the LSMB method has been discarded, selecting only the LSQR and LSMR since they provide similar results.

III.B. Block LSQR and LSMR

The experiments in this section were run using the C programs with MKL 2019.3.199, using a shared memory cluster. In particular, an Intel Xeon E5-4620 processor with 8 cores was employed, booking up to 72GB of RAM memory. The images were also selected from the DeepLesion dataset and re-projected with Joseph's method. In this case, the resolution used was 256×256 pixels and different number of views.

III.B.1. Execution Time and Convergence

To compare the time and iterations it takes both the block LSQR and LSMR algorithms to converge to a minimum tolerance of 1E-06, the internal iterations $iter_{BL}$ was set to 12. That means, one application of the WTD-STF and one of the FISTA method every 12 iterations of LSQR or LSMR.

Table IV shows the time per iteration of each method, measured for a matrix of 60 views and solving 1, 8, 16, 32, 64, or 128 right-hand-sides (slices). It can be observed that it is much more beneficial to solve a large number of slices at the same time when using more than one core, since it is more computationally efficient than to solve just one slice. The LSQR implementation is slightly better than the LSMR, since the times per iteration are slightly lower. In Table V the SpeedUp is shown, calculated as $S_p = T(1)/T(p)$, being T(1) the sequential time, and T(p) the time using p cores. The ideal SpeedUp is the number of cores used. In the table it is observed that the higher SpeedUps are obtained when the number of cores used match the number of slices or right-hand sides. From 8 slices, the SpeedUps start to descend, although in every case they are much higher than when solving just one slice. The ideal Speedup is not reached with either of the methods for the selected number of slices. Nevertheless, the parallel efficiency is good every case, and similar for both of the methods. Thus, it can be determined the implemented code is scalable.

Taking into account the time per iteration, the LSQR gets slightly better performance but it is not significant enough. However, taking into account the number of iterations it takes every method to converge, the difference is much more notable, as can be observed in Table VI. The LSMR method needs a lot more iterations to obtain the same tolerance, in some cases even twice as the LSQR. In conclusion, although the time per iterations is very similar, since the computational load is almost the same for both algorithms, the LSQR method converges faster so the total time to solve the problem is significantly smaller. In addition, it is observed how increasing the number of right-hand sides is beneficial to the total time, since the number of iterations if takes both algorithms to converge is inversely proportional to the number of slices.

To further test the convergence rate of both the proposed methods, a reconstruction with only one slice using the 75-views simulated data only with the BlLSQR or BlLSMR methods was performed, in order to study the evolution of the residual. The number of views is high so the matrix is full-rank and the equations system can be solved without the regularization or acceleration steps.

Figure 4 shows the evolution of the relative residual for both methods during 2000 iterations, on a logarithmic scale. It is worth mentioning that the LSQR reconstruction took 2011 iterations to converge to the desired tolerance and the LSMR took 2292. A zoom of the region from 1 to 40 iterations is also shown. It can be seen that the relative residual of the BlLSQR method is always slightly lower than that of the BlLSMR method except in the first iterations, where it is very similar. From iteration 5 they start to diverge and from there it is clear the BlLSQR converges a bit faster.

On the LSMR original paper [12], they studied the convergence rate of both algorithms, and although the LSMR was usually faster to converge for solving square or rectangular systems, there were cases when it was the LSQR that had better performance. The application analyzed in this work seems to behave better with the BlLSQR algorithm. With these results, it can determined that the LSQR method is faster to converge, more so when the number of slices to be solved simultaneously is increased, as shown in Table VI.

TABLE IV
LSQR/LSMR time (secs.) per iteration with 60 views.

	LSQR				LSMR			
No. of Slices	1 core	2 cores	4 cores	8 cores	1 core	2 cores	4 cores	8 cores
1	0.0655	0.0541	0.0471	0.0449	0.0658	0.0540	0.0470	0.0456
8	0.4862	0.2679	0.1423	0.0873	0.4892	0.2651	0.1439	0.0897
16	1.0039	0.5510	0.3049	0.1946	1.0179	0.5555	0.3089	0.2004
32	2.1059	1.1382	0.6692	0.4458	2.1249	1.1731	0.6775	0.4563
64	4.3858	2.3947	1.3990	0.9325	4.4567	2.4236	1.4211	0.9489
128	9.0922	4.9781	2.8567	1.9422	9.2505	4.9995	2.8971	1.9929

TABLE V
SpeedUp (per iteration) with 60 views.

