

TECHNICAL NOTE

Open Access



# Accelerating the computation for real-time application of the sinc function using graphics processing units

Sangwoo Kim<sup>1,2</sup> and Chulhyun Lee<sup>1,2\*</sup>

## Abstract

In magnetic resonance imaging, the fidelity of image reconstruction is an important criterion. It has been suggested that the infinite-extent sinc kernel is the ideal interpolation kernel for ensuring the reconstruction quality of non-Cartesian trajectories. However, the application of the sinc function has been limited owing to its computational overheads. Recently, graphics processing units (GPUs) have been employed as fast computation tools because of their efficient and versatile parallel computation abilities. We implemented an accelerated convolution function with the sinc kernel using GPUs computing and evaluated the reconstruction performance. The computation time was significantly improved: Computation using the proposed method was approximately 270 times faster than that on a central processing unit (CPU) and approximately 4.6 times faster than that on a CPU optimized by level-3 Basic Linear Algebra Subprograms. The images reconstructed using the fast sinc function exhibited no adverse errors at all matrix sizes (resolutions). The total reconstruction time was approximately 0.3–3 s for all matrices, indicating that the sinc function could be a practical option for image reconstruction. Ultimately, its application would present a fundamental improvement to the performance of image reconstruction, and the GPU implementation of the convolution function with the sinc kernel could resolve various challenges in image data processing.

**Keywords:** Graphics processing unit, Sinc function, Computation time, Convolution function, Non-Cartesian reconstruction

## Introduction

Magnetic resonance imaging (MRI) has been widely used in medical imaging as a safe and non-invasive method for the detection and prognosis of diseases (Kraff et al. 2015; Stone et al. 2008; Wright et al. 2014). It has advanced from two- and three-dimensional imaging to four-dimensional acquisition and has been combined with parallel imaging or compressed sensing techniques for rapid scanning (Hansen et al. 2008; Nam et al. 2013; Pratz and Xing 2011; Smith et al. 2012). The benefits of these acquisition methods are generally due to superior spatial resolution, inducing enhancements to the diagnosis of diseases (Hansen et al. 2008; Kraff et al. 2015; Nam et al. 2013). However, high computational

overheads are incurred by the large datasets and the complex reconstruction process that resolves the optimization problem (Hansen et al. 2008; Kraff et al. 2015; Nam et al. 2013; Pratz and Xing 2011; Smith et al. 2012). As the entire sequence time depends on the amount of both data processing and collection of  $k$ -space (Kasper et al. 2018), a reasonable balance between the higher image quality (IQ) and the efficiency of acquisition type is required in a clinical setting (Hansen et al. 2008; Nam et al. 2013; Pratz and Xing 2011).

The drop in IQ-related motion artifacts could be prevented by an attempt to decrease the total scan time (Kasper et al. 2018; Stone et al. 2008). Non-Cartesian (NC) imaging has emerged as an alternative to standard Cartesian imaging owing to its scanning speed (Schiwietz et al. 2006; Wright et al. 2014). There are various approaches to reconstruct an image from NC scanning raw data (Pauly 2005), but the convolution function method to resample the data is preferred

\* Correspondence: [chulhyun@kbsi.re.kr](mailto:chulhyun@kbsi.re.kr)

<sup>1</sup>Department of Bio-Analytical Science, University of Science and Technology, 217, Gajeong-ro, Yuseong-gu, Daejeon, Republic of Korea

<sup>2</sup>Center for Research Equipment, Korea Basic Science Institute, 162, Yeongudanji-ro, Ochang-eup, Cheongwon-gu, Cheongju-si, Chungcheongbuk-do, Republic of Korea

because it preserves reasonable IQ (Rasche et al. 1999). Its performance is closely related to the characteristics of the kernel that is used in the reconstruction process (Jackson et al. 1991; O'sullivan 1985; Rasche et al. 1999). Jackson et al. improved results using an oversampling factor and distinct kernel widths (Jackson et al. 1991). However, some parameter settings contributed to the optimal performance. The infinite-extent sinc kernel has been suggested as an ideal interpolation kernel; however, its clinical application has been hindered due to computational limitations (Bernstein et al. 2004; Jackson et al. 1991; O'sullivan 1985). Several studies have stated that the use of the sinc-interpolation function could reduce digitization error in various fields (Bernstein et al. 2004; Wang and Liu 2015). A previous study has employed the sinc kernel for mechanics and not the MRI field; the frequency offset yielded by the sinc function was significantly less than that yielded by traditional fixed-kernel functions (Li et al. 2017). Hence, mitigating the reconstruction time of the ideal kernel may substantially improve the accuracy of the approximation, leading to a remarkable reconstruction performance by reducing artifacts related to mismatches.

