Parallel CT reconstruction for multiple slices studies with SuiteSparseQR factorization package

Mónica Chillarón[1][0000−0002−7611−8908], Vicente Vidal[1][0000−0002−2384−7015], and Gumersindo Verdú[2][0000−0001−5098−080X]

[1] Departamento de Sistemas Informáticos y Computación (DSIC)
Universitat Politècnica de València, València, Spain
mnichipr@inf.upv.es, vvidal@dsic.upv.es
[2] Instituto de Seguridad Industrial, Radiofísica y Medioambiental (ISIRYM)
Universitat Politècnica de València, València, Spain
gverdu@iqn.upv.es

Abstract. Algebraic factorization methods applied to the discipline of Computerized Tomography (CT) Medical Imaging Reconstruction involve a high computational cost. Since these techniques are significantly slower than the traditional analytical ones and time is critical in this field, we need to employ parallel implementations in order to exploit the machine resources and obtain efficient reconstructions.
In this paper, we analyze the performance of the sparse QR decomposition implemented on SuiteSparseQR factorization package applied to the CT reconstruction problem. We explore both the parallelism provided by BLAS threads and the use of the Householder reflections to reconstruct multiple slices at once efficiently. Combining both strategies, we can boost the performance of the reconstructions and implement a reliable and competitive method that gets high-quality CT images.

Keywords: CT · Medical Imaging · Reconstruction · Matrix factorization · QR · Few projections· Parallel QR · SuiteSparseQR

1 Introduction and Background

In recent years, medical tests such as Magnetic Resonance Imaging (MRI) [1] have gained prominence in clinical practice. MRIs are not harmful to the patient, since the image is produced from the application of magnetic fields on a body. In contrast, Computerized Tomographies (CT) [2] project X-rays, which induce a dose of radiation that can be harmful to the patient. However, despite being harmful, CT scans are still necessary.

On the one hand, they obtain better images than the MRI for certain types of objects of interest (bones and tumors) while the magnetic resonance is mostly applied to soft tissues since it achieves greater contrast between different tissues. On the other hand, not all people are suitable for both tests. MRI is not recommended for patients who have a pacemaker or metal implants in their body, while

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_12

they can undergo a CT scan. But CT is contraindicated for pregnant patients, infants and children, due to the dose of radiation induced with the test.

For all the above, although the current perception is that the CTs are losing ground to the MRI, it is not entirely true, so we believe that it is necessary to continue improving CT scanners and also the techniques of image reconstruction they use.

In our previous works [3–7], we have studied the option of working with algebraic methods instead of the traditional analytical methods to reconstruct CT images. In this way, we can solve the problem mathematically by either iterative or direct algebraic algorithms. While other works focus on reducing radiation by applying X-rays with lower voltage [8], we focus on taking fewer shots. With this approach, we can work with a low number of projections and still get high-quality images. This means that we could reduce the radiation dose to which the patient is exposed, which is our main objective.

Using iterative methods, we can reduce the number of views or projections that we use to a really small number if we make a good selection of the projection angles [9, 10, 6]. However, with the direct methods [11, 7], we have reached the conclusion that it is necessary that the matrix of the CT system has full rank, which will determine the number of projections required according to the image resolution that you want to achieve. Regardless, both approaches need a lower number of X-ray projections than other methods.

However, the types of methods we use require more computational resources. In addition, the time needed to reconstruct the images is much higher than with the analytical methods (minutes or hours versus milliseconds). Therefore, if we want the algebraic approach to be employed, we have to reduce the reconstruction time to the maximum. For this we can use High-Performance Computing (HPC) techniques, exploiting the hardware resources of the machine through parallel implementations of the algorithms.

In this paper, we focus on the analysis of the performance of the QR [12] factorization employed to reconstruct CT images. To do this, we make use of the SuiteSparseQR factorization package [13], analyzing the effect of using a different number of BLAS threads in operations. In addition, we compare the time efficiency when we reconstruct a single slice, or when we have a volume formed by multiple slices, which fits best a real situation.

