Available online at www.sciencedirect.com

ScienceDirect

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

Original Research Article

A rapid algebraic 3D volume image reconstruction technique for cone beam computed tomography

Mohammed A. Al-masni[a,1], *Mugahed A. Al-antari*[a,1], *Mohamed K. Metwally*[a], *Yasser M. Kadah*[b], *Seung-Moo Han*[a], *Tae-Seong Kim*[a,*]

[a] *Department of Biomedical Engineering, College of Electronics and Information, Kyung Hee University, Yongin, Republic of Korea*

[b] *Electrical and Computer Engineering Department, King Abdulaziz University, Jeddah, Saudi Arabia*

ARTICLE INFO

Article history:
Received 8 August 2016
Received in revised form
26 June 2017
Accepted 1 July 2017
Available online 11 July 2017

Keywords:
Cone-beam computed tomography
Reconstruction from projections
Analytic reconstruction methods
Iterative reconstruction methods
Vectorization
Parallel computing

ABSTRACT

Computed tomography (CT) is a widely used imaging technique in medical diagnosis. Among the latest advances in CT imaging techniques, the use of cone-beam X-ray projections, instead of the usual planar fan beam, promises faster yet safer 3D imaging in comparison to the previous CT imaging methodologies. This technique is called Cone Beam CT (CBCT). However, these advantages come at the expense of a more challenging 3D reconstruction problem that is still an active research area to improve the speed and quality of image reconstruction. In this paper, we propose a rapid parallel Multiplicative Algebraic Reconstruction Technique (rpMART) via a vectorization process for CBCT which gives more accurate and faster reconstruction even with a lower number of projections via parallel computing. We have compared rpMART with the parallel version of Algebraic Reconstruction Technique (pART) and the conventional non-parallel versions of npART, npMART and Feldkamp, Davis, and Kress (npFDK) techniques. The results indicate that the reconstructed volume images from rpMART provide a higher image quality index of 0.99 than the indices of pART and npFDK of 0.80 and 0.39, respectively. Also the proposed implementation of rpMART and pART via parallel computing significantly reduce the reconstruction time from more than 6 h with npART and npMART to 580 and 560 s with the full 360° projections data, respectively. We consider that rpMART could be a better image reconstruction technique for CBCT in clinical applications instead of the widely used FDK method.

[*] *Corresponding author at*: Department of Biomedical Engineering, College of Electronics and Information, Kyung Hee University, 1732, Deogyeong-daero, Giheung-gu, Yongin-si, Gyeonggi-do 17104, Republic of Korea.

E-mail addresses: m.almasani@khu.ac.kr (M.A. Al-masni), en.mualshz@khu.ac.kr (M.A. Al-antari), m.kilany@khu.ac.kr (M.K. Metwally), ymk@k-space.org (Y.M. Kadah), smhan@khu.ac.kr (S.-M. Han), tskim@khu.ac.kr (T.-S. Kim).

[1] These authors contributed equally to this work.

http://dx.doi.org/10.1016/j.bbe.2017.07.001

1. Introduction

X-ray computed tomography (CT) is a medical imaging technique which gathers projection data from a patient and reconstructs tomograms from these projections. The projections data are acquired while a pair of X-ray tube and detectors rotates. Pencil, fan, and cone beam are the three main types of X-ray beam: so image reconstruction techniques rely directly on the beam type [1]. 2D image acquisition is achieved using the fan beam type and linear detector array. A line of projection data is acquired at every rotation angle to generate a sinogram which is used in the reconstruction process. Cone Beam computed tomography (CBCT) is a 3D extension of the 2D fan beam tomography: so a volume of projection data is acquired through only one rotation using 2D detector array. Due to this, an accumulative 2D projection data are generated at each rotation angle. A volume projection data at the detector planar surface is collected by integrals of rays diverging from the source that penetrate the object. There are many advantages of CBCT compared with the conventional CT, including rapid scan time, reduction of the X-ray dose, etc. [1]. Among them, the time reduction in data acquisition is one of the main advantages of CBCT [2]. Once a volume of sinogram data gets collected, 3D image reconstruction techniques are applied to reconstruct a volume of object images. There are two main categories for image reconstruction techniques, namely analytical techniques which utilize the concept of the projection slice theorem such as Feldkamp, Davis, and Kress (FDK) method [3], and iterative techniques that convert the reconstruction problem into a system of simultaneous linear equations and then to solve the problem via algebraic methods such as Algebraic Reconstruction Technique (ART) [4] and Multiplicative Algebraic Reconstruction Technique (MART) [5]. Lately, the latter is getting more attention due to the increased computing power via advanced Central Processing Unit (CPU), Graphics Processing Unit (GPU), and parallel computing.