	LSQR				LSMR			
No. of Slices	1 core	2 cores	4 cores	8 cores	1 core	2 cores	4 cores	8 cores
1	1.00	1.21	1.39	1.46	1.00	1.22	1.40	1.44
8	1.00	1.81	3.41	5.57	1.00	1.85	3.40	5.45
16	1.00	1.82	3.29	5.16	1.00	1.83	3.30	5.08
32	1.00	1.85	3.15	4.72	1.00	1.81	3.14	4.66
64	1.00	1.83	3.13	4.70	1.00	1.84	3.14	4.70
128	1.00	1.83	3.18	4.68	1.00	1.85	3.19	4.64

TABLE VI
Number of iterations with 60 views.

No. of Slices	LSQR	LSMR
1	1680	2220
8	1584	2124
16	1296	2002
32	984	1920
64	612	1812
128	432	1068

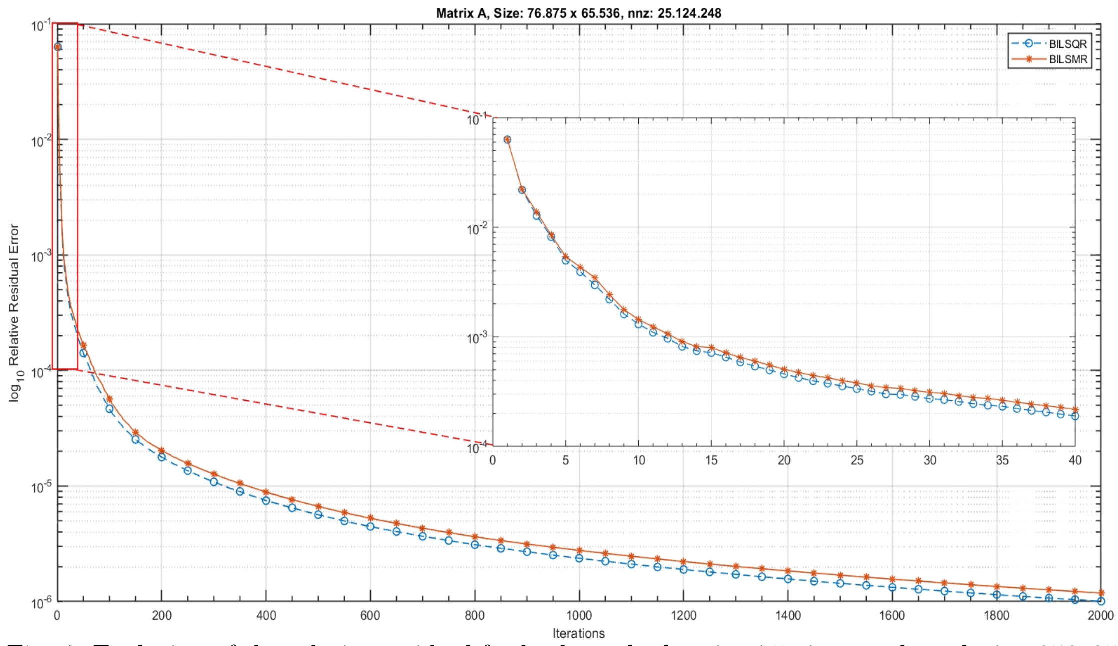


Fig. 4. Evolution of the relative residual for both methods using 75 views and resolution 256x256.