Graphics processing units (GPUs) have been considered a tool for fast computation because of their efficient and versatile parallel computation (Pratx and Xing 2011). The GPU programs, which are extensions of standard C programs, can be employed without understanding the hardware structure (Hansen et al. 2008; Nam et al. 2013; Pratx and Xing 2011). Therefore, there have been approximately 1600 articles on studies pertaining to the use of GPUs in the MRI field, from 2005 to 2016. These reports have demonstrated that intensive calculation on a GPU could be accelerated by a factor of 2–285 as compared with the computation on a central processing unit (CPU) (Wang et al. 2018). Guo et al. have demonstrated that the reconstruction performance for PROPELLER trajectory can be improved by a factor of 9, with suitable image quality (Guo et al. 2009). In addition, an enhanced algorithm—reverse gridding algorithm—improved the computation by approximately 7.5 times by using GPUs (Yang et al. 2013). Moreover, compressed sensing reconstruction for 3D radial trajectories has been accelerated by approximately 54 times in cardiac MR imaging (Nam et al. 2013). They suggested that GPU computing is suitable for real-time reconstruction. Although GPU implementations have mostly focused on massive reconstruction algorithms, the use of a GPU could sufficiently resolve the fundamental challenge of sinc function computation.

Many functions are presently implemented on GPUs, for instance, a parallel of the nonequispaced fast Fourier transform for arbitrary trajectory (Sørensen et al. 2008), but the application of the sinc function on a GPU for

NC reconstruction process is yet to be reported. We hypothesized that the images reconstructed by the sinc interpolation on GPUs do not differ from the reference images and that the fast sinc function can be practically utilized in clinical settings. In this article, we review the theoretical concept in part and present an implementation of an accelerated convolution function with the sinc kernel. We then report its computational power for different spatial resolutions and evaluate its reconstruction performance. Using the proposed strategy, the computation time is reduced to a level suitable for real-time applications. Reconstruction fidelity is demonstrated by the outstanding reproduction of reference images. Lastly, we conclude the paper with a short summary of the current study and mention the scope for future work.

## Methods

### Theory

We utilized the formulation of inverse gridding (INV) operation on a 2D non-Cartesian trajectory because it is the most time-consuming step of non-Cartesian reconstruction [5]. This mathematical formulation partially follows those of Rasche et al. (1999) and Pauly (2005), and the objective is to pass a function over the data sampled on a rectilinear grid. Let  $m(x, y)$  and  $M(k_x, k_y)$  be a Fourier transform pair. To perform INV,  $m$  is divided by kernel  $c(x, y)$  for deapodization. Here,  $m$  is the intermediate image on Cartesian sampling points obtained by the gridding function as follows:

$$m_k(x, y) = \frac{m(x, y)}{c(x, y)}$$

In  $k$ -space, this yields

$$M_k(k_x, k_y) = M(k_x, k_y) \otimes C^{-1}(k_x, k_y)$$

$$\text{where } C^{-1}(k_x, k_y) = \mathcal{F} \left\{ \frac{1}{c(x, y)} \right\}$$

At this stage,  $M_k$  remains on the Cartesian data points. To estimate the non-Cartesian data, the image  $M_k$  is convolved by the kernel  $C(k_x, k_y)$  used in the gridding operation. Subsequently, it is sampled by the Shah function.

$$M(k_x, k_y) = [M_p(k_x, k_y) \otimes C(k_x, k_y)]S(k_x, k_y)$$

$$m(x, y) = [m_p(x, y) \otimes s(x, y)]c(x, y)$$

In this equation, we change the kernel  $C$  into the sinc function in the  $k$ -space domain as follows:

$$\begin{aligned}
C(k_x, k_y) &= \text{sinc}\left(\frac{\omega_x - k_x \Delta\omega_x}{\Delta\omega_x}\right) \text{sinc}\left(\frac{\omega_y - k_y \Delta\omega_y}{\Delta\omega_y}\right) \\
&= \text{sinc}\left(N \omega'_x - k_x\right) \text{sinc}\left(N \omega'_y - k_y\right) \\
&\quad k_x, k_y \in \left[-\left\lfloor \frac{N}{2} \right\rfloor, \left\lfloor \frac{N}{2} \right\rfloor - 1\right]
\end{aligned}$$

where  $\omega' = \omega/N \Delta\omega$  and  $N$  is the matrix size in the  $x$  and  $y$  directions. To reduce the computation time as much as possible, this equation is reformulated as the following matrix multiplication operation:

$$S = HEM$$

where  $M$  is the acquired  $k$ -space data, and  $H$  and  $E$  are matrices for the  $x$ - and  $y$ -axes, respectively. The first step is  $y$ -axis interpolation in  $k$ -space, yielding

$$K_{ij} = \sum_{j=1}^{nky} e_{ij} m_{ji}$$

To complete the convolution function of the 2D image, the successive step is performed by using the Hadamard product for the  $x$ -axis components.