In Section 2.1, we will describe the CT image reconstruction problem. We also make a brief description of the scanner we simulate and the dataset DeepLesion, used to take as reference. In Sections 2.2 and 2.3 we explain how to perform the CT reconstructions using the explicit QR factorization or the Householder form. The results of the study are discussed in Section 3, analyzing on the one hand the performance of the QR factorization using a different number of threads (Sect 3.1) as well as the performance of the reconstruction step (Sect 3.2). Additionally, in Section 3.3 we show the obtained images measuring their quality. Finally, in Section 4 we summarize the work done and discuss the future lines of work.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_12

2 Materials and Methods

2.1 Algebraic CT image reconstruction

When dealing with the reconstruction of CT images in an algebraic way, it is necessary to model the associated problem. Initially, we only have the data obtained through the scanner using X-rays. Therefore, it will be necessary to transform this data into an image that represents the projected object or body part.

We pose the problem as a system of linear equations as proposed in equation (1). Here, g is the data acquired by the scanner. It is usually called projections vector or sinogram for fanbeam CTs. As we see in (2), g is a vector of size M, which depends on the physical parameters of the scanner. We calculate M as the product of the number of detectors of the CT and the number of projections taken. It is worth mentioning than one projection (one X-ray shot) obtains as many data as detectors we have.

The system matrix A is defined in equation (4). This is a sparse weight matrix that represents the influence a of each ray beam traced (i) on each image pixel (j). As we can see, the number of rows of A is M, and the number of columns is N, which is the resolution in pixels of the reconstructed image we want to get.

In our case, both the matrix and the projections vector is simulated. We calculate both using the forward projection ray-tracing algorithm proposed by Joseph [14], simulating a CT scanner with 1025 detectors and taking equiangular projections around the 360 degrees of rotation. The images we projected are a selection of the DeepLesion dataset [15], which contains thousands of CT images of numerous patients for the study of different types of lesions.

Finally, u is the solution of our equations system. It is the reconstructed image. If we store in vector form, as we see in (3), its size is the total number of pixels on the image. For instance, for a final image of resolution 256×256 pixels, u is a vector of size 1×256^2. It is very easy to go from vector form to image form and vice versa.

$$A * u = g \quad (1) \qquad u = [u_1, u_2, ..., u_N]^T \in \mathbb{R}^N \quad (3)$$

$$g = [g_1, g_2, ..., g_M]^T \in \mathbb{R}^M \quad (2) \qquad A = a_{i,j} \in \mathbb{R}^{MxN} \quad (4)$$

2.2 QR Factorization applied to the CT problem

Ideally, the previously modeled problem could be solved very simply by obtaining the inverse of the system matrix as shown in the equation (5), provided it had full rank and the matrix was square. But in our application, this matrix dimensions can be really large for the highest resolutions and rectangular (more rows than columns). It is not feasible to explicitly compute the inverse since it requires a high computational cost and really advanced hardware. Besides, it is a highly unstable computation, so the errors could spoil the resulting image.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_12

For this reason, we need to solve the problem with an iterative method as we did in [3, 6] or apply a factorization to the matrix so we can solve it directly. The factorization we propose is the QR with pivoting [16], as described in (6). Here, Q is an orthonormal matrix (its columns are orthogonal unit vectors so $Q^tQ = I$). R is an upper triangular matrix and P is the permutation matrix used to reduce the filling.

With this decomposition done, we can emulate the inverse of the matrix A as shown in (7), or the pseudoinverse if the matrix is non-invertible [17, 18]. Now we can solve the problem as (8). And since the factorization can be performed only once and stored for future use, every time we need to reconstruct we will only have to perform a matrix-matrix product, a permutation and solving one upper triangular equations system. This means faster reconstructions.

In addition, here g can be a matrix $\mathbb{R}^{MxS}$ with S columns (the number of slices we want to reconstruct), so we can get multiple images within the same operation, which also reduces the time per slice.

$$u = A^{-1} * g \tag{5}$$

$$A * P = Q * R \tag{6}$$

$$A^{-1} = PR^{-1}Q^T \tag{7}$$

$$u = P * (R^{-1}(Q^T * g)) \tag{8}$$

2.3 Q-less factorization using Householder reflections

As we mentioned, our matrices can get relatively big depending on the desired image resolution, so it is possible the computational resources needed to compute the decomposition are extensive. Even if the system matrix A is sparse, Q can be dense. As a way to spare main memory resources, we could decide to not calculate the Q matrix explicitly. Instead, we could perform the factorization using Householder reflections, and store only the set of Householder vectors, which describe transforms to be applied as shown in (9). Here H_i are the successive reflection matrices to apply to our right-hand side g. Apart from main memory, this technique will also reduce the computation time.

$$Q = H_1 H_2 \cdots H_{N-2} H_{N-1} \tag{9}$$

2.4 SuiteSparseQR factorization package

SuiteSparseQR [13] is an implementation of the multifrontal sparse QR factorization method. It uses both BLAS and Intel's Threading Building Blocks, a shared-memory programming model for modern multicore architectures to exploit parallelism. The package is written in C++ with user interfaces for MATLAB, C, and C++. It works for real and complex sparse matrices.

