Mejoras en métodos directos e iterativos de reconstrucción de imagen TC

Mónica Chillarón
Dpto. Sistemas Informáticos y Computación.
Universitat Politècnica de València.

Reunión Anual Proyecto TED

Ayuda TED2021-131091B-I00, financiada por MCIN/AEI/10.13039/501100011033/ y por la Unión Europea NextGenerationEU/PRTR

Valencia, 28 de febrero de 2024

Utilización de nueva librería de reconstrucción: Astra Toolbox

ASTRA Toolbox es una herramienta valiosa para investigadores y desarrolladores que trabajan en el campo de la tomografía. La toolbox ofrece una amplia gama de funcionalidades y algoritmos, lo que la hace ideal para una variedad de aplicaciones.

- Funcionalidad:

  Toolbox de MATLAB y Python para primitivas GPU de alto rendimiento para tomografía 2D y 3D.
  Soporta geometrías de haz paralelo y abanico en 2D, y haz paralelo y cono en 3D.
  Posicionamiento flexible de fuente/detector.

- Algoritmos:

  Gran cantidad de algoritmos 2D y 3D disponibles, incluyendo FBP, SIRT, SART, CGLS. Proyecciones hacia adelante y hacia atrás aceleradas por GPU.
  Capacidad de construir nuevos algoritmos.

- Instalación:

  Disponible para Windows, Linux y macOS.
  Instalación binaria o mediante conda para Python.
  Código fuente disponible para compilación personalizada.

- Documentación:

  Amplia documentación en línea con tutoriales y ejemplos.
  Guía de instalación detallada.
  Referencia de API para MATLAB y Python.

- Soporte:

  Foro de la comunidad para preguntas y respuestas.
  Soporte por correo electrónico para usuarios con licencia.
  Ejemplos de código y casos de prueba.

- Aplicaciones:

  Tomografía computarizada (TC)
  Tomografía por emisión de positrones (PET)
  Tomografía de rayos X
  Microscopía de rayos X
  Radiografía industrial

- Recursos adicionales:

  Sitio web de ASTRA Toolbox: https://astra-toolbox.com/
  GitHub: https://github.com/astra-toolbox/astra-toolbox
  Documentación: https://astra-toolbox.com/docs/

Comprensión y manejo del dataset DICOM-CT-PD

- Dataset desarrollado por la clínica Mayo.
- Formato estandarizado para diferentes fabricantes.
- Proporcionan proyecciones y las correspondientes reconstrucciones con FBP.
- 99 exploraciones de la cabeza, 100 exploraciones de tórax y 100 exploraciones de abdomen.
- Adquisiciones a dosis completa.
- Simulaciones de baja dosis: 25 % de la dosis para cabeza y abdomen, 10 % de la dosis para tórax.

Se han desarrollado diferentes herramientas para el manejo del dataset:

- Descarga e indexación de todos sus archivos en servidor AURUS.
- Herramientas en Matlab para leer proyecciones y/o reconstrucciones.
- Conversión entre coeficientes de atenuación y Unidades Hounsfield.
- Adecuación del código **helix2fan** para convertir las proyecciones a formato axial con detectores rectos.
- Reconstrucción con librería ASTRA de dichas proyecciones.
- Guardado de datos para acceder desde códigos en C, Python, Keras, lectores de DICOM.

Las herramientas para la adecuación de los datos del dataset están en desarrollo contínuo, porque dependen mucho de la aplicación que se quiera desarrollar.
Lo importante es que ya se conoce en profundidad el dataset, toda la información que contiene y cómo tratarlo.

Aplicación de métodos iterativos a bloques en CPU y GPU

Reconstrucción de imagen TC con pocas vistas por bloques usando GPU: rendimiento y calidad.

M.Chillarón[1], V.Vidal[2], G.Verdú[1]

Implementación a bloques de los métodos LSQR y LSMR:

- Reconstrucción de múltiples cortes al mismo tiempo.
- Mayor eficiencia temporal.
- Implementación en CPU: BLAS, Intel's MKL.
- Implementación en GPU: cuBLAS, cuSPARSE, cuSOLVER.
- Estudios completos en varios minutos.

NUCLEAR SCIENCE AND ENGINEERING

DOI: https://doi.org/10.1080/00295639.2023.2199677

Check for updates

Few-View CT Image Reconstruction via Least-Squares Methods: Assessment and Optimization

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

[a]*Universitat Politècnica de València, Depto. de Sistemas Informáticos y Computación, Edificio 1F, Camí de Vera S/n, València, 46022, Spain*
[b]*Universitat Politècnica de València, Instituto de Seguridad Industrial, Radiofísica Y Medioambiental, València, 46022, Spain*
[c]*Universidad Jaime I, Depto. de Ingeniería y Ciencia de Computadores, Castellón, 12071, Spain*

Received November 29, 2022
Accepted for Publication April 1, 2023

Métodos LSQR y LSMR a bloques:

- Resuelven el sistema AX=G
- Explotan paralelismo operaciones matriciales
- Múltiples lados derechos simultáneos
- Pueden trabajar con pocas vistas
- Combinados con STF y FISTA

Algorithm 1 Block LSQR algorithm.