Among the previous studies of investigating 3D reconstruction techniques for CBCT, some studies considered non-parallel reconstruction approaches and others utilized parallel computing approaches to speed up the reconstruction process. For instance, in the non-parallel reconstruction, Cengiz et al. in [6] implemented both ART and MART with the Shepp-Logan head phantom of 64 × 64 × 51 in its size. Low reconstruction speed was achieved and the computation time for one iteration took almost 13 min for both techniques with only a set of 11 projections. Also, Aviles implemented a MART method with limited number of projections. Although he used one third of the projection data, the processing time to reconstruct the results took over three hours and half [5]. In 2015, Fu et al. presented the implementation of FDK and ART [7]. Their reconstructed images had low quality in both FDK and ART. Also they indicated that their algorithms needed some advanced works to improve the reconstruction speed.

On the other hand, in the parallel computing studies, a major improvement in the time of computation was shown. In [8], Qiu et al. presented a significantly reduced reconstruction time of 1600 s for one iteration via ART on a computer of 64 bit 3.33 GHz Linux with RAM of 32 GB for an image volume of 256 × 256 × 256. In [9], Fan and Xie utilized a GPU on a total of 768 processor cores and improved the computation time of ART for only 60 projections of an object size of 256 × 256 × 256 from 40 to two seconds per projection, but they mentioned that their technique did not produce high quality images. To our best knowledge, there has been no studies of MART via parallel computing.

In this study, we propose a rapid parallel version of MART (rpMART) via a vectorization process for CBCT and investigate its performance against the parallel version of ART (pART) and the conventional non-parallel 3D image reconstruction techniques, namely npART, npMART and npFDK. The image reconstruction performances are compared in terms of reconstruction quality, speed, and dose quantity of X-ray radiation (which is proportional to the number of projections) using analytical phantoms. The results of pART and rpMART show significant reduction in computation time compared with the non-parallel techniques. Also our rpMART implementation provides better image quality index than pART and npFDK even with the limited numbers of projections. This paper is organized as follows. First, we present an overview of the projections data generation. Second, we describe in details the FDK, ART and MART reconstruction techniques. Then, the proposed implementation of parallel image reconstruction is presented. Finally, we compare the performance of rpMART against pART, npART, npMART and npFDK.

2. Materials and methods

2.1. Projection data generation

Projection is a combination of rays that penetrate an object at the same orientation. The main principle of projection is considered as ray-sum, line integral, or Radon transformation [10]. The general 2D projection $p(i, \theta)$ with a projection angle of ε is as follows:

$$p(i,\theta) = w_{i1}x_1 + w_{i2}x_2 + w_{i3}x_3 + \cdots + w_{in}x_n, \quad i = 1, 2, 3, \ldots, n, \tag{1}$$

where x_j is the intensity of jth pixel in an image and the w_{ij} is the weight of the ray in the pixel j that hit the detector bin i.

For CBCT, this 2D projection is extended into 3D. Siddon's algorithm is one of the best techniques to perform this 3D projection [11–13]. This algorithm is implemented by the following consecutive processes. First, the object is placed into three equal sets of orthogonal planes N_X, N_Y and N_Z. Second, the entrance and exit points of the ray, that penetrates the object, are determined by obtaining the intersection of the ray with the plane boundaries in each direction. Then, the range of the plane indices between these points was calculated. Finally, the voxel indices and the segment length of the ray within each voxel (i.e., weight of voxels) are calculated. After computing the first projection at the first angle, the system is rotated to get the next projection position at a different angle. In this work, we have simulated the projection data with a digital Shepp-Logan head phantom of 128 × 128 × 128 in its size.

