



METHOD

Accelerating electron tomography reconstruction algorithm ICON with GPU

Yu Chen^{1,2}, Zihao Wang^{1,2}, Jingrong Zhang^{1,2}, Lun Li^{1,3}, Xiaohua Wan¹, Fei Sun^{2,4,5}✉, Fa Zhang¹✉

¹ Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

² University of Chinese Academy of Sciences, Beijing 100049, China

³ School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China

⁴ National Key Laboratory of Biomacromolecules, CAS Center for Excellence in Biomacromolecules, Institute of Biophysics, Chinese Academy of Sciences, Beijing 100101, China

⁵ Center for Biological Imaging, Institute of Biophysics, Chinese Academy of Sciences, Beijing 100101, China

Received: 9 February 2017 / Accepted: 7 April 2017 / Published online: 4 July 2017

Abstract Electron tomography (ET) plays an important role in studying *in situ* cell ultrastructure in three-dimensional space. Due to limited tilt angles, ET reconstruction always suffers from the “missing wedge” problem. With a validation procedure, iterative compressed-sensing optimized NUFFT reconstruction (ICON) demonstrates its power in the restoration of validated missing information for low SNR biological ET dataset. However, the huge computational demand has become a major problem for the application of ICON. In this work, we analyzed the framework of ICON and classified the operations of major steps of ICON reconstruction into three types. Accordingly, we designed parallel strategies and implemented them on graphics processing units (GPU) to generate a parallel program ICON-GPU. With high accuracy, ICON-GPU has a great acceleration compared to its CPU version, up to 83.7×, greatly relieving ICON’s dependence on computing resource.

Keywords Acceleration, Electron tomography, GPU, ICON, Missing wedge restoration

INTRODUCTION

Electron tomography (ET) plays an important role in studying *in situ* cell ultrastructure in three-dimensional space (Yahav *et al.* 2011; Fridman *et al.* 2012; Rigort *et al.* 2012; Lučić *et al.* 2013). Combining with a sub-volume averaging approach (Castaño-Díez *et al.* 2012), ET demonstrates its power in investigating high-resolution *in situ* conformational dynamics of macromolecular complexes. Due to limited tilt angles, traditional ET reconstruction algorithms including weighted

back projection (WBP) (Radermacher 1992), simultaneous iterative reconstruction technique (SIRT) (Gilbert 1972), direct Fourier reconstruction (DFR) (Mersereau 1976), iterative non-uniform fast Fourier transform (NUFFT) reconstruction (INFR) (Chen and Förster 2014), *etc.*, always suffer from the “missing wedge” problem, which causes density elongation and ray artifacts in the reconstructed structure. Such ray artifacts will blur the structural details of the reconstruction and weaken the further biological interpretation (Lučić *et al.* 2005).

In recent years, many algorithms have been proposed to deal with the “missing wedge” problem. Some of them apply prior constraints to the reconstructed tomogram to compensate the missing wedge, such as filtered

Yu Chen and Zihao Wang have contributed equally to this work.

✉ Correspondence: feisun@ibp.ac.cn (F. Sun), zhangfa@ict.ac.cn (F. Zhang)

iterative reconstruction technique (FIRT) (Chen *et al.* 2016), discrete algebraic reconstruction technique (DART) (Batenburg and Sijbers 2011), and projection onto convex sets (POCS) (Sezan and Stark 1983; Carazo and Carrascosa 1987). These constraints include density smoothness, density non-negativity, density localness, *etc.* Others try to solve the reconstruction problem as an underdetermined problem based on a theoretical framework called “compressed sensing” (CS) (Donoho 2006). Compressed sensing electron tomography (Saghi *et al.* 2011, 2015; Goris *et al.* 2012; Leary *et al.* 2013) demonstrated certain success for the data with a high signal to noise ratio (SNR) (*e.g.*, material science data or resin-embedded section data). To cope with the low SNR case (*e.g.*, biological cryo-ET data, in which a low total dose of electron is used to avoid significant radiation damage), Deng *et al.* proposed iterative compressed-sensing optimized NUFFT reconstruction (ICON) by combining CS and NUFFT together (Deng *et al.* 2016). With a validation procedure, ICON not only restores the missing information but also measures the fidelity of the information restoration. ICON demonstrated its power in the restoration of validated missing information for low SNR biological ET dataset.