Input: $A_{MxN}, G_{MxS}, Xini_{NxS}, iter$

Output: X_{NxS}

$X = Xini$
$[U, B] = econQR(G - AX)$
$[V, A_1] = econQR(A^T U)$
$W_1 = V, \Phi = B, \rho = A_1^T$
for $i = 1, \ldots, iter$ **do**
  $W = AV - UA_1^T$
  $[U, B] = econQR(W)$
  $S = A^T U - VB^T$
  $[V, A_1] = econQR(S)$
  $[Q, R] = QR(\begin{bmatrix}\rho \\ B\end{bmatrix})$
  $\begin{bmatrix}a & b \\ c & d\end{bmatrix} = Q$
  $\theta = c^T A_1^T$
  $\rho = d^T A_1^T$
  $\phi = a^T \phi_1$
  $\phi_1 = b^T \phi_1$
  $P = W_1 R^{-1}$
  $X = X + P\phi$
  $W_1 = V - P\theta$
end for

Algorithm 2 Block LSMR algorithm.

Input: $A_{MxN}, G_{MxS}, Xini_{NxS}, iter$
Output: X_{NxS}

$X = Xini$
$[U, B] = econQR(G - AX)$
$[V, A_1] = econQR(A^T U)$
$B_1 = A_1 B$
$\sigma_0 = -B_1$
$\sigma_1 = [0]_{s \times s}$
$a, b_0, c, d_0 = [0]_{s \times s}$
$b, d = [I]_{s \times s}$
$P_1, P_0 = [0]_{n \times s}$
for $i = 1, \ldots, iter$ **do**
  $W = AV - UA_1^T$
  $[U, B] = econQR(W)$
  $A_2 = A_1 A_1^T + B^T B$
  $S = A^T U - VB^T$
  $AUX = V$
  $[V, A_1] = econQR(S)$
  $\beta = d_0 B_1^T$
  $\alpha = c\beta + dA_2$
  $\beta_1 = a\beta + bA_2$
  $\theta = b_0 B_1^T$
  $B_1 = A_1 B$
  $[Q, R] = QR(\begin{bmatrix}\alpha \\ B_1\end{bmatrix})$
  $d_0 = d$
  $b_0 = b$
  $\begin{bmatrix}a & b \\ c & d\end{bmatrix} = Q^T$
  $\sigma = -(R^{-T}((\theta^T \sigma_1) + (\beta_1 \sigma_0)))$
  $P = (AUX - (P_1\theta) - (P_0\beta_1))(R^{-1})$
  $P_1 = P_0$
  $P_0 = P$
  $X = X + P\sigma$
  $\sigma_1 = \sigma_0$
  $\sigma_0 = \sigma$
end for

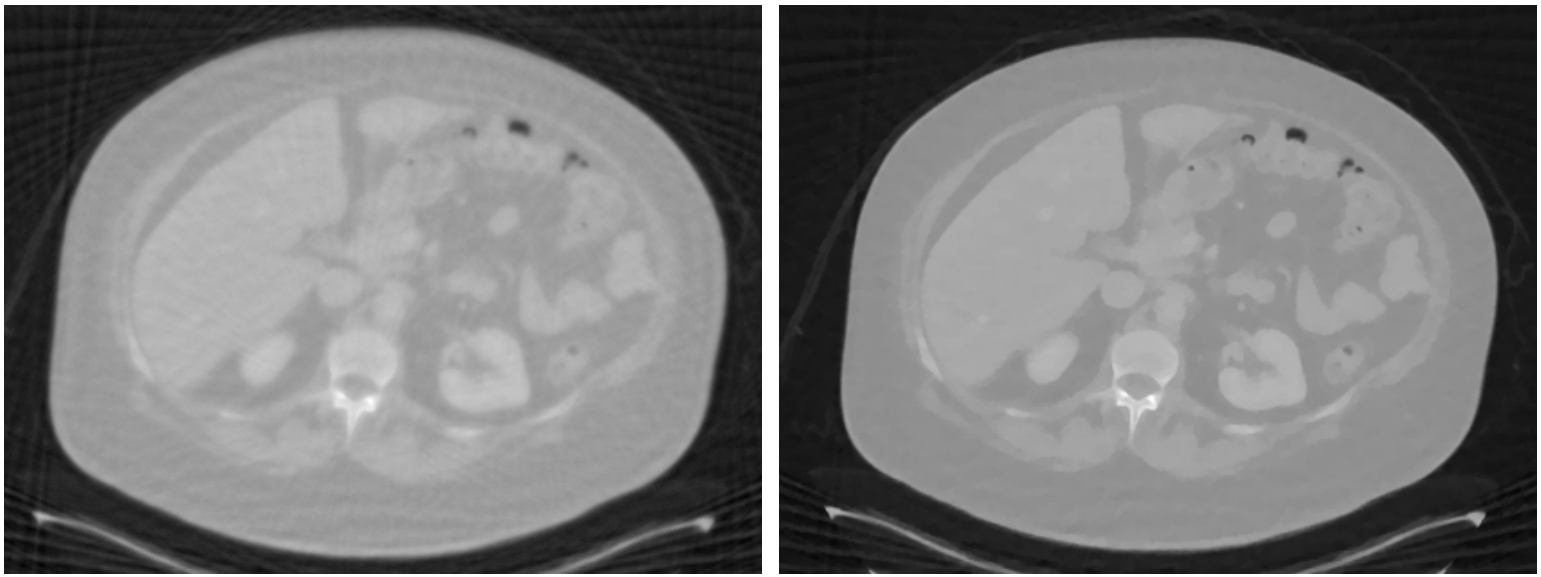
Operation on the Algorithm	Description	MKL Call	CUDA Call
$G - AX$ (Alg. 1, line 2) $A^T U$ (Alg. 1, line 3)	Computes the product of a sparse matrix and a dense matrix. $Y := \alpha AX + \beta Y$ where α and β are scalars, A is a sparse matrix, and X and Y are dense matrices.	mkl_sparse_d_mm	cusparseDcsrmm
$QR(Y)$ (Alg. 1, line 10)	Computes the QR factorization of a general m-by-n matrix Y.	LAPACKE_dgeqrf	cusolverDnDgeqrf
$econQR(Y)$ (Alg. 1, line 2)	Generates the real orthogonal matrix Q of the QR factorization formed by dgeqrf.	LAPACKE_dorgqr	cusolverDnDorgqr
UA_1^T (Alg. 1, line 6)	Computes a matrix-matrix product with general dense matrices.	cblas_dgemm	cublasDgemm
$P = W_1 R^{-1}$ (Alg. 1, line 16)	Solves a triangular linear system with multiple right-hand-sides.	LAPACKE_dtrtri cblas_dgemm	cublasDtrsm