2.2. Image reconstruction techniques

2.2.1. FDK method

The FDK method, which is also called filtered back-projection (FBP) method, is the most widely used algorithm for the cone beam image reconstruction. It is an analytical approach for image reconstruction which is based on Fourier Transform [3]. The FDK image reconstruction is performed in three steps. First, each projection data is weighted according to the position of each ray from the center of the flat panel detector. Second, row-by-row filtering of the weighted projection data is achieved. Third, the whole filtered projection data gets back-projected to obtain a reconstructed image volume data [14–18]. For FDK method, we utilized an open source cone-beam CT reconstruction (OSCaR) Tool for Imaging Research [19]. In this work, we have implemented only a non-parallel version of FDK (npFDK).

2.2.2. *ART method*

In ART, the reconstruction problem can be formulated as a set of linear system equations. Compared to the analytical approach of FDK, ART requires more computational time [14,15]. The process is summarized in the following steps [20–23]. First, an initial volume is created with N elements which represent the number of voxels in a reconstructed volume. Then, projections p_i are calculated for all the rays that pass through the voxels x_j as follows:

$$\sum_{j=1}^{N} w_{ij} x_j = p_i, \tag{2}$$

where $\{j = 1, 2, \ldots, N\}$ is the index for the voxel of x, $\{i = 1, 2, \ldots, M\}$ is the index of the rays, and the weights, w_{ij}, are calculated using the Siddon's algorithm. After that, the correction is computed as the difference between the measured projections p_i and rebuilt $\sum_{j=1}^{N} w_{ij} x_j$. Finally, ART solves the set of equations where the update for each voxel j is achieved through k iterations as:

$$x_j^{(k+1)} = x_j^{(k)} + \lambda \frac{p_i - \sum_{n=1}^{N} w_{in} x_n^{(k)}}{\sum_{n=1}^{N} w_{in}^2}, \tag{3}$$

where λ is a relaxation factor that controls the convergence rate, typically chosen $\ll 1.0$ [8,24]. We have implemented pART and npART in this work.

2.2.3. *MART methods*

MART is a nonlinear iterative technique for 2D and 3D image reconstruction in CT. It has an ability to reconstruct a volume of images with an underdetermined set of linear equations [6]. Unlike ART, MART must has nonzero initial volume. Also MART has different update and correction formula which is in a multiplicative form as follows [25,26]:

$$x_j^{(k+1)} = x_j^{(k)} * \left(\frac{p_i}{\sum_{n=1}^{N} w_{in} x_n^{(k)}} \right)^{\lambda}. \tag{4}$$

MART produces the reconstructed volume with a higher contrast of the object. Thereby, MART is considered as a better method in the reconstruction of sparse distributions than the additive ART in which it leaves the artifact in the reconstructed volume [6]. This advantage leads MART to be a better choice than other iterative methods. In this study, we have implemented npMART and a rapid parallel version of MART (rpMART).

2.3. *Parallel implementation of ART and MART*

In this section, we describe our proposed parallel computing on 3D image reconstruction process via ART and MART (i.e., pART and rpMART).

There have been studies of investigating parallel computing of ART. For instance, in [8], Qiu et al. implemented an algorithm for sparse linear equation and least square to speed up the ART reconstruction time using a high computer specification of 64 bit 3.33 GHz Linux with RAM of 32 GB. In 2015, Fan and Xie implemented a parallel algorithm via blockwise estimation that distributes the rays into blocks and then fabricating the rays in each block in parallel pattern [9]. They utilized a GPU on a total of 768 processor cores, but it is difficult to use in practice. As seen, the previous studies attempted to reduce the computation time of CBCT images reconstruction by utilizing a CPU or GPU parallel computing without vectorization process of projection calculations. Also most previous studies investigated the parallel computing of ART, and no attempt has been made on the parallel version of MART so far.

In our study, we implement two types of parallel computing processes sequentially in both ART and MART, which are vectorization and partitioning parallelization via CPU, to speed up the volume reconstruction time. In this work, we propose the vectorization process which is the essential part to reduce the computation time. We applied our vectorization process to parallelization via the partitioning approach, which utilizes four processor cores, to show the ability of the proposed process to work with multiple processors.