$$S_i = \sum_{j=1}^{nkx} h_{ij} k_{ij}$$

### Reconstruction performance measurements

Reconstruction algorithms for the matrix calculations were built for both the CPU and the GPU. There are two manners to implement the sinc function for the CPU. First, MATLAB (Mathwork, MA, USA) was utilized to evaluate general computation. Subsequently, an advanced method was employed for maximizing the speed of the data processing. This method is the level-3 Basic Linear Algebra Subprograms (BLAS) technique, which performs high-performance matrix–matrix operations (Dongarra et al. 1990). The GPU program was constructed almost identically to the CPU program by using the cuBLAS technique of the CUDA library (version 7.2). The CPU algorithm was implemented on a CPU with 8 GB of memory and a core clock speed of approximately 1200 MHz. The GPU algorithm was implemented on a NVIDIA Geforce GTX 1070 with 8 GB of global memory and a core clock speed of 1506 MHz.

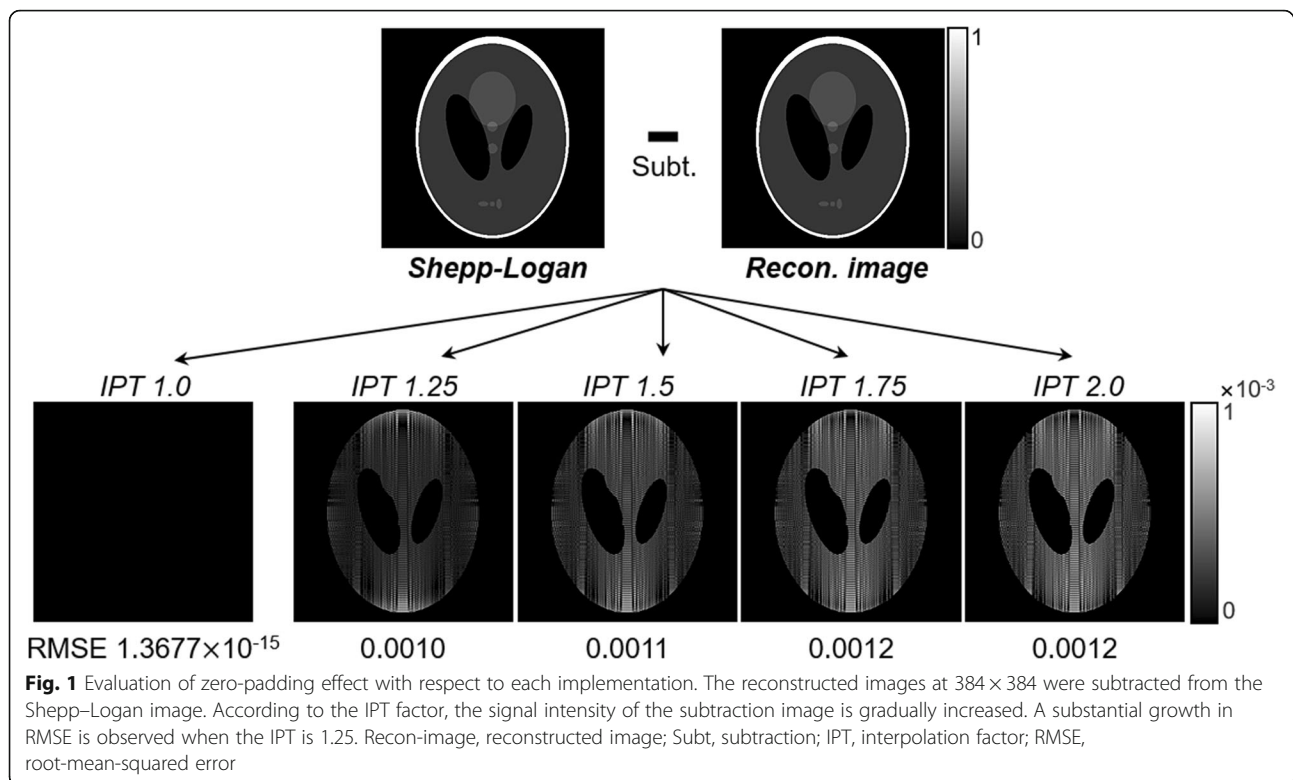
The Shepp–Logan phantom (SL) image was employed for measuring the computational time and evaluating reconstruction performance. The matrix size was varied from  $64 \times 64$  to  $512 \times 512$  in the field of view of  $240 \times 240$  mm. For the interpolation factor (IPT) setting of the sinc kernel, we evaluated the zero-padding effect with several numbers, ranging from 1 to 2 in steps of 0.25, in single resolution setting ( $384 \times 384$ ). To fairly compare

the original and reconstructed images, the voxel-based root-mean-squared error (RMSE) was employed. The increase in RMSE from which the IPT was 1.25 ( $1.3677 \times 10^{-15}$  (considered sufficiently as 0.0 in digital processing), 0.0010, 0.0011, 0.0012, 0.0012 from IPT 1 to 2) is illustrated in Fig. 1. It could be caused by the fact that the increased sampling rate due to the IPT enhances Gibbs-ringing artifacts in MR images (Bernstein et al. 2004). Thus, zero padding in our study was not considered. All reconstruction processes among the GPU and two-type CPU processing were individually performed 20 times. The average reconstruction times were recorded, and reconstruction images were acquired. To validate the fidelity of the reconstruction on the GPU, the GPU images at all resolutions were subtracted from the images computed by the CPU libraries and the SL image. Subsequently, an analysis of RMSE and percent error (PE) was conducted. The PE is the RMSE of the reconstructed image divided by the root-mean-squared value of the reference image, as presented by Stone et al. (2008).