Tabla: 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

Tabla: 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



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

Fig. : Reconstructions with LSMB and 90 projections.

Tabla: 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

Tabla: 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

Tabla: 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

Tabla: Time per iteration / total time in seconds for resolution 512x512 pixels, varying the number of projections and right-hand-sides.

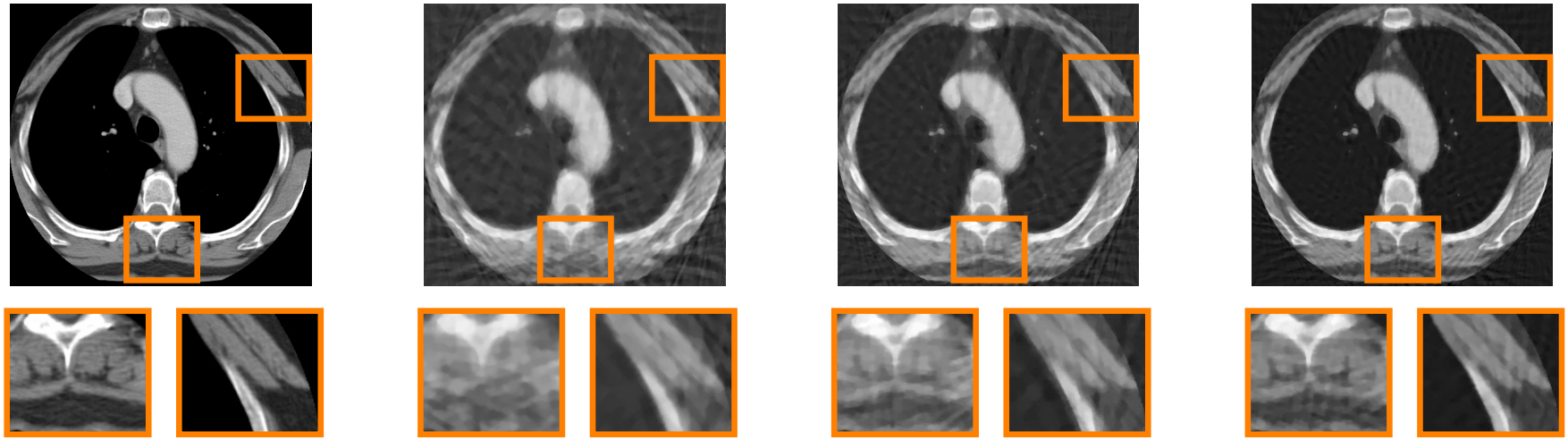
		CPU times			GPU times		
		1 slice	32 slices	64 slices	1 slice	32 slices	64 slices
Projections	32	0.144 50	1.383 623	2.777 927	0.004 1.4	0.166 75	0.551 184
	45	0.206 190	1.802 980	3.572 1354	0.004 3.7	0.178 97	0.567 215
	60	0.268 250	2.144 1354	4.256 1935	0.006 5.6	0.182 115	0.585 266
	75	0.331 274	2.606 2176	5.149 2747	0.007 5.8	0.188 157	0.594 317
	90	0.398 352	3.069 2874	6.084 3628	0.007 6.2	0.189 177	0.607 362
	120	0.415 337	3.311 2105	6.514 4142	0.008 6.5	0.211 207	0.618 393
	150	0.489 374	3.965 4605	7.922 6055	0.009 6.8	0.223 259	0.645 493
	180	0.512 302	4.251 5880	8.503 7611	0.012 7.1	0.227 314	0.670 600

- GPU obtiene SpeedUp de 35 para 32 proyecciones y un corte.
- Speedup de 50 cuando el número de proyecciones es mayor.
- Para 32 cortes, ejecuciones en GPU son de 8 a 18 veces más rápidas.
- Para 64 cortes, Speedup entre 5 a 12.
- Para 90 proyecciones y 64 cortes, 60 minutos en CPU y 6 en GPU.