In our case, we are using the MATLAB interface, making use of BLAS parallelism and working with real sparse matrices.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_12

3 Results and Discussion

In order to test the performance of the method we run both the matrix factorization and the reconstructions in a server with four Intel Xeon E5-4620 8c/16T processors (8 cores/processor, 32 cores/node) and 256GB DDR3 RAM memory (ratio 8GB/core).

We use a matrix corresponding to resolution 256x256 pixels and 90 projections. Thus, the size of the matrix is 90200x65536, with 40871478 non-zero elements and 0.0069 density. In the next subsections we show the experimental time results when using a different number of BLAS threads for the two alternatives of the factorization method, using one physical processor per thread.

3.1 Factorization step

In the fist step we need to compute the factorization shown in equation (6). As explained in Section 2.2, the factorization of the matrix can be done once before the reconstruction of the images. Therefore, the computational speed is not going to be that important in this case. Even so, it is desirable to reduce it to the maximum, both to save computing time and to avoid possible system failures that can abort our process and spoil hours of work.

As we can see in Table 1, it is much faster to store the factorization in Householder form than to form the Q matrix explicitly. Regardless of the number of BLAS threads employed, it is around 3 times faster, which is a significant difference.

In Figures 1 and 2 we can see the Speedup and Efficiency of both of the methods. We compute the Speedup for p processors as $S_p = T_1/T_p$, being T_1 the time with 1 processor and T_p the time with p processors. The Efficiency is $E_p = S_p/p$. In a perfect parallel algorithm, the Speedup is equal to the number of processors and the Efficiency is 1, which means we are taking advantage of the 100% of the resources.

However, we can observe that we get lower results. The Householder factorization has slightly better performance except when using 32 processors, with a Speedup of 8 versus the 10.2 of the explicit Q factorization. In Figure 2 we can see that with more than 4 processors we are using less than a 50% of the computational resources. Since we are working with sparse matrices with a low density it is usual to get lower efficiency that with dense matrices.

Table 1: Factorization time

	Factorization Method		
BLAS threads	Explicit Q	Householder	Improvement Factor
	Time (secs.)		
1	172784	57628	3.00
2	119407	31955	3.74
4	72186	21105	3.42
8	56325	18481	3.05
16	36451	10422	3.50
32	16805	7191	2.34

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_12

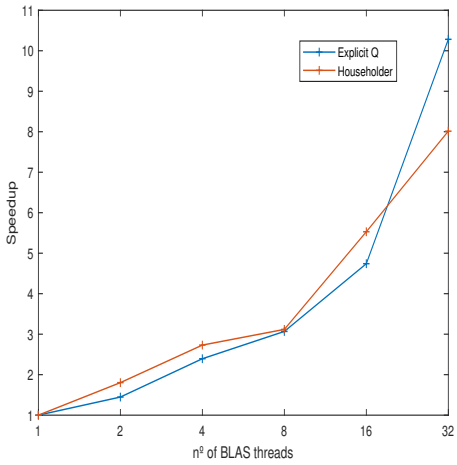

Fig. 1: Factorization Speedup

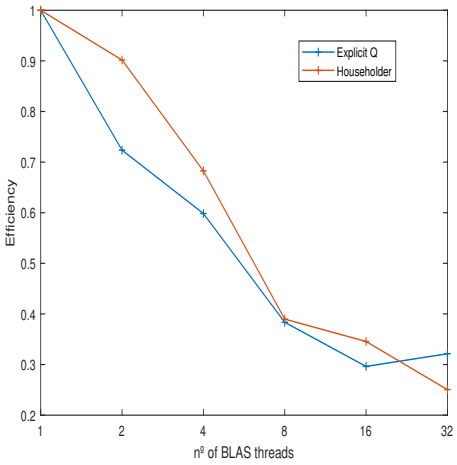

Fig. 2: Factorization Efficiency

3.2 Reconstruction step

We also need to verify which method is better in the reconstruction phase and if we get good performance when using more resources. In Table 2 we observe the results for reconstructing just 1 slice (1 right-hand side vector) or 128 slices, as per equation (8). As we can see, when we want to reconstruct only 1 slice, the Explicit Q factorization gets lower computational times. Besides, they improve slightly when using more processors, up to 16. With the Householder form, we only get better performance with up to 8 processors. However, the performance here is worse than in the factorization step, as we can see in Figures 3 and 4. The Speedup is very low regardless of the number of threads used, and the highest efficiency we get is only 0.5 using 2 threads.