However, the convergence process of ICON is time-consuming. The huge computational demand has become a major problem for the application of ICON. The traditional solution to cope with the high computational cost has been the use of supercomputers and large computer clusters (Fernández *et al.* 2004; Fernández 2008), but such hardware is expensive and can also be difficult to use. Graphics processing units (GPU) (Lindholm *et al.* 2008) can be the attractive alternative solution in terms of price and performance. In this work, we developed the parallel strategies of ICON and implemented a GPU version of ICON, named ICON-GPU. Experimental results based on a Tesla K20c GPU card showed that ICON-GPU exhibits the same accuracy and a significant acceleration in comparison with the CPU version of ICON (ICON-CPU).

RESULTS AND DISCUSSION

Reconstruction precision

First, we evaluated the numerical accuracy of ICON-GPU using the root-mean-square relative error (RMSRE) ε as Eq. 1. To avoid dividing 0 when calculating the RMSRE, we first normalized the reconstructed slices into [0,1] using Eq. 2.

$$\varepsilon = \sqrt{\frac{\sum_{i=1}^N \left(\frac{P_{\text{norm}_i} - C_{\text{norm}_i}}{C_{\text{norm}_i}} \right)^2}{N}}, \quad (1)$$

where N is the size of one slice; C_{norm} is the normalized slice reconstructed by ICON-CPU; C_{norm_i} is the value of the i th pixel in C_{norm} ; P_{norm} is the normalized slice reconstructed by ICON-GPU; P_{norm_i} is the value of the i th pixel in P_{norm} .

$$P_{\text{norm}} = \frac{P - \min P}{\max P - \min P} + c, \quad (2)$$

where P_{norm} is the normalized slice; P is the originally reconstructed slice; $\min P$ is the minimum value of P ; $\max P$ is the maximum value of P ; c is a small constant to avoid 0 in P_{norm} , in this work, $c = 10^{-7}$.

The RMSRE of ICON-GPU increases slowly with the image size; they are in the range of (6×10^{-7} , 4×10^{-6}) yielding a reasonable numerical accuracy for the float format data (Fig. 1).

Then, we evaluated the reconstruction accuracy by investigating the reconstructed tomograms. The XY-slices reconstructed by ICONs (Fig. 2B, C) show better SNR than that by WBP (Fig. 2A), yielding a better contrast to discriminate the cellular ultrastructures. Besides, ICON-CPU and ICON-GPU are identical with each other and the normalized cross-correlation (NCC) between them is 1. The XZ-slices reconstructed by ICONs (Fig. 2E, F) are also identical with each other and the ray artifacts in ICONs are significantly reduced in comparison with WBP (Fig. 2D). To be noted that, to eliminate any suspicion on the gray-scale manipulation (which could enhance the visual advantage), all images were normalized and displayed based on their minimum and maximum value.

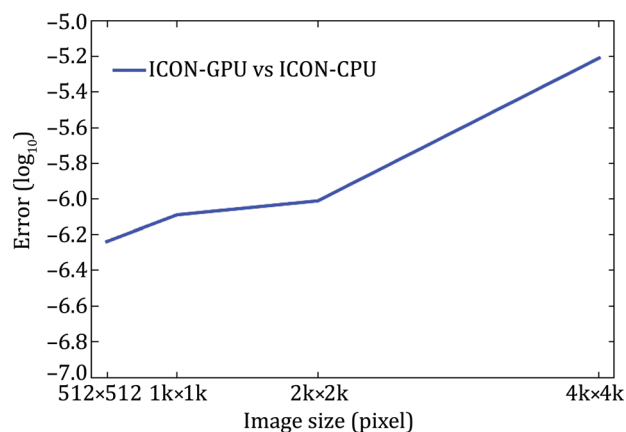


Fig. 1 The RMSREs of ICON-GPU

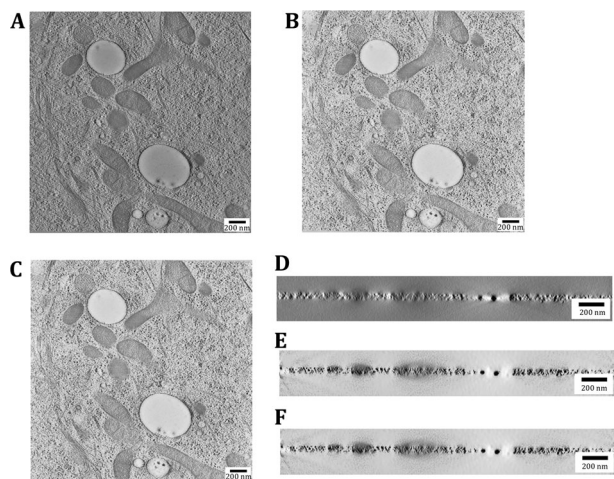


Fig. 2 Evaluate ICON-GPU by investigating the reconstructed tomograms. **A–C** The *XY*-slices of the tomograms reconstructed by WBP, ICON-CPU, and ICON-GPU, respectively; **D–F** The *XZ*-slices of the tomograms reconstructed by WBP, ICON-CPU, and ICON-GPU, respectively