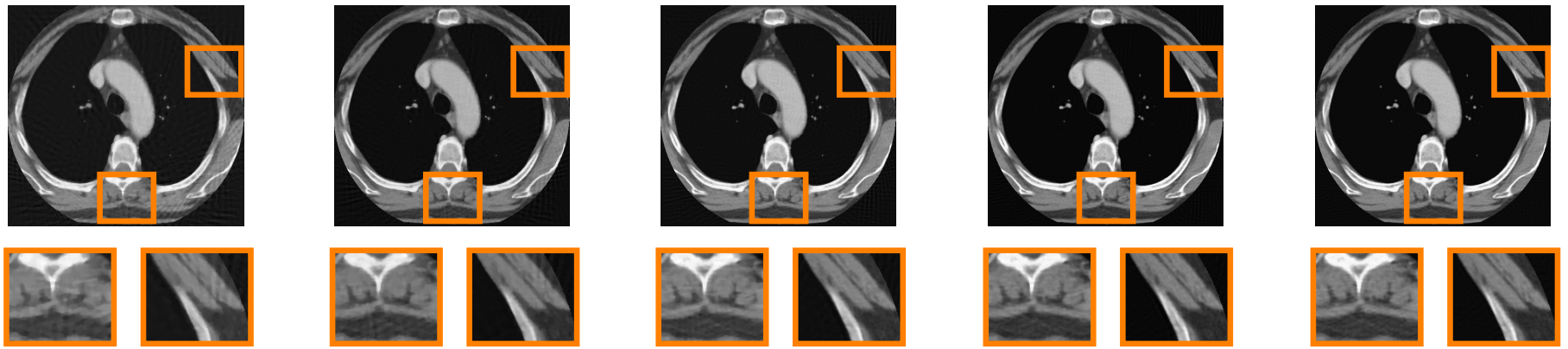
Tabla: Thorax Image Quality (512x512 pixels)

	Projections							
	32	45	60	75	90	120	150	180
PSNR	32.17	35.27	38.02	41.07	44.10	46.26	49.28	50.37
SSIM	0.866	0.906	0.937	0.960	0.977	0.983	0.990	0.992

- Reconstrucciones de poca calidad hasta 75 vistas.
- Artefactos desaparecen casi al completo con 120 vistas.
- Aun así no se obtiene una calidad estructural perfecta.
- Incluso empleando 180 vistas se reducen respecto a otros métodos (260 para algebraicos directos).



(a) Reference image (b) 32 views (c) 45 views (d) 60 views



(e) 75 views (f) 90 views (g) 120 views (h) 150 views (i) 180 views

- Métodos a bloques que aprovechan operaciones matriciales.
- Pueden trabajar con pocas proyecciones minimizando el ruido: mayor calidad que métodos analíticos.
- No necesitan rango completo en la matriz: mayor reducción de dosis que los métodos directos.
- Múltiples lados derechos.
- Versión en GPU: estudio de 64 cortes en 5 minutos.
- Empiezan a ser competitivos con métodos analíticos.

- Implementación más optimizada de ambos métodos.
- Mejorar los tiempos y el formato de entrada/salida.
- Probar a combinar con filtro Fuzzy para reducir artefactos.
- Probar con proyecciones reales.
- Comparar con las implementaciones de ASTRA.

Método QR out-of-core: precisión doble vs simple

Computer Methods and Programs in Biomedicine 193 (2020) 105488

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

Computed tomography medical image reconstruction on affordable equipment by using Out-Of-Core techniques

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

Computer Methods and Programs in Biomedicine 218 (2022) 106725

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

High-performance reconstruction of CT medical images by using out-of-core methods in GPU

Gregorio Quintana-Ortí[a,1], Mónica Chillarón[b,*], Vicente Vidal[c], Gumersindo Verdú[b,2]

M&C 2023 - The International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering · Niagara Falls, Ontario, Canada · August 13 – 17, 2023

CT Image Reconstruction Using the QR Method: Single vs Double Precision

M. Chillarón[1], G. Quintana-Ortí[2], V. Vidal[1] and G. Verdú[3]

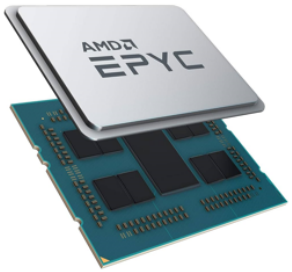
[1]Depto. de Sistemas Informáticos y Computación
Universitat Politècnica de València, València, 46022, Spain.

[2] Depto. de Ingeniería y Ciencia de Computadores
Universitat Jaume I, Castellón, 12071, Spain.

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

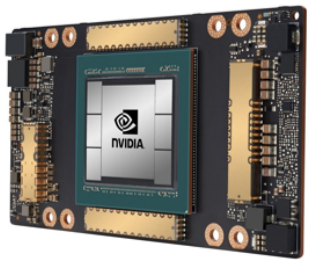
- Método QR out-of-core para reconstrucción de CT.
- Versiones CPU y GPU.
- Versión en simple precisión para acelerar las reconstrucciones.
- Comparativa tiempo y calidad para ambas precisiones.

Experimental Setup:



2 AMD EPYC 7282 processors (32 cores)

768 GiB of RAM DDR4



NVIDIA A100 GPU with 40 GiB of RAM

Samsung SSD 970 EVO 2TB

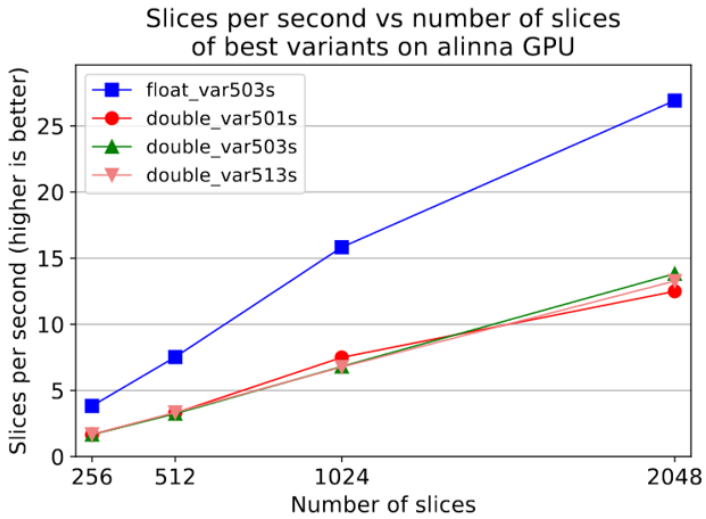
$$\text{Residual} = ||AX - B||_F / ||A||_F$$