When we are dealing with multiple right-hand sides, in this case 128, the performance is different. We can see in the table that in this case, it is faster to perform the reconstruction using the Householder reflections. We get the reconstructions around twice as fast (2.8 times faster for 16 threads). On the other hand, the Speedup and Efficiency for each of the methods is not better than in the previous case, as we can see in Figures 5 and 6. We get very low Efficiency, wasting resources.

Table 2: Reconstruction time

	1 Slice		128 Slices	
BLAS threads	Explicit Q	Householder	Explicit Q	Householder
	Time (secs.)			
1	239	281	2432	1392
2	226	272	2370	1381
4	221	256	2351	1370
8	210	232	2638	1296
16	177	253	2539	910
32	222	269	2607	1111

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_12

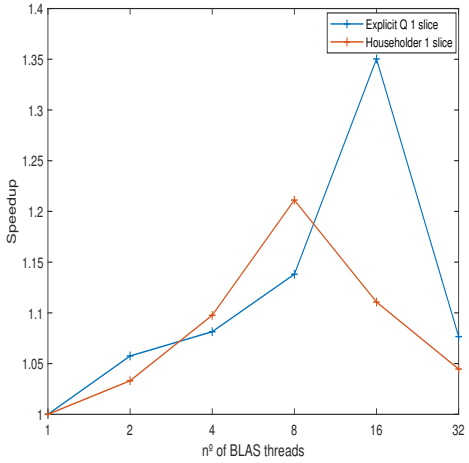

Fig. 3: 1 slice rec. Speedup

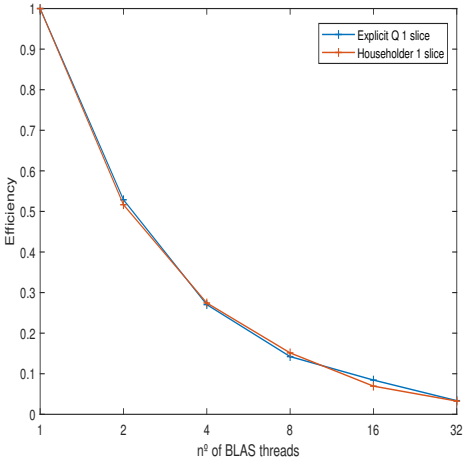

Fig. 4: 1 slice rec. Efficiency

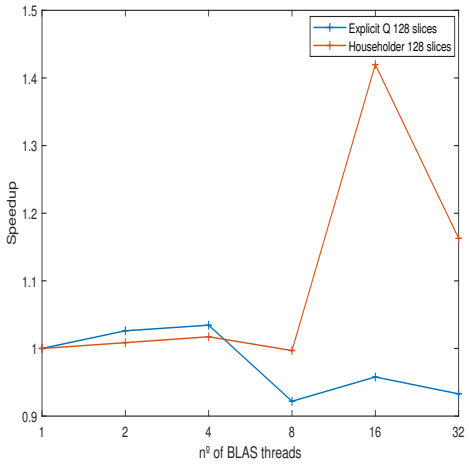

Fig. 5: 128 slices rec. Speedup

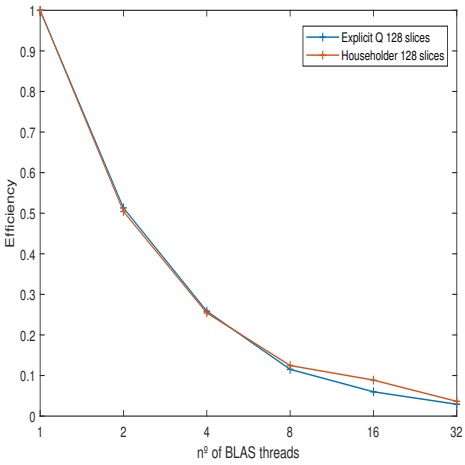

Fig. 6: 128 slices rec. Efficiency

3.3 Image Quality

In our preliminary work [7] we concluded that if we work with a full-rank system matrix, the images reconstructed by the QR factorization have really high quality. In order to verify this for the matrix we are studying here, we show the quality results, measuring with Mean Absolute Error (MAE), PSNR (Peak Signal-to-Noise Ratio) [19]. We also measure the SSIM (Structural Similarity Index). Since we are reconstructing 128 slices, we show the minimum, maximum and average results of both reconstruction techniques.