### Results

The average processing times of the sinc convolution were measured to compare the computational performances of the CPU and the GPU. The computational time on the GPU substantially decreased as the amount of data increased. In the highest resolution here, the total reconstruction time was 727.6, 12.5, and 2.7 s with the sequence of the normal CPU time ( $\text{CPU}_{\text{ref}}$ ), the BLAS-optimized CPU time ( $\text{CPU}_{\text{opt}}$ ), and the GPU time, respectively. The corresponding values of the respective matrix sizes are listed in Table 1. To evaluate the GPU's performance, the speed-up factors were calculated as follows: (1) each CPU time was divided by the GPU time and (2)  $\text{CPU}_{\text{ref}}$  was divided by the  $\text{CPU}_{\text{opt}}$ . Although  $\text{CPU}_{\text{opt}}$  was faster than  $\text{CPU}_{\text{ref}}$  by approximately 58 times, the GPU time showed substantially rapid computation at approximately 270 times faster than  $\text{CPU}_{\text{ref}}$  and 4.6 times faster than  $\text{CPU}_{\text{opt}}$  (Fig. 2). In contrast,  $\text{CPU}_{\text{opt}}$  at low resolutions was faster than the GPU time. This implies that the time required to transfer data to the GPU device was longer than the activation of threads (Cheng et al. 2014; Hansen et al. 2008; Smith et al. 2012). These results are in good agreement with those of previous studies (Hansen et al. 2008; Nam et al. 2013; Pratz and Xing 2011; Sørensen et al. 2008; Smith et al. 2012).

We evaluated the image reconstruction errors to validate the reconstruction fidelity of the GPU program. The GPU images were identical to the images reconstructed using the CPU methods in terms of RMSE (=0.0) as shown in Fig. 3. Subsequently, we compared the GPU images to the SL image. Reconstruction errors were exhibited in the subtraction images; however, they were



sufficiently trivial ( $\text{RMSE} = 1.58 \times 10^{-15}$ ). There was no reconstruction error at all resolutions (Fig. 4). The highest RMSE value was 0.0 ( $1.08 \times 10^{-15}$ ) and the PE was similar ( $4.38 \times 10^{-13}$ ). This indicates that the GPU-based reconstructed images closely matched the reference image.

## Discussion

The objective of this study was to reduce the computation time of the convolution function using the sinc

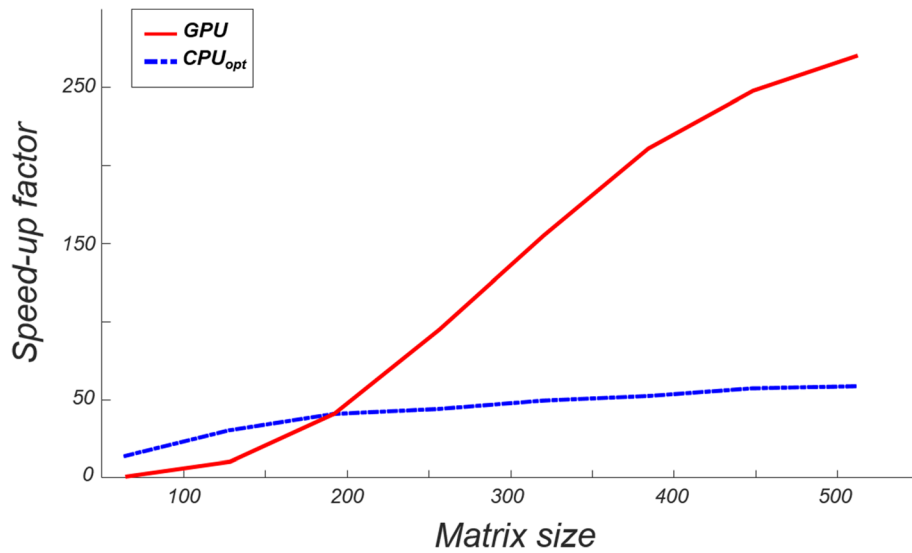
**Table 1** Average time for each individual resolution step implemented as a convolution function of the band-limited kernel