Precision	Residual
Double	7.52e-12
Single	2.64e-06

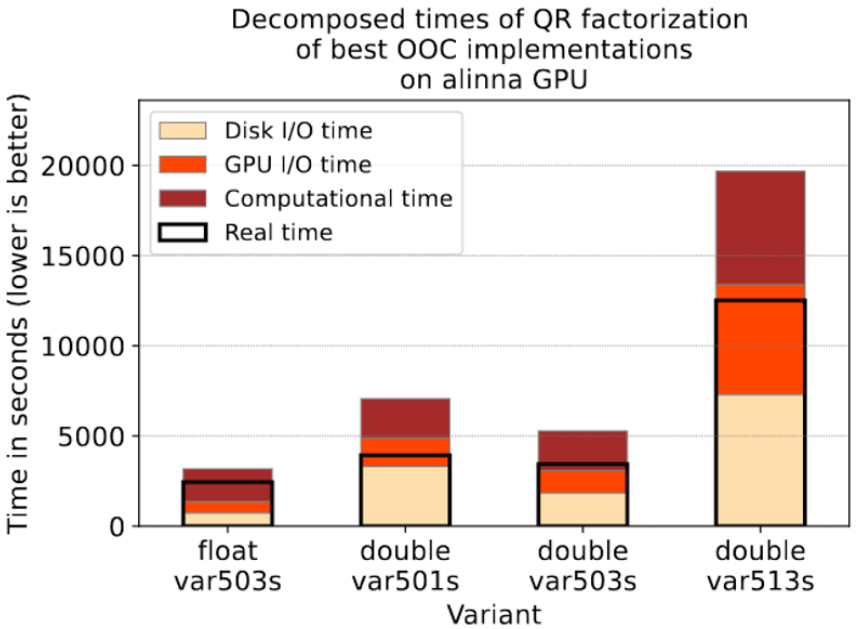
Residuals for the resolution.

Precision	QR calculation	Resolution AX = B
Double	5357,6	248,1
Single	3702,0	100,8

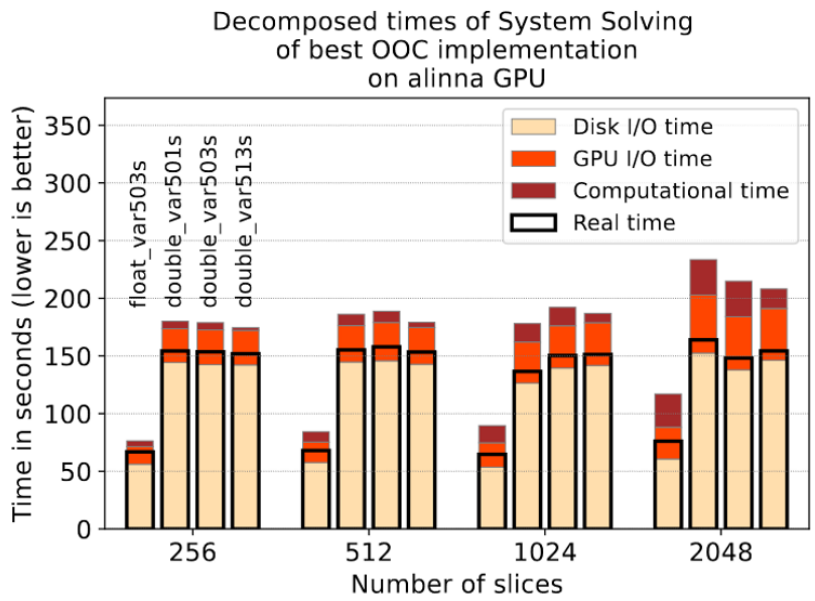
Computational time in seconds for the QR factorization and total time of the resolution of 2048 simultaneous slices.



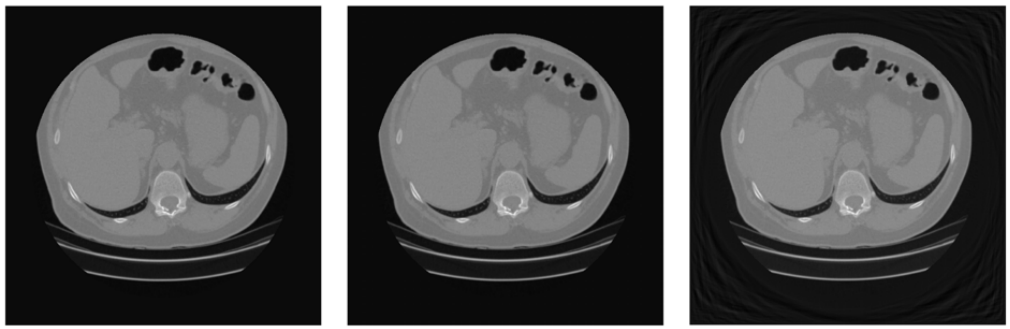
Slices per second versus number of slices being simultaneously computed for best variants on Alinna GPU.