We further investigated the reconstruction accuracy by the pseudo-missing-validation procedure (Deng *et al.* 2016). Here, the -0.29° tilt (the minimum tilt) projection was excluded as the omit-projection (“ground truth”) (Fig. 3A). We re-projected the reconstructed omit-tomograms at -0.29° . The re-projections of ICONs

(Fig. 3C, D) are identical with each other and the NCC between them is 1. The re-projections of ICONs are clearer in detailed structures and close to the “ground truth”, compared to that of WBP (Fig. 3B). Such visual assessments were further verified quantitatively by comparing the Fourier ring correlation (FRC) curves between the re-projections and the “ground truth”. The FRCs of ICONs coincide with each other, and they are better than that of WBP (Fig. 3E). The coincident FRCs of ICONs further demonstrate the accuracy of ICON-GPU from the perspective of restoring missing information.

Speed up

We evaluated the acceleration of ICON-GPU by comparing the running time of reconstructing one slice under 200 iterations. We reconstructed the datasets with sizes of 512×512 , $1 \text{ k} \times 1 \text{ k}$, $2 \text{ k} \times 2 \text{ k}$, $4 \text{ k} \times 4 \text{ k}$, respectively. The acceleration of ICON-GPU improves when the slice size increases (Fig. 4; Table 1). The maximum speedup is $83.7\times$ in the reconstruction of a $4 \text{ k} \times 4 \text{ k}$ slice. With the efficient acceleration, the reconstruction time of one $4 \text{ k} \times 4 \text{ k}$ slice is reduced from hours to minutes, which greatly relieves ICON’s dependence on computing resource.

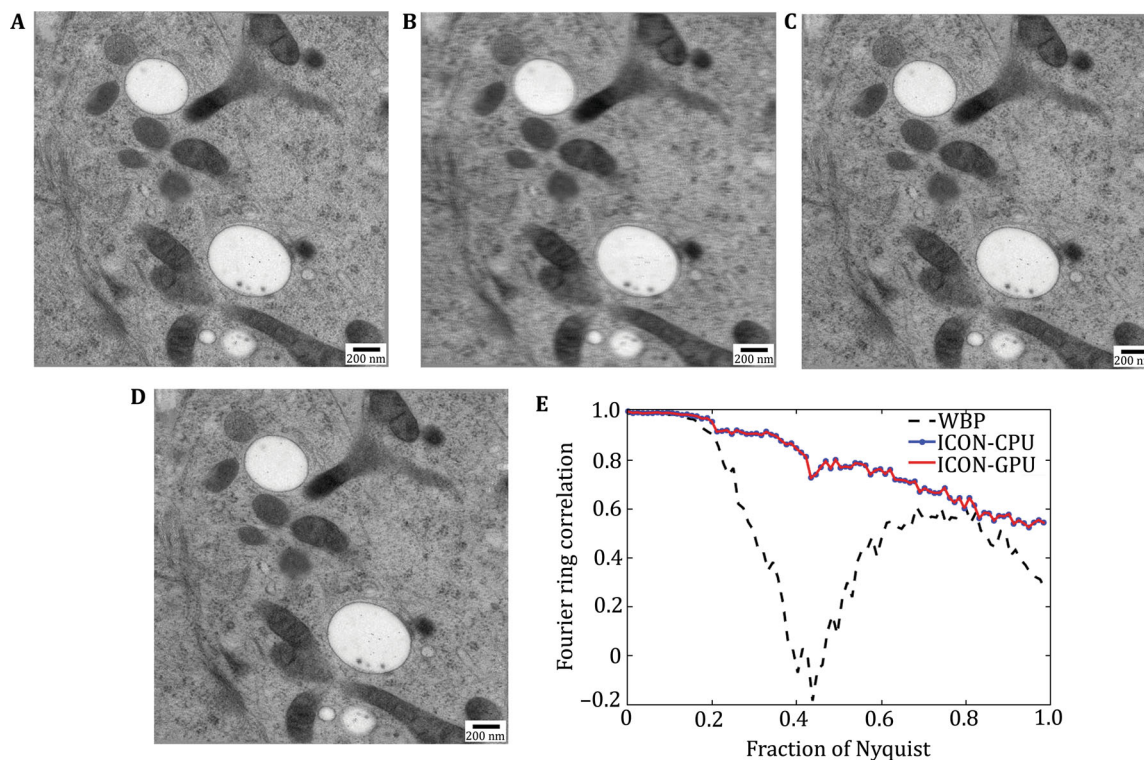


Fig. 3 Evaluate ICON-GPU by the pseudo-missing-validation procedure. **A** The omit-projection (“Ground truth”); **B–D** The re-projections of the omit-tomograms reconstructed by WBP, ICON-CPU, and ICON-GPU, respectively; **E** The pseudo-missing-validation FRCs of WBP, ICON-CPU, and ICON-GPU