Vectorization is a digital procedure that utilizes vector operations instead of element by element loop-based operations. Automatic vectorization in parallel computing is a linear transformation that converts 2D data into 1D vector. In other words, it is a compiler optimization that transforms loops to vector operations. This process in general speeds up the processing time. In ART and MART, we implemented the proposed vectorization process through projection calculations to speed up the volume reconstruction time. These calculations are utilized to update the weight of voxels to reconstruct the volume of images corresponding to Eqs. (3) and (4). In this study, vectorization process is achieved through three consecutive steps. First, for each 2D projection view, we determine the indices of all voxels x_m through which the ray passes as shown in Fig. 1(a). These voxels are a subset of the whole voxels in the volume where $m = 1, 2, \ldots, L$ and $L \ll N$. Second, utilizing the Siddon's ray tracing algorithm, we obtain the segment length or weight w_m of each voxel at each determined index. The other voxels are excluded in this step because their weights at this particular ray are zeros, producing a sparse weighting matrix. Thus, the only components $x_m = [x_1, x_2, \ldots, x_L]$, which have nonzero weights $w_m = [w_1, w_2, \ldots, w_L]$, are processed as follows to get the computed projection CP,

$$CP_i = \sum_{m=1}^{L} w_{im} x_m^{(k)}. \tag{5}$$

To avoid the complicated multiplication and addition operations, we only transform these voxels' intensities and their corresponding segment lengths into two separate vectors as illustrated in Fig. 1(b) and (c), respectively. Finally, to get the calculated projection (CP) value, we reduce the computation operations through the summation of product for voxel intensities $[x_1, x_2, \ldots, x_L]$ with weights $[w_1, w_2, \ldots, w_L]$ at each index. Eqs. (6) and (7) explain algorithmically how the vectorization process is achieved and also express the alternative formula for Eqs. (3) and (4), respectively, as follows,

$$x_j^{(k+1)} = \begin{cases} x_j^{(k)}; & \text{if } w_{ij} = 0, \\ x_m^{(k)} + \lambda \dfrac{p_i - CP_i}{\sum_{m=1}^{L} w_{im}^2}; & \text{if } w_{ij} \neq 0 \text{ and } j = m \end{cases} \tag{6}$$

$$x_j^{(k+1)} = \begin{cases} x_j^{(k)}; & \text{if } w_{ij} = 0, \\ x_m^{(k)} * \left(\dfrac{p_i}{CP_i}\right)^{\lambda}; & \text{if } w_{ij} \neq 0 \text{ and } j = m. \end{cases} \tag{7}$$

This vectorization process improves the computation time of pART and rpMART.

Domain decomposition or partitioning is a key step in the partitioning parallelization process. In general, ART and MART can be executed so that the entire "blocks" of projections can be made simultaneously instead of sequentially. The simultaneous projections in each such block are mathematically equivalent to perform the projections in any sequential order as reported in [23]. Also Fan and Xie in [9] achieved their parallel implementation by allotting rays into blocks, and then the rays in the same block are computed parallel. The partitioning process is considered as a single instruction multiple data (SIMD) approach which has multiple processing units that work under the control of a single control unit (i.e., single instruction). All processing units simultaneously execute the same instruction and each processing unit can operate on a different data [27]. According to the number of processors, partitioning parallelization technique divides the projection data into a set of partitions. Thus, partitioning parallelization process should not be arbitrary and the partitions should be executed independently. In our case, only a CPU with four processor cores are used to achieve this process with a computer of 64 bit, 2.2 GHz, Intel(R) Core(TM) i7-3632QM, and 6 GB RAM. Thus, each partition only consists of four projections data that are selected randomly as shown in Fig. 2. Those four projections are synchronously passed into four independent processors at a time without any overhead. At each processor, vectorization process is achieved with each projection as mentioned above. In pART and rpMART, each projection partition is processed with the next partition to reconstruct a 3D volume of images. The data processing in this work was done using MATLAB version 8.5.0.197613 (R2015a) [28]. In general, the amount of RAM is important to process a larger volume of 3D data while the number of cores affects the speed of the pART and rpMART. Our proposed pART and rpMART provide a rapid algebraic 3D volume reconstruction. Algorithm steps of rpMART is illustrated in Algorithm 1.