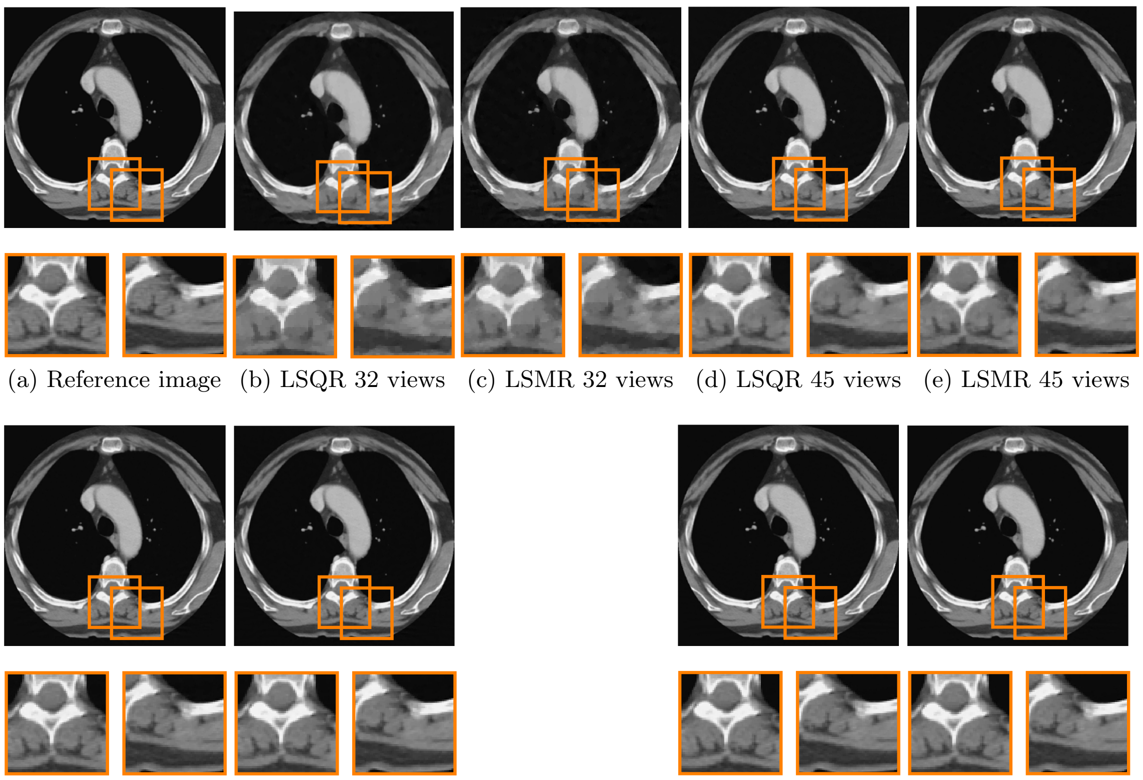
III.B.2. Image Quality

From the previous results, it was determined that every least-squares method tested provided similar image quality for the same reconstructions. Nevertheless, the results for these particular implementations will be analyzed to conclude which method is the best choice. Table VII presents the metrics results for both of the methods, using a different number of views. These results correspond to the images shown in Figure 5. The reference image is a thorax CT image selected from the dataset DeepLesion, focused on the Mediastinum area. Even when a reduced number of views is used, the reconstructions obtained have decent quality, but the Mean Square Error varies two orders of magnitude from the worse to the best reconstruction, which is significant. In every case, the LSMR images have slightly worse quality, as shown in Table VII. It can be observed that the LSQR method obtains better PSNR and SSIM value in every case. Although this is not easily perceived by the human eye as shown in the images of Figure 5, where the reconstructed images are displayed. The images are displayed focusing on the Hounsfield Units (HU) window specified by the dataset authors for this particular image, which goes from -175 to 275 HU. Looking to the zoomed-in regions of the images it can be observed how in the reconstructions from a reduced number of views (32) some areas are blurred and the internal structures of the original images are not well-preserved. However, the results obtained using 60 or 75 views are very similar, and no loss can be seen.

In order to have a better idea of the general error of the reconstructions, the absolute error images (the difference between the reconstructed image and the reference image, obtained by subtracting them) are shown in Figure 6. For each image, it is also shown the minimum and maximum value of the error image, as well as the mean and the standard deviation. It is worth mentioning that for the simulation of the data and resolution of the problem, the images are always represented taking the linear attenuation coefficients as gray values, instead of the Hounsfield

TABLE VII
Reconstructed Images Quality (256x256 pixels).

	LSQR				LSMR			
	32	45	60	75	32	45	60	75
PSNR	41.13	46.59	53.06	57.96	40.31	45.93	52.38	55.82
SSIM	0.9728	0.9891	0.9971	0.9990	0.9656	0.9874	0.9966	0.9983
MSE	1.28e-04	3.74e-05	1.32e-05	2.70e-06	1.43e-04	4.35e-05	1.38e-05	4.46e-06


(a) Reference image (b) LSQR 32 views (c) LSMR 32 views (d) LSQR 45 views (e) LSMR 45 views