In Figures 7 and 8 we can see the PSNR and MAE results respectively. We don't show the SSIM results, since all the images get a SSIM equal to 1. This means that all the reconstructions have the same structure as the reference

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_12

image (we are not losing significant internal structures). As we observe in the Figures, the reconstructions obtained through the Householder matrix have better quality, which reflects the better numerical stability of the factorization. It gets around 10 units of PSNR higher and lower error. But both of the methods get really high quality. Notice that we are getting MAEs of order 10^{-12}, which is almost insignificant.

In Figure 9 we show the reference CT image of an abdomen and the reconstructed images with both techniques. As we can see, the images are almost identical to the human eye, since we can not discern significant differences or information loss.

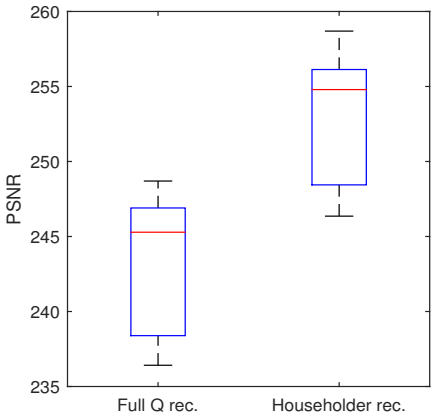

Fig. 7: PSNR results

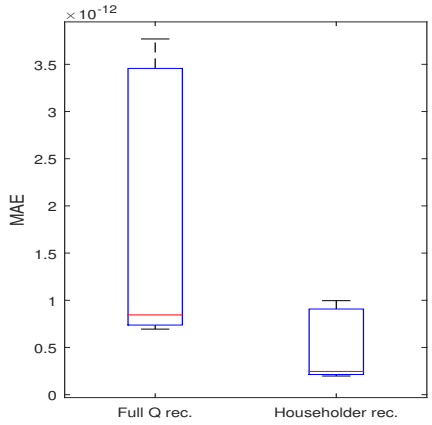

Fig. 8: MAE results

(a) Reference (b) Full Q rec. (c) Householder rec.

Fig. 9: Abdomen CT Reconstructions

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_12

4 Conclusion

In this work, we have conducted a study of the efficiency of applying a direct algebraic technique to the CT image reconstruction problem. We have compared two methods, the QR factorization forming the matrix Q explicitly, or using the matrix Q in the form of Householder reflections. To verify the performance of the methods we have used the SuiteSparseQR library, which includes a parallel implementation of these algorithms. We have used a server to get the experimental time performance, using up to 32 processors.

Regarding the quality obtained by both methods, we have determined that the Householder factorization is more numerically stable, so the images have less error. However, we speak of very small magnitudes that are not perceptible to the human eye.

On the other hand, we have verified that the time used to perform the factorization of the system matrix is much higher if we form the matrix Q in an explicit manner. This can lead to errors as we expose ourselves to system failures, power outages, etc. In addition, we have verified that the reconstructions are faster when we reconstruct several images simultaneously. We get less time per slice in general, and less time using the Householder form in particular. Since in reality we are going to reconstruct more than one slice at a time (usually a CT study has at least 100 slices), we determine that it is more efficient to use Householder reflections for our application. In this way we can reconstruct volumes with high quality in less than 30 minutes.

Finally, we have determined that the efficiency that is achieved using fine-grained parallelism in many-core servers is not good. We observe that we do not take advantage of all the allocated resources, obtaining very low Speedups. Still, the time used to solve 128 slices with 32 threads is slightly lower than using the 32 threads to solve each slice independently using coarse-grain parallelism.