Matrix size	$\text{CPU}_{\text{ref}}$	$\text{CPU}_{\text{opt}}$	GPU
$64 \times 64$	$0.173 \pm 0.007$	$0.013 \pm 0.001$	$0.295 \pm 0.010$
$128 \times 128$	$3.195 \pm 0.164$	$0.107 \pm 0.006$	$0.317 \pm 0.009$
$192 \times 192$	$15.733 \pm 0.169$	$0.389 \pm 0.007$	$0.385 \pm 0.010$
$256 \times 256$	$49.451 \pm 1.050$	$1.133 \pm 0.029$	$0.522 \pm 0.023$
$320 \times 320$	$116.072 \pm 1.059$	$2.366 \pm 0.016$	$0.749 \pm 0.013$
$384 \times 384$	$236.919 \pm 3.379$	$4.811 \pm 0.055$	$1.124 \pm 0.017$
$448 \times 448$	$432.670 \pm 6.107$	$7.596 \pm 0.018$	$1.750 \pm 0.017$
$512 \times 512$	$727.586 \pm 8.031$	$12.479 \pm 0.076$	$2.693 \pm 0.023$

The reconstruction execution time is given in seconds.  $\text{CPU}_{\text{ref}} = \text{matrix}$ , matrix operation without optimization;  $\text{CPU}_{\text{opt}} = \text{matrix}$ , matrix operation with BLAS optimization

kernel. Compared with the CPU-based computations, the GPU-based computation achieved a significant acceleration of 4.6 to 270 times. Moreover, the reconstructed images were virtually identical to the reference images. GPU-based processing tends to perform substantially better than CPU-based processing. This could lead to fundamental improvements in image reconstruction.

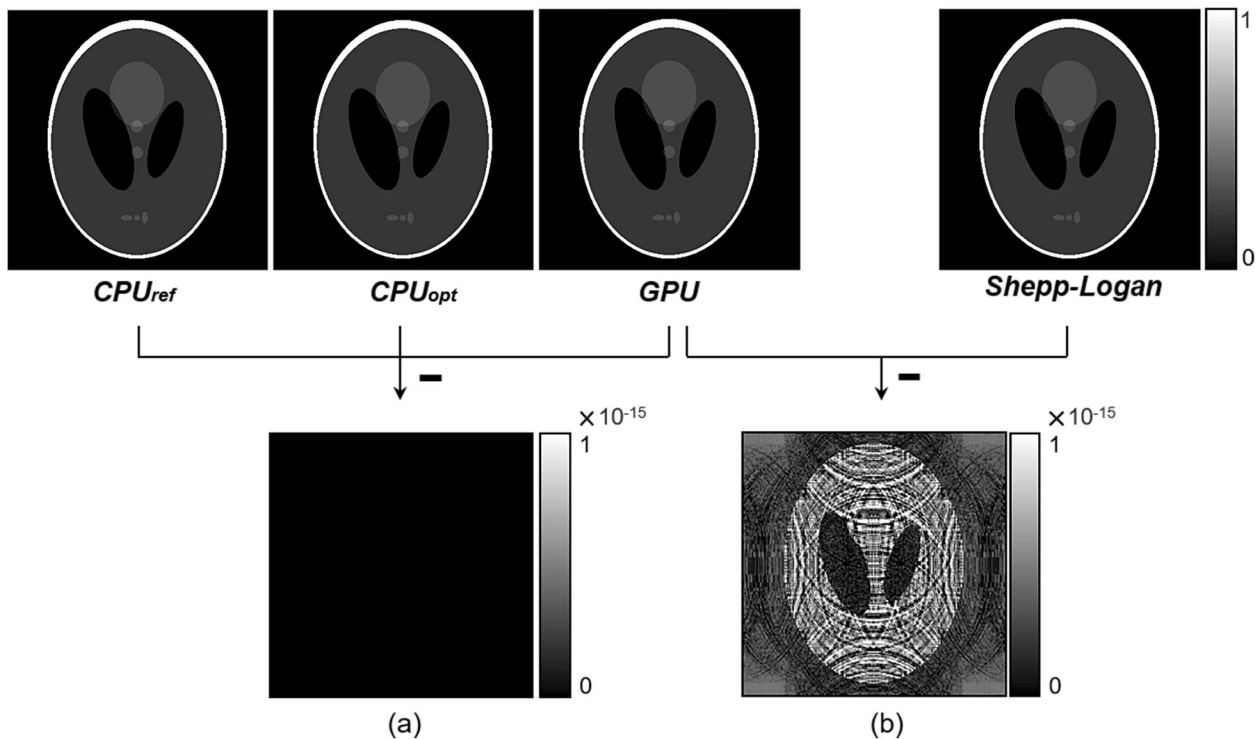
In clinical and/or research MRI settings, a fast reconstruction time is required for instantaneous and reliable responses with respect to IQ (Kasper et al. 2018; Prax and Xing 2011; Smith et al. 2012). However, the computation time is closely related to the square of the data points (Pauly 2005) and the number of algorithms connected to the process (Oppenheim and Schafer 2014). Thus, the application of the sinc kernel, which is of infinite extent, has been practically limited (Bernstein et al. 2004; Jackson et al. 1991; O'sullivan 1985). We significantly reduced the computation time of the sinc function by approximately 3 s (around 63% compared with  $\text{CPU}_{\text{opt}}$ ), suggesting that the GPU-based sinc function could be practically used in image reconstruction. The effect of GPU computing has been demonstrated by reducing massive calculations such as compressed sensing and/or parallel imaging techniques by 5–65% (Hansen et al. 2008; Nam et al. 2013; Prax and Xing 2011; Stone et al. 2008). More specifically, the total reconstruction time of prior studies (Hansen et al. 2008; Nam et al. 2013; Smith et al. 2012) was approximately 3–150 s,