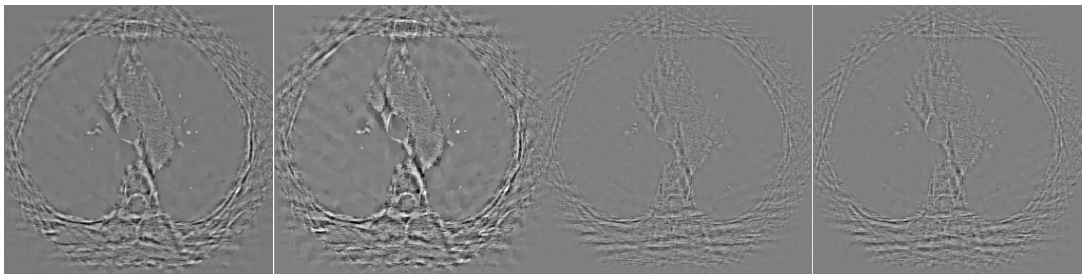
(f) LSQR 60 views (g) LSMR 60 views (h) LSQR 75 views (i) LSMR 75 views

Fig. 5. Thorax CT image reconstructions. HU Window: [-175, 275].

Units. The attenuation coefficients are expressed relative to that of water, that is set to 1. For this particular image, the grey values go from 0.8250 (-175 HU) to 1.2750 (275 HU). In the Figure, it can be observed how the error is much more notable when the number of views is low, and how it also affects the internal structures. When 75 views are used, the error is much lower so it is harder to appreciate. These images are high-quality, as a higher PSNR is obtained and the SSIM is very close to 1. This is due to the system matrix having complete rank, so the solution is more exact. Comparing the LSQR and LSMR results, it can be seen how in every case the LSMR images contain more error, which was not easily seen in Figure 5.

IV. CONCLUSION

In this work, a study of three algebraic methods has been performed in order to evaluate their performance when reconstructing CT images using few views. The LSMB method was a promising proposal since it was seen how the number of iterations was greatly reduced with respect to the iterations needed by LSQR and LSMR. Although the cost is almost double, the iterations were reduced to less than half, so the overall time would be reduced. For all of this, it can be concluded

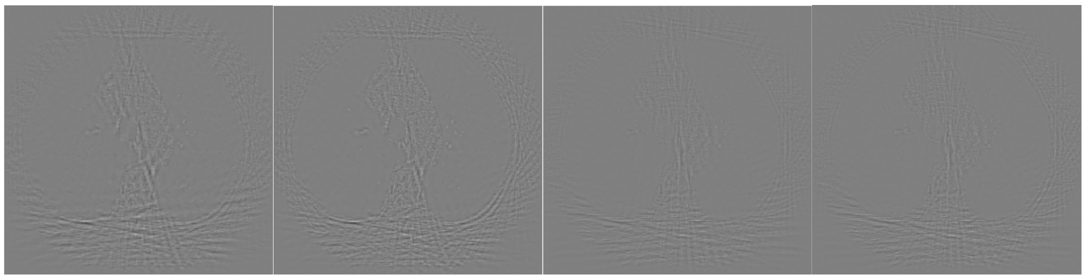


(a) LSQR 32 views Min=-0.082, Max=0.089, Mean=3.82e-07, Std=0.0113

(b) LSMR 32 views Min=-0.084, Max=0.095, Mean=4.51e-07, Std=0.0120

(c) LSQR 45 views Min=-0.059, Max=0.044, Mean=1.93e-08, Std=0.0061

(d) LSMR 45 views Min=-0.063, Max=0.049, Mean= 6.91e-08, Std=0.0066



(e) LSQR 60 views Min=-0.034, Max=0.043, Mean=2.80e-09, Std=0.0036

(f) LSMR 60 views Min=-0.040, Max=0.043, Mean=6.79e-08, Std=0.0037

(g) LSQR 75 views Min=-0.013, Max=0.015, Mean= 3.87e-08, Std=0.0016

(h) LSMR 75 views Min=-0.016, Max=0.023, Mean=1.62e-09, Std=0.0021

Fig. 6. Reconstruction error images.

that the LSMB method is an interesting approach to solving the least-squares problem in far fewer iterations than LSMR or LSQR due to its robust stopping criteria. However, for this problem of projections reduction, it loses its best feature, since it needs to be combined with other techniques that have influence over the convergence rate and thus the total iterations. When the number of projections is very low and the system matrix that models the CT scanner has a low rank, it is necessary to combine the resolution method with the STF and FISTA to boost the convergence rate. Compared to the other lest-squares methods, the number of iterations is very similar, and the quality of the images is not improved.