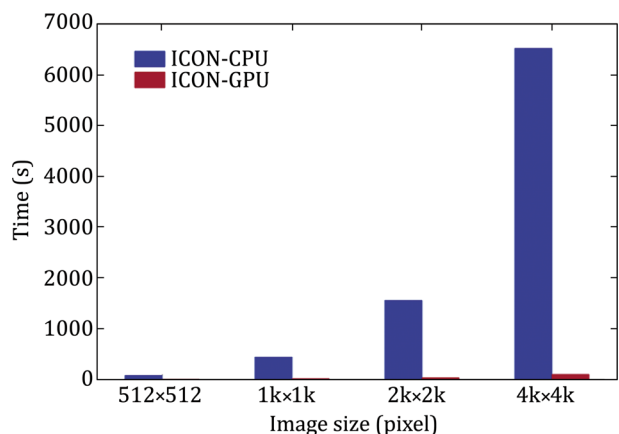


Fig. 4 The comparison of time-consuming of ICON-CPU and ICON-GPU

Table 1 The speedups of ICON-GPU compared to ICON-CPU

Image size	Speedup
512^2	$8.7\times$
1024^2	$41.5\times$
2048^2	$61.2\times$
4096^2	$83.7\times$

CONCLUSIONS

In the present work, we analyzed the iterative framework of ICON and classified the operations of ICON reconstruction into three types. Accordingly, we designed parallel strategies and implemented them on GPU to generate a parallel program ICON-GPU. We tested ICON-GPU on a resin-embedded ET dataset of MDCK cell section. The RMSRE between ICON-GPU and ICON-CPU is about $10e^{-6}$, yielding a reasonable numerical accuracy of ICON-GPU compared to ICON-CPU. In addition, ICON-GPU has the same ability of restoring missing information with ICON-CPU. In addition, ICON-GPU has a great acceleration, up to $83.7\times$ in the reconstruction of one $4\text{ k} \times 4\text{ k}$ slice in comparison with ICON-CPU.

To be noted that, ICON-GPU can also run on multiple GPU system such as TIANHE-2, a supercomputer developed by China's National University of Defense Technology, which is based on multi-core and many-core architectures (Liao *et al.* 2014).

The software package of ICON-GPU can be obtained from our website (<http://feilab.ibp.ac.cn/LBEMSB/ICON.html> or <http://ear.ict.ac.cn>).

MATERIALS AND METHODS

Iterative compressed-sensing optimized NUFFT reconstruction (ICON)

ICON is an iterative reconstruction algorithm based on the theoretical framework of "compressed sensing" and is designed to restore missing information caused by limited angular sampling (Deng *et al.* 2016). ICON is formulated as Eq. 3.

$$\operatorname{argmin} \|Px\|_{L_0}, \text{ subject to } : \|A^h W A x - A^h W f\|_{L_2} < \varepsilon, \quad (3)$$

where x is the two-dimensional (2D) reconstructed slice; W follows INFR's description (Chen and Förster 2014) and contains the weights that account for the non-uniform sampling in the Fourier space (similar to the ramp filtering in WBP); A is the projection operation, defined as a non-uniform Fourier sampling matrix, which performs Fourier transform on the non-integer grid points (NUFFT); A^h stands for the conjugate transpose of A ; f is the Fourier transform of acquired projections; $\|\cdot\|_{L_2}$ is an operator that calculates the Euclidean norm (L_2 -norm); ε is a control parameter that is determined empirically according to the noise level; $\|\cdot\|_{L_0}$ stands for the operator that calculates the number of the non-zero terms. P is a diagonal sparse transformation matrix, whose diagonal element \emptyset is defined as in Eq. 4.

$$\emptyset x = \emptyset(x) \stackrel{\text{def}}{=} \begin{cases} 0, & (\text{if } x < 0) \\ 1, & (\text{if } x \geq 0) \end{cases}. \quad (4)$$

The complete workflow of ICON can be divided into four steps (Deng *et al.* 2016):

- Step 1 Pre-processing. Align tilt series and correct contrast transfer function (CTF).
- Step 2 Gray value adjustment. Subtract the most frequently appeared pixel value in the micrographs, which is given from the embedding material (*e.g.*, resin or vitrified ice).
- Step 3 Reconstruction and pseudo-missing-validation. Reconstruct tilt series into a 3D volume with an iterative procedure of fidelity preservation and prior sparsity restriction, and evaluate the restored information with pseudo-missing-validation.
- Step 4 Verification filtering. Exclude the incorrectly restored information.