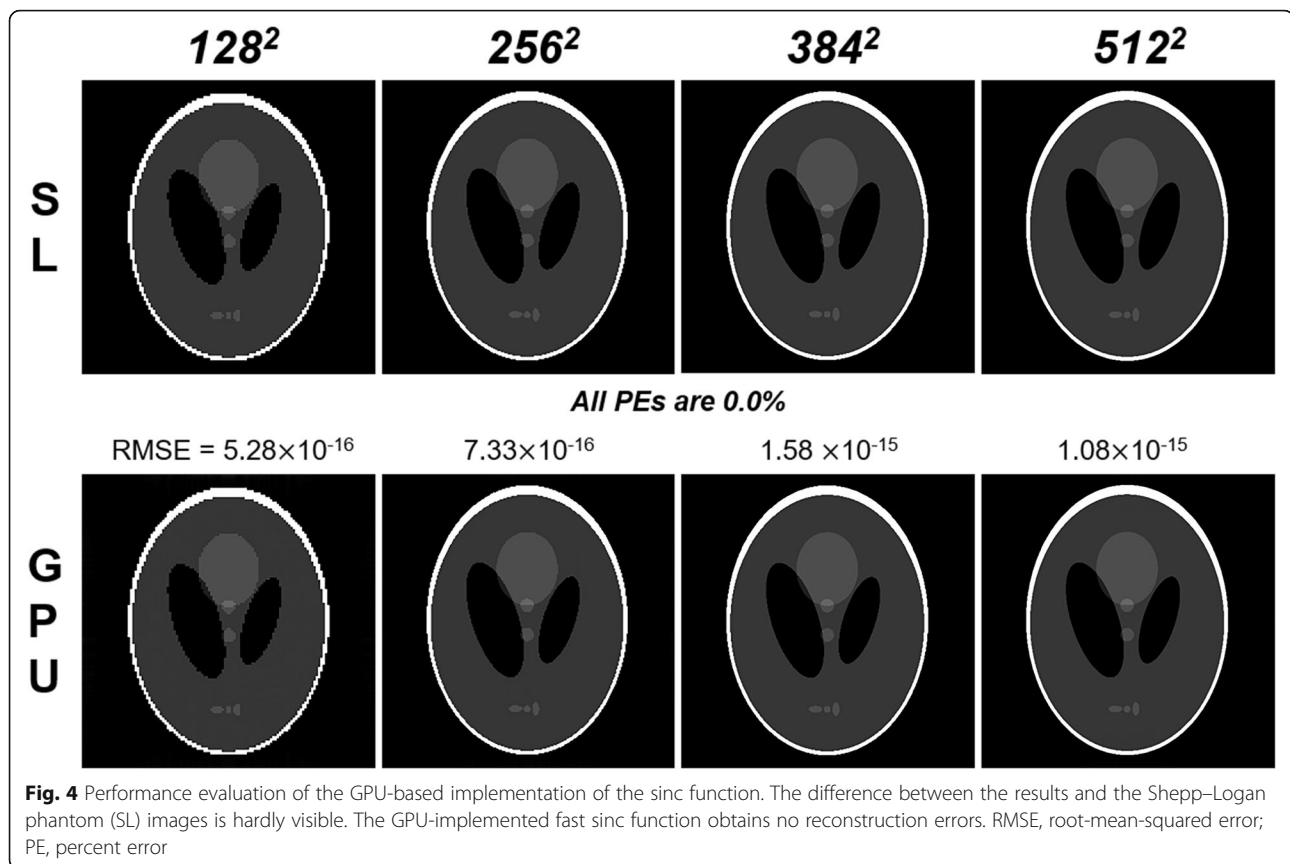
**Fig. 2** Comparison of the GPU-based reconstruction time with that of optimized CPU time (CPU<sub>opt</sub>). The speed-up factor is CPU<sub>opt</sub> and the GPU time divided by CPU<sub>ref</sub>. Note that the performance of the GPU is not substantially better at lower data quantities

which has been mentioned as real-time reconstruction conditions in  $196^2$ – $512^2$  resolutions. Hansen et al. suggested that low-latency reconstruction is suitable for real-time reconstruction, provided that the speed for

reconstruction is substantially faster than the data acquisition time (Hansen et al. 2008). In addition, the spiral acquisition for high-resolution 2D-brain imaging (FOV 230 mm, 0.5 mm in-plane resolution) requires a minute