Decomposed times of QR factorization for best variants on Alinna GPU.



Decomposed times of system solving for best variants on Alinna GPU.

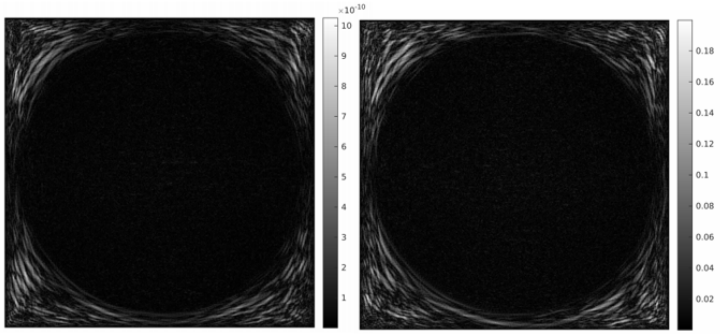


(a) Reference. (b) Double precision. (c) Single precision.

Reference and reconstructed images

Precision	SNR	PSNR	SSIM
Double	197.72	210.32	1
Single	31.18	44.06	0.99998

Mean Image quality
of the 2048 slices

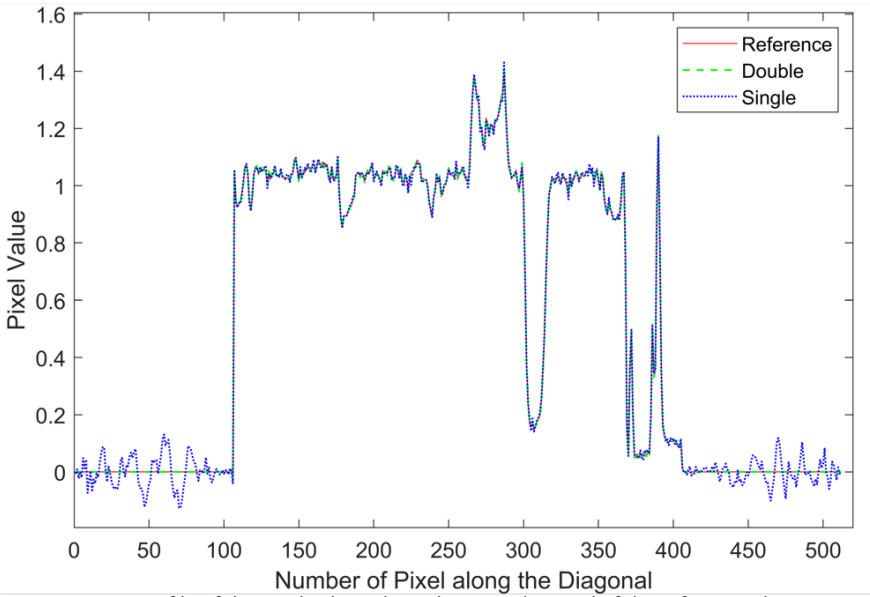


(a) Double precision. (b) Single precision.

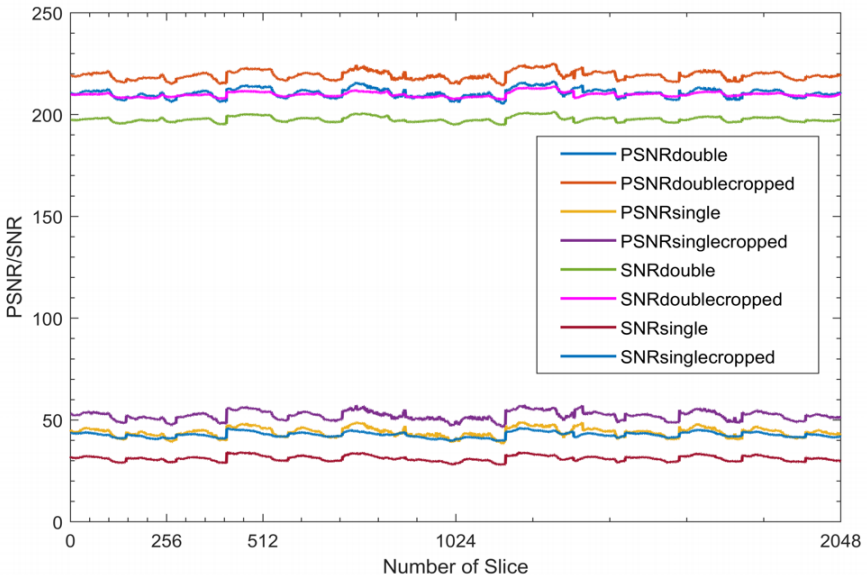
Reconstructed images errors

Precision	SNR	PSNR	SSIM
Double	209.84	219.27	1
Single	42.74	52.18	0.9999993

Mean Image quality of the
2048 slices cropping the edges



Profile of the pixel values along the main diagonal of the reference slice



PSNR and SNR of the 2048 slices for every variant

- Se están analizando los resultados empleando ASTRA para crear las proyecciones y las matrices: mucho más rápido recalcular la matriz.
- Se pretende hacer la comparativa: Precisión simple vs. doble empleando baja dosis o dosis completa.
- Si se obtienen buenos resultados con precisión simple y baja dosis (aunque se tengan que añadir filtros) se bajaría tanto la dosis como el tiempo.
- Se ha aumentado la resolución a 768×768 píxels.
- Posible combinación de la versión más rápida y con dosis más baja y herramientas de deep learning para mejorar la calidad.