A series of tests showed that Step 3 accounts for at least 95% of the execution time of ICON. Thus, the major task for accelerating ICON is parallelizing Step 3

effectively on GPU. Since the procedures of “reconstruction” and “pseudo-missing-validation” are similar, only the parallelization of “reconstruction” will be discussed in this paper.

The major steps of “reconstruction” can be briefly described as followed.

Step 3.1: Fidelity preservation step. In this step, steepest descent method (Goldstein 1965) is used to calculate the subject function of Eq. 3 as follows:

$$r = A^h W A x^k - A^h W f, \quad (5)$$

$$\alpha = \frac{r^T r}{r^T A^h W A r}, \quad (6)$$

$$y^{k+1} = x^k - \alpha r, \quad (7)$$

where x^k is the 2D reconstructed slice of the k th iteration, r is the residual, α is the coefficient used to control the step of updating, y^{k+1} is the intermediate updating result of the $(k + 1)$ th iteration.

Step 3.2: Prior sparsity restriction step. The diagonal sparse transformation matrix P can be re-formulated as a “hard threshold”-like operation as in Eq. 8:

$$x^{k+1} = H(y^{k+1}) = \begin{cases} 0, & \text{if } y^{k+1} < 0 \\ y^{k+1}, & \text{if } y^{k+1} \geq 0 \end{cases}, \quad (8)$$

where y^{k+1} is the intermediate updating result of the $(k + 1)$ th iteration. $H(\cdot)$ is a thresholding function, x^{k+1} is the 2D reconstructed slice of the $(k + 1)$ th iteration.

We classified the operations of these two steps into three types: (1) the summation of a matrix; (2) element-wise operations of matrices; (3) the NUFFT and the adjoint NUFFT. For a fast summation of matrix, we took advantage of the API function *cublasSasum* from the standard CUDA library cuBLAS (NVIDIA Corp, 2007). For type 1 and 2, parallel strategies are proposed in the following sections.

Parallelizing element-wise operations of matrices

GPU is a massively multi-threaded data-parallel architecture, which contains hundreds of scalar processors (SPs) (Lindholm et al. 2008). NVIDIA provides the programming model on GPU called CUDA. The CUDA program running on GPU is called Kernel, which consists of thousands of threads. Thread is the basic running unit in CUDA programming model and it has a three-level hierarchy: grid, block, thread. Besides, CUDA devices use several memory spaces including global, shared, texture, and registers. Of these different memory spaces, global memory is the largest but slowest in data accessing. CUDA provides API function *cudaMemcpy* to transfer data between host memory and device

memory; the time-consuming of such transfer sometimes is non-negligible especially for an iterative procedure like ICON reconstruction.

Since micrographs in ET are usually large (e.g., $2k \times 2k$ or $4k \times 4k$ in float or short format) and exceed the limitation of most types of CUDA device memory (e.g., 16 or 48 KB for shared memory), data in ICON-GPU are restored in global memory using float format. In order to cut down the time-consuming of memory transfer, we parallelized all operations of ICON on GPU even though some operations may have negligible speedups.

To deal with element-wise operation, ICON-GPU uses a 2D distribution of threads with a fixed block size of 32×32 and a fixed grid size of 4β , β is a parameter to be determined according to the matrix size N . ICON-GPU assigns the operation of one element to one thread according to the index of element. Pseudo codes for calling a kernel function and the operations inside a kernel function are shown in Fig. 5.