At this point, it is still necessary to employ HPC computers since this implementation requires a high amount of main memory. That is the reason we are not able to compute the problem for higher image resolutions. As future work, we plan to work with out-of-core techniques that read the data stored in blocks from the hard drive when the particular block is needed for the computation, instead of having it always loaded in main memory. In this way, we could achieve to reconstruct bigger problems in workstations with lower amount of RAM memory and thus lower cost.

4.1 Acknowledgements

This research has been supported by "Universitat Politècnica de València", "Generalitat Valenciana" under PROMETEO/2018/035 co-financed by FEDER funds, as well as ACIF/2017/075 predoctoral grant, and the "Spanish Ministry of Economy and Competitiveness" under Grant TIN2015-66972-C5-4-R and TIAMHA co-financed by FEDER funds.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_12

References

1. Brown, R.W., Haacke, E.M., Cheng, Y.C.N., Thompson, M.R., Venkatesan, R.: Magnetic resonance imaging: physical principles and sequence design. John Wiley & Sons (2014)
2. Brooks, R., Chiro, G.D.: Principles of computer assisted tomography (CAT) in radiographic and radioisotopic imaging. Physics in Medicine and Biology **21**(5), 689–732 (1976)
3. Flores, L., Vidal, V., Verdú, G.: Iterative reconstruction from few-view projections. Procedia Computer Science **51**, 703–712 (2015)
4. Parcero, E., Flores, L., Sánchez, M., Vidal, V., Verdú, G.: Impact of view reduction in CT on radiation dose for patients. Radiation Physics and Chemistry **137**, 173–175 (2017)
5. Flores, L.A., Vidal, V., Mayo, P., Rodenas, F., Verdú, G.: Parallel CT image reconstruction based on GPUs. Radiation Physics and Chemistry **95**, 247–250 (2014)
6. Chillarón, M., Vidal, V., Segrelles, D., Blanquer, I., Verdú, G.: Combining grid computing and Docker containers for the study and parametrization of CT image reconstruction methods. Procedia Computer Science **108**, 1195–1204 (2017)
7. Chillarón, M., Vidal, V., Verdú, G., Arnal, J.: CT Medical Imaging Reconstruction Using Direct Algebraic Methods with Few Projections. In: Computational Science – ICCS 2018. pp. 334–346. Springer International Publishing, Cham (2018)
8. Padole, A., Ali Khawaja, R.D., Kalra, M.K., Singh, S.: CT radiation dose and iterative reconstruction techniques. American Journal of Roentgenology **204**(4), W384–W392 (2015)
9. Andersen, A.H., Kak, A.C.: Simultaneous algebraic reconstruction technique (SART): a superior implementation of the ART algorithm. Ultrasonic Imaging **6**(1), 81–94 (1984)
10. Yu, W., Zeng, L.: A Novel Weighted Total Difference Based Image Reconstruction Algorithm for Few-View Computed Tomography. PLoS ONE **9**(10) (2014)
11. Rodríguez-Alvarez, M.J., Sanchez, F., Soriano, A., Moliner, L., Sanchez, S., Benlloch, J.M.: 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), 459–469 (2018)
12. Golub, G.H., Van Loan, C.F.: Matrix Computations, vol. 3. Johns Hopkins University Press (2013)
13. Davis, T.A.: Algorithm 915, SuitesparseQR: Multifrontal Multithreaded Rank-revealing sparse QR factorization. ACM Trans. Math. Softw. **38**(1), 8:1–8:22 (2011)
14. Joseph, P.: An improved algorithm for reprojecting rays through pixel images. IEEE Transactions on Medical Imaging **1**(3), 192–196 (1982)
15. Yan, K., Wang, X., Lu, L., Summers, R.M.: DeepLesion: automated mining of large-scale lesion annotations and universal lesion detection with deep learning. Journal of Medical Imaging **5**(3), 036501 (2018)
16. Golub, G.H., Ortega, J.M.: Scientific Computing: An Introduction with Parallel Computing. Academic Press Professional, Inc. (1993)
17. Arridge, S., Betcke, M., Harhanen, L.: Iterated preconditioned LSQR method for inverse problems on unstructured grids. Inverse Problems **30**(7), 075009 (2014)
18. Hansen, P.C.: The L-curve and its use in the numerical treatment of inverse problems (1999)
19. Hore, A., Ziou, D.: Image Quality Metrics: PSNR vs. SSIM. In: 2010 20th International Conference on Pattern Recognition. pp. 2366–2369. IEEE (2010)

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_12