3. Results and discussion

3.1. Computing projection data

In this study, with a rotation angle step of one degree, the resulting size of the projections of 128 × 128 × 360 was obtained, where 128 × 128 is the detector matrix size and 360 is the number of projection angles.

In Fig. 3(a)–(c), examples of three projections of a digital Shepp-Logan head phantom at different angles 0°, 45° and 90°

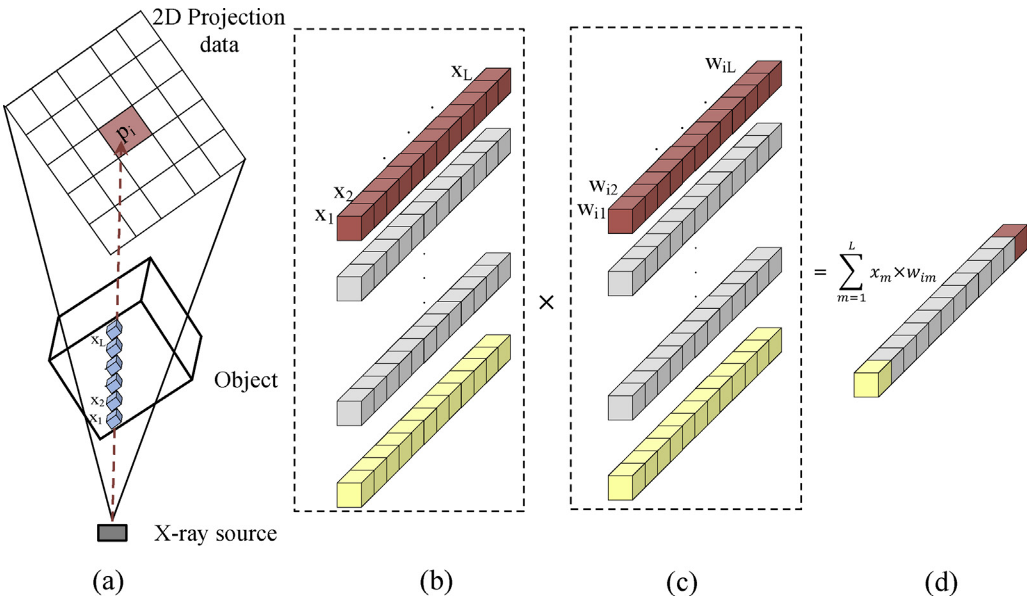

Fig. 1 – Conceptual diagram of vectorization process at a specific 2D projection data, (a) ray tracing from a source point to each detector element, (b) intensity values of the voxels through which the ray passes through, (c) segment lengths or voxel weights, and (d) vectorization of the calculated projection.