Parallelizing NUFFT and adjoint NUFFT

First, we give a brief description of NUFFT. Given the Fourier coefficients $\hat{f}_k \in \mathbb{C}$, $k \in I_N$ and $I_N = \{k = (k_t)_{t=0, \dots, d-1} \in \mathbb{Z}^d : -\frac{N_t}{2} \leq k_t < \frac{N_t}{2}, t = 0, \dots, d-1\}$ as input, NUFFT tries to evaluate the following trigonometric polynomial efficiently at the reciprocal points $x_j \in [-\frac{1}{2}, \frac{1}{2}]^d$, $j = 0, \dots, M-1$:

$$f_j = f(x_j) = \sum_{k \in I_N} \hat{f}_k e^{-2\pi i k x_j}, \quad j = 0, \dots, M-1. \quad (9)$$

Correspondingly, the adjoint NUFFT tries to evaluate Eq. 10 at the frequency k .

$$\hat{h}_k = \sum_{j=0}^{M-1} f_j e^{2\pi i k x_j}. \quad (10)$$

Call of Kernel function:

```
operationFunc<<dim3(4,  $\frac{N-1}{4}+1$ ), dim3(32,32)>>(input, output);
```

Operation inside Kernel function:

```
input, output: global memory
//find the index of element (thread)
blockid=blockIdx.x+blockIdx.y*gridDim.x
threadid=threadIdx.x+threadIdx.y*blockDim.x
index=threadid+blockid*(blockDim.x*blockDim.y)
//operation
if (index<N)
    output[index]=operation(input[index])
```

Fig. 5 Pseudo codes for calling a kernel function and the operations inside a kernel function for element-wise operations

NFFT3.0 (Keiner *et al.* 2010), a successful and widely used open source C library, is used in ICON-CPU for NUFFT and adjoint NUFFT. Yang *et al.* proposed a different theoretical derivation of NFFT and demonstrated the high efficiency of GPU acceleration of NFFT (Yang *et al.* 2015, 2016). To make ICON-GPU consistent with ICON-CPU, in this work, we parallelized the NUFFT and the adjoint NUFFT based on the algorithms described in NFFT3.0 and the algorithm of 2D NUFFT is displayed in Algorithm 1.

Algorithm 1: NUFFT

Input : $M, N = \{N_1, N_2\}, \sigma = \{\sigma_1, \sigma_2\}, m, x_j \in \left[-\frac{1}{2}, \frac{1}{2}\right]^2, j = 0, \dots, M-1, \hat{f}_k \in \mathbb{C}, k \in I_N,$

$$\mathbf{n} = \sigma \mathbf{N} = \{n_1, n_2\} = \{\sigma_1 N_1, \sigma_2 N_2\}.$$

1: For $k \in I_N$ compute

$$\hat{g}_k = \frac{\hat{f}_k}{|n|c_k(\varphi)},$$

$$c_k(\varphi) = \hat{\varphi}(k_1)\hat{\varphi}(k_2).$$

2: For $l \in I_n$ compute by 2-variate FFT

$$g_l = \sum_{k \in I_N} \hat{g}_k e^{-2\pi i k(n^{-1} \odot l)}.$$

3: For $j = 0, \dots, M-1$ compute

$$f_j = \sum_{l \in I_n} g_l \tilde{\psi}(x_j - n^{-1} \odot l),$$

$$I_{n,m}(x_j) = \{l \in I_n: n \odot x_j - m \mathbf{1} \leq l \leq n \odot x_j + m \mathbf{1}\},$$

$$\tilde{\psi}(x_j - n^{-1} \odot l) = \varphi(x_{1j} - \frac{l_1}{n_1})\varphi(x_{2j} - \frac{l_2}{n_2}).$$

$\varphi(x)$ and $\hat{\varphi}(k)$ are the window functions. In this work, the (dilated) Gaussian window functions (Eqs. 11, 12) are used.

$$\varphi(x) = (\pi b)^{-\frac{1}{2}} e^{-\frac{(mx)^2}{b}} \quad \left(b = \frac{2\sigma}{2\sigma - 1} \frac{m}{\pi} \right), \quad (11)$$

$$\hat{\varphi}(k) = \frac{1}{n} e^{-b\left(\frac{\pi k}{n}\right)^2}, \quad (12)$$

where x is a component of the reciprocal points \mathbf{x} , k is a component of the frequencies \mathbf{k} , σ is a component of the oversampling factors σ with $\sigma > 1$. In this work, $\sigma = 2$, n is one component of $n = \sigma N$, $m \in \mathbb{N}$ and $m \ll n$. In this work, $m = 6$.