In this work, we have also conducted a study of the advantages of using block algebraic algorithms to reconstruct CT images when using few views and thus having rank-deficient matrices. Two methods have been compared: the Block LSQR, and the Block LSMR. We have developed a parallel implementation for both of them, using BLAS and LAPACK routines through MKL.

It has been determined that the time efficiency of the algorithms is very similar. The time per iteration using up to 8 cores is slightly higher for LSMR but not significantly. The parallel efficiency is lower than the number of cores in every case, but the code is still scalable. A higher efficiency is obtained when the number of cores used is equal to the right-hand sides to solve. On the other hand, it has been verified that the number of iterations the programs need to converge to the desired tolerance is lower when the LSQR method is used, so the total time of the reconstructions will be lower.

Finally, it was proved that the quality of the reconstructions is very similar for both methods. For resolution 256×256 pixels, the reconstructions are acceptable using a reduced number of views. When the matrix has complete rank, with 75 views, the reconstructions are high-quality. To reach full rank with resolution 512×512 the number of projections needed would be 260. However, with 180 projections the quality reached was very high combining the methods with the regularization techniques, with a SSIM above 0.99. Even with 90 projections, the SSIM was above 0.9. For all of this, it can be concluded that the methods we propose for the reconstruction of CT images using few projections are high-quality.

Regarding the radiation dose reduction, preliminary studies using Montecarlo simulations for the CT acquisitions show that the dose decreases linearly with the number of projections used. However, this may not be the case in a real scanner. Due to the physical effects caused by starting and stopping the X-ray source, it may be possible that the projections acquired by a sparse-view scanner suffer from information loss or new artifacts. Abbas et al. [23] performed a study of the effects that using different sparse-sampling schemes can have on image quality. Different schemes have different issues to solve, but they all obtain image qualities comparable to full-dose reconstructions with a 75% dose reduction (1/4 of the total projections). With the interrupted-beam strategy used in the scanner prototype proposed by Muckley et al. [24, 25], it was verified that the sinograms could suffer from elevated quantum noise and a *penumbra* effect. However, they adapted a reconstruction method to minimize these effects [17] and were able to reconstruct high-quality images with a 75% dose reduction compared to the full-dose images. They also were able to verify that when compared to the images acquired with tube current reduction, the error rate is similar, but it is more diffused over the images and not concentrated in the interior of the object, which is better. Overall, the current studies indicate that these strategies to reduce radiation dose on CTs are promising.

ACKNOWLEDGMENTS

This research has been supported by "Universitat Politècnica de València", and is part of the TED2021-131091B-I00 project, funded by MCIN/AEI/10.13039/501100011033 and by the "European Union NextGenerationEU/PRTR".

REFERENCES

[1] X. Tang, J. Hsieh, R. A. Nilsen, S. Dutta, D. Samsonov, and A. Hagiwara, "A three-dimensional-weighted cone beam filtered backprojection (CB-FBP) algorithm for image reconstruction in volumetric CT—helical scanning," *Physics in Medicine & Biology*, **51**, *4*, 855 (2006).

[2] G. H. Golub and J. M. Ortega, *Scientific Computing: An Introduction with Parallel Computing* (1993).

[3] M. Chillarón, G. Quintana-Ortí, V. Vidal, and G. Verdú, "Computed tomography medical image reconstruction on affordable equipment by using Out-Of-Core techniques," *Computer Methods and Programs in Biomedicine*, **193**, 105488 (2020); https://doi.org/10.1016/j.cmpb.2020.105488., URL https://www.sciencedirect.com/science/article/pii/S0169260719316190.

[4] G. Quintana-Ortí, M. Chillarón, V. Vidal, and G. Verdú, "High-performance reconstruction of CT medical images by using out-of-core methods in GPU," *Computer Methods and Programs in Biomedicine*, **218**, 106725 (2022).

[5] M. J. Rodríguez-Alvarez, F. Sanchez, A. Soriano, L. Moliner, S. Sanchez, and J. M. Benlloch, "QR-factorization Algorithm for Computed Tomography (CT): Comparison with FDK and Conjugate Gradient (CG) Algorithms," *IEEE Transactions on Radiation and Plasma Medical Sciences*, **2**, *5*, 459 (2018).

[6] A. H. Andersen, "Algebraic reconstruction in CT from limited views," *IEEE Transactions on Medical Imaging*, **8**, *1*, 50 (1989).