- Al cambiar las simulaciones a Astra hay que adecuar medidas.
- El número de vistas tiene que ser ligeramente más alto que antes.
- Con precisión simple o baja dosis tener rango completo no garantiza imágenes limpias.
- Al aumentar la resolución 768×768 píxels aumentan los artefactos: necesarias más vistas.
- Se están buscando posibles soluciones.

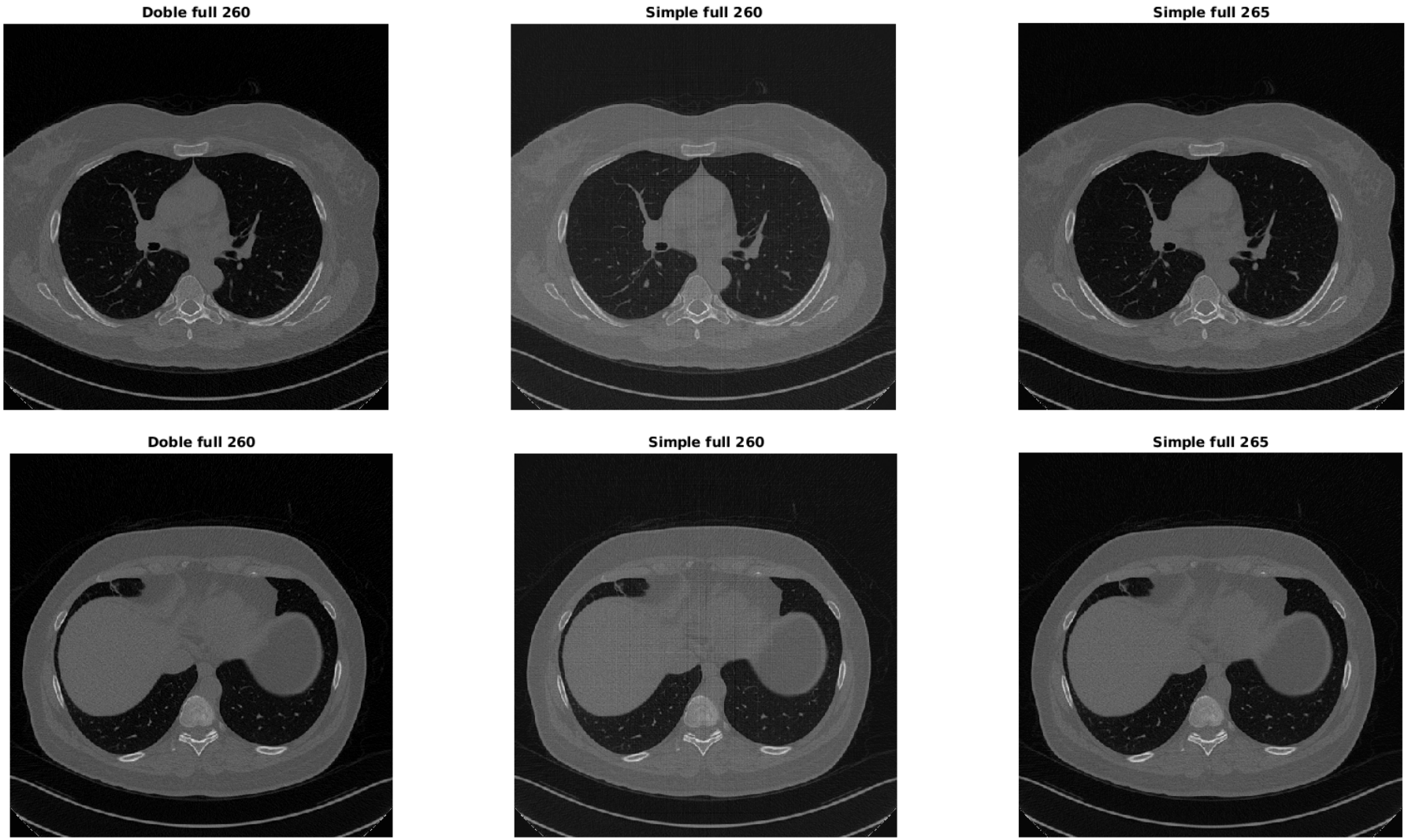


Fig. : Resolución 512: artefactos en precisión simple (hay que aumentar vistas).