The operations in 2D NUFFT and 2D adjoint NUFFT can be classified into three types: (1) element-wise operations of matrices; (2) 2D FFT; (3) calculation of window functions $\varphi(x)$ and $\hat{\varphi}(k)$. The parallel strategy of type 1 is the same as the strategy described in Section "Parallelizing element-wise operations of matrices." For type 2, to achieve a high performance FFT, we took advantage of the NVIDIA's FFT library, CUFFT (NVIDIA Corp 2007). Since ICON is an iterative algorithm, 2D NUFFT and 2D adjoint NUFFT will be repeated many

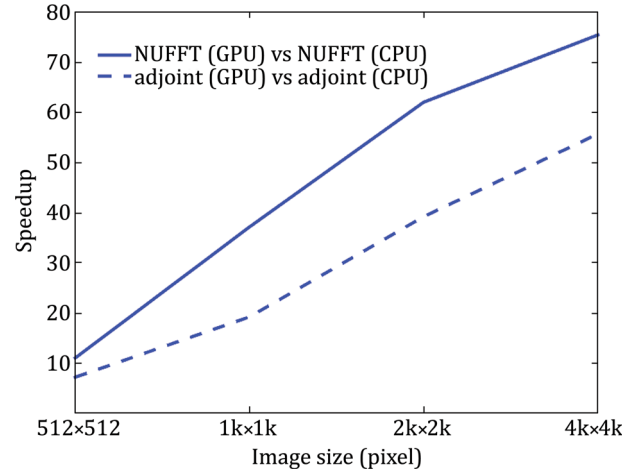


Fig. 6 The speedups of parallel NUFFTs compared to NFFT3.0

times. To cut down the time of calculation and memory transfer, we pre-computed the window functions for once and stored them in the device memory.

Parallel NUFFTs were tested using a resin-embedded ET dataset (see "Resin embedded ET Dataset" for details). Here, all CPU programs ran on one core (thread) of an Intel[®] Xeon[™] CPU E5-2620 v2 @ 2.1 GHz (six cores per CPU) and all GPU programs ran on a NVIDIA Tesla K20c (2496 CUDA cores and 5 GB device memory). The accelerations of parallel NUFFTs improve when the image size increases and are up to 75.4x for NUFFT and 55.7x for adjoint NUFFT in the transform of one 4 k × 4 k image (Fig. 6).

Resin-embedded ET dataset

We tested ICON-GPU using a resin-embedded ET dataset of MDCK cell section. The tilt angles of the dataset originally ranged from -68° to $+68^\circ$ with 1° increment. In order to verify ICON-GPU's ability of restoring missing information, we extracted every other projection from the original dataset to generate a new tilt series with 2° increment for the following experiments. The tilt series were aligned using atom align (Han *et al.* 2014). The original image size is 4 k × 4 k with a pixel size of 0.72 nm. We also compressed the tilt series with factors of two, four, eight to generate datasets with smaller sizes of 2 k × 2 k, 1 k × 1 k, and 512 × 512, respectively.

Acknowledgements We would like to thank Prof. Wanzhong He (NIBS, Beijing) for providing the resin-embedded ET dataset. All the intensive computations were performed at TIANHE-2

supercomputer in National Supercomputer Center in Guangzhou and at the high performance computers in Center for Biological Imaging, Institute of Biophysics, Chinese Academy of Sciences (<http://cbi.ibp.ac.cn>). This work was supported by the National Natural Science Foundation of China (U1611263, U1611261, 61232001, 61472397, 61502455, 61672493), Special Program for Applied Research on Super Computation of the NSFC-Guangdong Joint Fund (the second phase), the Strategic Priority Research Program of Chinese Academy of Sciences (XDB08030202), and the "973" Program of Ministry of Science and Technology of China (2014CB910700).

Compliance with Ethical Standards

Conflict of interest Yu Chen, Zihao Wang, Jingrong Zhang, Lun Li, Xiaohua Wan, Fei Sun, Fa Zhang declare that they have no conflict of interest.