[7] A. H. Andersen and A. C. Kak, "Simultaneous algebraic reconstruction technique (SART): a superior implementation of the ART algorithm," *Ultrasonic Imaging*, **6**, *1*, 81 (1984).

[8] W. Yu and L. Zeng, “A Novel Weighted Total Difference Based Image Reconstruction Algorithm for Few-View Computed Tomography,” *PLoS ONE*, **9**, *10* (2014).

[9] C. C. Paige and M. A. Saunders, “LSQR: An algorithm for sparse linear equations and sparse least squares,” *ACM Transactions on Mathematical Software*, **8**, *1*, 43 (1982).

[10] E. Parcero, L. Flores, M. Sánchez, V. Vidal, and G. Verdú, “Impact of view reduction in CT on radiation dose for patients,” *Radiation Physics and Chemistry*, **137**, 173 (2017).

[11] M. Chillarón, V. Vidal, and G. Verdú, “Evaluation of image filters for their integration with LSQR computerized tomography reconstruction method,” *PLOS ONE*, **15**, *3*, 1 (2020); 10.1371/journal.pone.0229113., URL https://doi.org/10.1371/journal.pone.0229113.

[12] D. C.-L. Fong and M. Saunders, “LSMR: An iterative algorithm for sparse least-squares problems,” *SIAM Journal on Scientific Computing*, **33**, *5*, 2950 (2011).

[13] E. Hallman and M. Gu, “LSMB: Minimizing the backward error for least-squares problems,” *SIAM Journal on Matrix Analysis and Applications*, **39**, *3*, 1295 (2018).

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

[15] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM Journal on Imaging Sciences*, **2**, *1*, 183 (2009).

[16] K. Yan, X. Wang, L. Lu, and R. M. Summers, “DeepLesion: automated mining of large-scale lesion annotations and universal lesion detection with deep learning,” *Journal of Medical Imaging*, **5**, *3*, 036501 (2018).

[17] M. J. Muckley, B. Chen, T. Vahle, T. O’Donnell, F. Knoll, A. D. Sodickson, D. K. Sodickson, and R. Otazo, “Image reconstruction for interrupted-beam x-ray CT on diagnostic clinical scanners,” *Physics in Medicine & Biology*, **64**, *15*, 155007 (2019).

[18] F. Toutounian and S. Karimi, “Global least squares method (Gl-LSQR) for solving general linear systems with several right-hand sides,” *Applied Mathematics and Computation*, **178**, *2*, 452 (2006).

[19] F. Toutounian and M. Mojarrab, "The block LSMR algorithm for solving linear systems with multiple right-hand sides," *Iranian Journal of Numerical Analysis and Optimization*, **5**, *2*, 11 (2015).

[20] M. Chillarón, V. Vidal, D. Segrelles, I. Blanquer, and G. Verdú, "Combining grid computing and Docker containers for the study and parametrization of CT image reconstruction methods," *Procedia Computer Science*, **108**, 1195 (2017).

[21] A. Hore and D. Ziou, "Image Quality Metrics: PSNR vs. SSIM," *2010 20th International Conference on Pattern Recognition*, 2366–2369, IEEE (2010).

[22] P. Joseph, "An improved algorithm for reprojecting rays through pixel images," *IEEE Transactions on Medical Imaging*, **1**, *3*, 192 (1982).

[23] S. Abbas, T. Lee, S. Shin, R. Lee, and S. Cho, "Effects of sparse sampling schemes on image quality in low-dose CT," *Medical Physics*, **40**, *11*, 111915 (2013); https://doi.org/10.1118/1.4825096., URL https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.4825096.

[24] M. Muckley, B. Chen, T. O'Donnell, M. Berner, T. Allmendinger, K. Stierstorfer, T. Flohr, B. Schmidt, A. Sodickson, D. Sodickson et al., "Reconstruction of reduced-dose sparse CT data acquired with an interrupted-beam prototype on a clinical scanner," *The Fifth international conference on image formation in X-ray computed tomography, Salt Lake City*, 56–59 (2018).

[25] B. Chen, M. Muckley, A. Sodickson, T. O'Donnell, M. Berner, T. Allmendinger, K. Stierstorfer, T. Flohr, B. Schmidt, D. Sodickson et al., "First multislit collimator prototype for sparse CT: design, manufacturing and initial validation," *The Fifth international conference on image formation in x-ray computed tomography, Salt Lake City*, 52–55 (2018).