**Fig. 3** Reconstruction performance of the sinc function with respect to each implementation at  $384 \times 384$ . **a** The GPU-computed image is subtracted from CPU-computed reconstructed images. **b** There is no RMSE with respect to the CPU images. Subsequently, the difference image is obtained by subtracting the GPU image from the Shepp–Logan image. Reconstruction errors are observed; however, the corresponding values (RMSE =  $1.58 \times 10^{-15}$ ) are considered as 0.0 in digital processing



at 7 T MRI (Kasper et al. 2018). Our result in  $512 \times 512$  resolution took less than 3 s to complete. This indicates that our study can achieve real-time image reconstruction using the sinc kernel.

Alternative kernels for a reasonable computation time not only generate aliasing artifacts but also attenuate the signal toward the edges of the field of view (Bernstein et al. 2004; Jackson et al. 1991; Rasche et al. 1999). It requires compensation steps, leading artifacts to be accentuated (O’sullivan 1985; Pauly 2005). In our results, the images reconstructed by the fast sinc function showed no adverse effects. This outstanding performance (RMSE and PE = 0.0) is attributable to the wide range of the kernel, which multiplies the center of an image domain by a constant value as a rectangular shape (Bernstein et al. 2004; O’sullivan 1985; Rasche et al. 1999). It sufficiently supports that a band-limited function basically has the least transition by the apodization (Pauly 2005). We anticipate that a realistic application of this function could simplify the non-Cartesian reconstruction process. An efficient way to decrease influences by several kernels has been demonstrated by applying an oversampling factor (Rasche et al. 1999), but zero-filling could induce increases in reconstruction time and truncation artifacts (Bernstein et al. 2004). We exhibited the reconstruction errors caused by IPT (RMSE = 0.001 – 0.0012), although

they were minor quantities. Hence, there was no IPT application step in our study. Moreover, the deapodization stage, which compensates for alternative kernels (Pauly 2005; Rasche et al. 1999), could be extracted owing to the excellent performance of the sinc function. Furthermore, this could presumably relieve the overheads of the iterative step for an actual trajectory estimation (Pauly 2005), which can additionally reduce the computation time. Consequently, the sinc function-based on GPU would present fundamental improvements in image data processing.

Our GPU-based implementation is restricted to a maximum resolution of  $512 \times 512$ . This inherently depends on the size of the global memory in the GPU and could be improved by further parallelization methods such as utilizing shared cache memory access, grid-level concurrency, and multi-GPU techniques (Cheng et al. 2014; Praxt and Xing 2011). These methods should further increase the computation power for larger datasets. We used the Shepp–Logan phantom image as an intermediate image and employed a uniform sampling pattern. Hence, the initial reconstruction condition in the inverse gridding operation was not satisfied. To complete the progress of non-Cartesian reconstruction, a gridding function with an identical kernel should be implemented for an intermediate image (Rasche et al. 1999) and the arbitrary sampling should be estimated

for the actual trajectory (Pauly 2005; Wright et al. 2014). Because the IQ obtained by non-Cartesian acquisition has been competitively improved (Kasper et al. 2018), the performance with the entire reconstruction process should be demonstrated in the in vivo imaging by a comparison with the Cartesian acquisition.

## Conclusion

We implemented a GPU-based method for the accelerated computation of the sinc function. Its application enables a band-limited function to be practically used, resulting in an improved performance with few errors. A GPU-based MRI reconstruction could be used to dramatically reduce image delivery time to physicians and researchers. In addition, the GPU-based implementation of the convolution function with the sinc kernel may help resolve various challenges in the field of MRI research.

## Abbreviations

BLAS: Basic Linear Algebra Subprograms; CPU: Central processing unit; CPU<sub>opt</sub>: BLAS-optimized CPU time; CPU<sub>ref</sub>: Normal CPU time; GPU: Graphics processing units; INV: Inverse gridding; IPT: Interpolation factor; IQ: Higher image-quality; MRI: Magnetic resonance imaging; NC: Non-Cartesian; PE: Percent error; RMSE: Root-mean-squared error; SL: Shepp–Logan phantom

## Acknowledgements

Not applicable.

## Authors' contributions

Both authors read and approved the final manuscript.

## Funding

This work was supported by the Technology Innovation Program (10067787) funded by the Ministry of Trade, Industry and Energy, Republic of Korea.

## Availability of data and materials

The computational code sets used in this study are available on a reasonable request from the corresponding author.

## Competing interests

The authors declare that they have no competing interests.