Human and animal rights and informed consent All institutional and national guidelines for the care and use of laboratory animals were followed.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Batenburg K, Sijbers J (2011) Dart: a practical reconstruction algorithm for discrete tomography. *IEEE Trans Image Process* 20(9):2542
- Carazo JM, Carrascosa JL (1987) Restoration of direct Fourier three-dimensional reconstructions of crystalline specimens by the method of convex projections. *J Microsc* 145(Pt 2):159–177
- Castañó-Díez D, Kudryashev M, Arbeit M, Stahlberg H (2012) Dynamo : a flexible, user-friendly development tool for subtomogram averaging of cryo-em data in high-performance computing environments. *J Struct Biol* 178(2):139
- Chen Y, Förster F (2014) Iterative reconstruction of cryo-electron tomograms using nonuniform fast Fourier transforms. *J Struct Biol* 185(3):309–316
- Chen Y, Zhang Y, Zhang K, Deng Y, Wang S, Zhang F, Sun F (2016) Firt: filtered iterative reconstruction technique with information restoration. *J Struct Biol* 195(1):49–61
- Deng Y, Chen Y, Zhang Y, Wang S, Zhang F, Sun F (2016) Icon: 3d reconstruction with 'missing-information' restoration in biological electron tomography. *J Struct Biol* 195(1):100
- Donoho DL (2006) Compressed sensing. *IEEE Trans Inf Theory* 52(4):1289–1306
- Fernández JJ (2008) High performance computing in structural determination by electron cryomicroscopy. *J Struct Biol* 164(1):1–6
- Fernández JJ, Carazo JM, García I (2004) Three-dimensional reconstruction of cellular structures by electron microscope tomography and parallel computing. *J Parallel Distrib Comput* 64(2):285–300
- Fridman K, Mader A, Zwerger M, Elia N, Medalia O (2012) Advances in tomography: probing the molecular architecture of cells. *Nat Rev Mol Cell Biol* 13(13):736–742
- Gilbert P (1972) Iterative methods for the three-dimensional reconstruction of an object from projections. *J Theor Biol* 36(1):105–117
- Goldstein AA (1965) On steepest descent. *J Soc Ind Appl Math* 3(1):147–151
- Goris B, Broek WVD, Batenburg KJ, Mezerji HH, Bals S (2012) Electron tomography based on a total variation minimization reconstruction technique. *Ultramicroscopy* 113(1):120–130
- Han R, Zhang F, Wan X, Fernández JJ, Sun F, Liu Z (2014) A marker-free automatic alignment method based on scale-invariant features. *J Struct Biol* 186(1):167–180
- Keiner J, Kunis S, Potts D (2010) Using NFFT 3—a software library for various nonequispaced fast Fourier transforms. *ACM Trans Math Softw* 36(4):1–30
- Leary R, Saghi Z, Holland PAMDJ (2013) Compressed sensing electron tomography: theory and applications. *Ultramicroscopy* 131(8):70–91
- Liao X, Xiao L, Yang C, Lu Y (2014) Milkyway-2 supercomputer: system and application. *Front Comput Sci* 8(3):345–356
- Lindholm E, Nickolls J, Oberman S, Montrym J (2008) NVIDIA tesla: a unified graphics and computing architecture. *IEEE Micro* 28(2):39–55
- Lučić V, Förster F, Baumeister W (2005) Structural studies by electron tomography: from cells to molecules. *Annu Rev Biochem* 74(1):833
- Lučić V, Rigort A, Baumeister W (2013) Cryo-electron tomography: the challenge of doing structural biology *in situ*. *J Cell Biol* 202(3):407–419
- Mersereau RM (1976) Direct Fourier transform techniques in 3-d image reconstruction. *Comput Biol Med* 6(4):247
- NVIDIA Corp (2007) CUDA CUFFT Library
- Radermacher M (1992) Weighted back-projection methods. In: Frank J (ed) *Electron tomography*. Springer, Berlin, pp 91–115
- Rigort A, Villa E, Bäuerlein FJB, Engel BD, Plitzko JM (2012) Chapter 14—integrative approaches for cellular cryo-electron tomography: correlative imaging and focused ion beam micromachining. *Methods Cell Biol* 111:259–281
- Saghi Z, Holland DJ, Leary R, Falqui A, Bertoni G, Sederman AJ, Gladden LF, Midgley PA (2011) Three-dimensional morphology of iron oxide nanoparticles with reactive concave surfaces. A compressed sensing-electron tomography (CS-ET) approach. *Nano Lett* 11(11):4666–4673
- Saghi Z, Divitini G, Winter B, Leary R, Spiecker E, Ducati C, Midgley PA (2015) Compressed sensing electron tomography of needle-shaped biological specimens—potential for improved reconstruction fidelity with reduced dose. *Ultramicroscopy* 160:230–238
- Sezan MI, Stark H (1983) Image restoration by convex projections in the presence of noise. *Appl Opt* 22(18):2781
- Yahav T, Maimon T, Grossman E, Dahan I, Medalia O (2011) Cryo-electron tomography: gaining insight into cellular processes by structural approaches. *Curr Opin Struct Biol* 21(5):670–677
- Yang SC, Wang YL, Jiao GS, Qian HJ, Lu ZY (2015) Accelerating electrostatic interaction calculations with graphical processing units based on new developments of ewald method using non-uniform fast Fourier transform. *J Comput Chem* 37(3):378
- Yang SC, Qian HJ, Lu ZY (2016) A new theoretical derivation of NFFT and its implementation on GPU. *Appl Comput Harmon Anal*. doi:10.1016/j.acha.2016.04.009