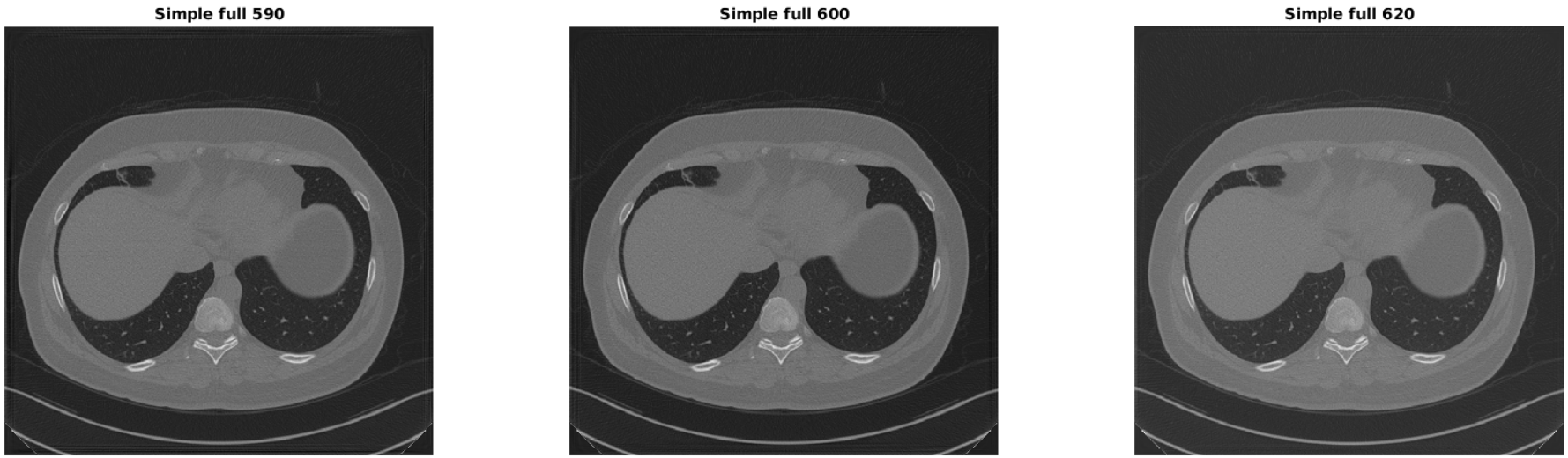


Fig. : Resolución 768: artefactos en precisión simple (hay que aumentar vistas pero no desaparecen).

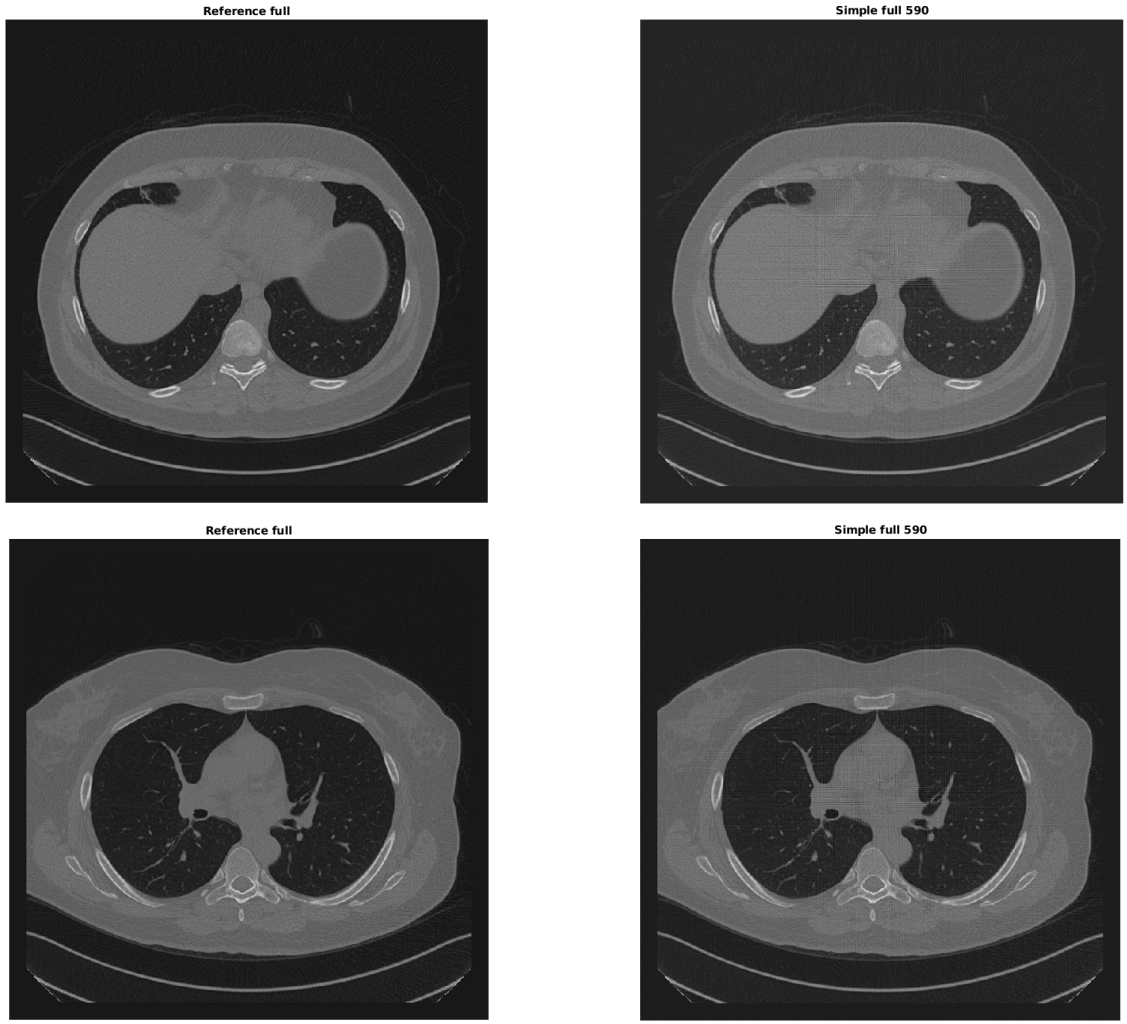


Fig. : Resolución 768: se añade borde negro.

Gracias por su atención.

Mónica Chillarón
Dpto. Sistemas Informáticos y Computación.
Universitat Politècnica de València.

Ayuda TED2021-131091B-I00, financiada por MCIN/AEI/10.13039/501100011033/
y por la Unión Europea NextGenerationEU/PRTR