Received: 5 September 2019 Accepted: 5 February 2020

Published online: 15 February 2020

## References

- Bernstein MA, King KF, Zhou XJ. Handbook of MRI pulse sequences. Amsterdam: Elsevier; 2004.
- Cheng J, Grossman M, McKercher T. Professional CUDA C programming. Hoboken: Wiley; 2014.
- Dongarra JJ, Duff I, Hammarling S, Du Croz J. A set of level 3 basic linear algebra subprograms: model implementation and test programs. *ACM Trans Math Softw.* 1990;16(1):1–17.
- Guo H, Dai J, He Y. GPU acceleration of propeller MRI using CUDA. In: 2009 3rd international conference on bioinformatics and biomedical engineering: IEEE; 2009. p. 1–4. <https://doi.org/10.1109/icbbe.2009.5162890>.
- Hansen MS, Atkinson D, Sorensen TS. Cartesian SENSE and k-t SENSE reconstruction using commodity graphics hardware. *Magn Reson Med.* 2008;59:463–8.
- Jackson JJ, Meyer CH, Nishimura DG, Macovski A. Selection of a convolution function for Fourier inversion using gridding (computerised tomography application). *IEEE Trans Med Imaging.* 1991;10:473–8.
- Kasper L, Engel M, Barmet C, Haeberlin M, Wilm BJ, Dietrich BE, Schmid T, Gross S, Brunner DO, Stephan KE. Rapid anatomical brain imaging using spiral acquisition and an expanded signal model. *Neuroimage.* 2018;168:88–100.

- Kraff O, Fischer A, Nagel AM, Monninghoff C, Ladd ME. MRI at 7 Tesla and above: demonstrated and potential capabilities. *J Magn Reson Imaging.* 2015;41:13–33.
- Li Z, Zhu M, Chu F, He X. Adaptive radial sinc kernel distribution and its application in mechanical fault diagnosis. *Proc Inst Mech Eng C.* 2017;231:485–93.
- Nam S, Akcakaya M, Basha T, Stehning C, Manning WJ, Tarokh V, Nezafat R. Compressed sensing reconstruction for whole-heart imaging with 3D radial trajectories: a graphics processing unit implementation. *Magn Reson Med.* 2013;69:91–102.
- Oppenheim AV, Schafer RW. Discrete-time signal processing. Upper Saddle River: New Jersey: Prentice Hall; 2014.
- O'sullivan J. A fast sinc function gridding algorithm for Fourier inversion in computer tomography. *IEEE Trans Med Imaging.* 1985;4:200–7.
- Pauly J (2005) Non-Cartesian reconstruction. [http://mri-q.com/uploads/3/4/5/7/34572113/pauly-non-cartesian\\_recon.pdf](http://mri-q.com/uploads/3/4/5/7/34572113/pauly-non-cartesian_recon.pdf). Accessed 8 Jan 2020.
- Prax G, Xing L. GPU computing in medical physics: a review. *Med Phys.* 2011;38:2685–97.
- Rasche V, Proksa R, Sinkus R, Bornert P, Eggers H. Resampling of data between arbitrary grids using convolution interpolation. *IEEE Trans Med Imaging.* 1999;18:385–92.
- Schiwietz T, Chang T-C, Speier P, Westermann R. MR image reconstruction using the GPU. In: Medical imaging: physics of medical imaging; 2006. <https://doi.org/10.1117/12.652223>.
- Smith DS, Gore JC, Yankeelov TE, Welch EB. Real-time compressive sensing MRI reconstruction using GPU computing and split Bregman methods. *Int J Biomed Imaging.* 2012;2012:1–6. <https://doi.org/10.1155/2012/864827>.
- Sørensen TS, Schaeffter T, Noe KØ, Hansen MS. Accelerating the nonequispaced fast Fourier transform on commodity graphics hardware. *IEEE Trans Med Imaging.* 2008;27:538–47.
- Stone SS, Haldar JP, Tsao SC, Hwu WM, Sutton BP, Liang ZP. Accelerating advanced MRI reconstructions on GPUs. *J Parallel Distrib Comput.* 2008;68:1307–18.
- Wang H, Peng H, Chang Y, Liang D. A survey of GPU-based acceleration techniques in MRI reconstructions. *Quant Imaging Med Surg.* 2018;8(2):196–208.
- Wang Y, Liu T. Quantitative susceptibility mapping (QSM): decoding MRI data for a tissue magnetic biomarker. *Magn Reson Med.* 2015;73:82–101.
- Wright KL, Hamilton JJ, Griswold MA, Gulani V, Seiberlich N. Non-Cartesian parallel imaging reconstruction. *J Magn Reson Imaging.* 2014;40:1022–40.
- Yang J, Feng C, Zhao D. A CUDA-based reverse gridding algorithm for MR reconstruction. *Magn Reson Imaging.* 2013;31(2):313–23.

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)