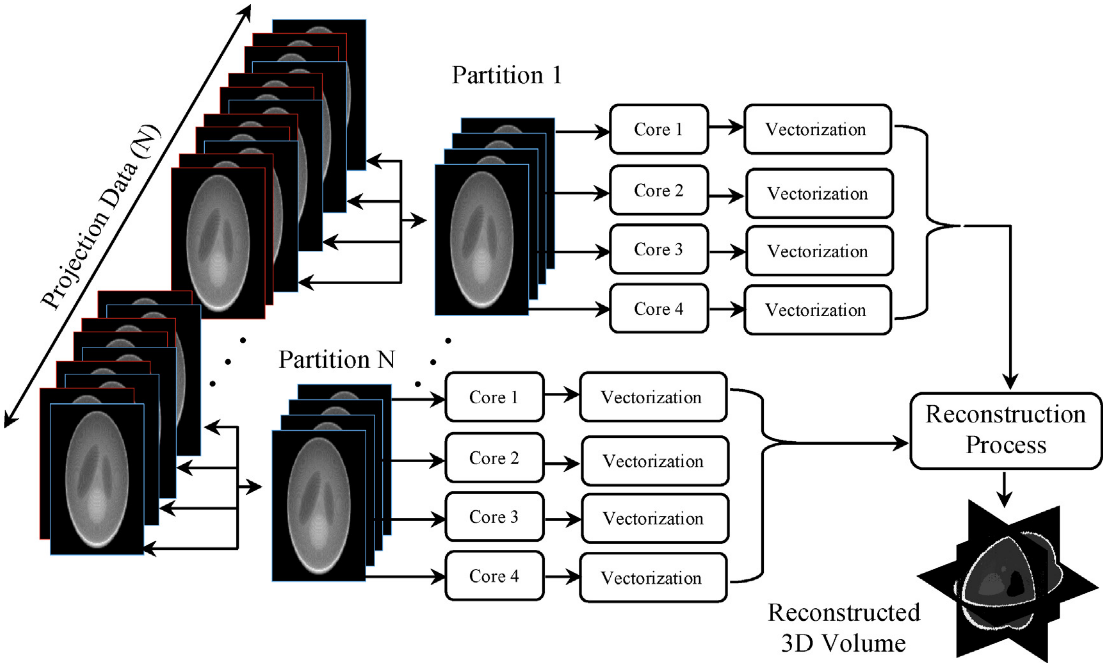

Fig. 2 – Parallelization processes for iterative 3D image volume reconstruction techniques.

are illustrated. Fig. 3(d) shows the sinogram of the horizontal central lines for all projections data. The results show that the computational time of the projection data generation is high. So, reducing the computation time is achieved by applying the proposed vectorization parallel process. Fig. 3(e) illustrates the computation time with and without the vectorization parallel process for different phantom sizes. The computation time of the projection process is clearly decreased utilizing the vectorization process. With a phantom of 128 × 128 × 128 in its size, the projection computation time of 360 views before the vectorization process takes 3.41 h, but only 16.81 min with the parallel process. The generation of projection data from our test object took 16.81 min with our process. When the object size increases, the reduction in time will increase significantly. These projections data are used as raw input data to the reconstruction techniques to get the 3D volume of images.

3.2. *Reconstruction results*

The reconstructed volumes of images from rpMART are compared with those from pART, npART, npMART and npFDK.

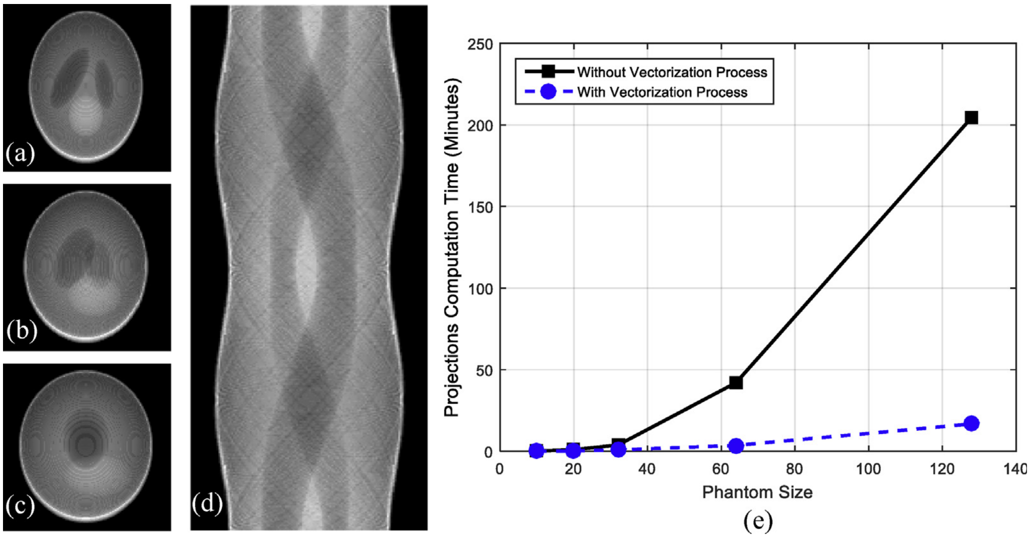

Fig. 3 – Cone beam projections of a Shepp-Logan head phantom at the angles of (a) 0°, (b) 45°, (c) 90°. (d) Sinogram of the horizontal central lines for all projections data. (e) Impact of vectorization process on the computation time of the projection data generation with respect to the phantom size.

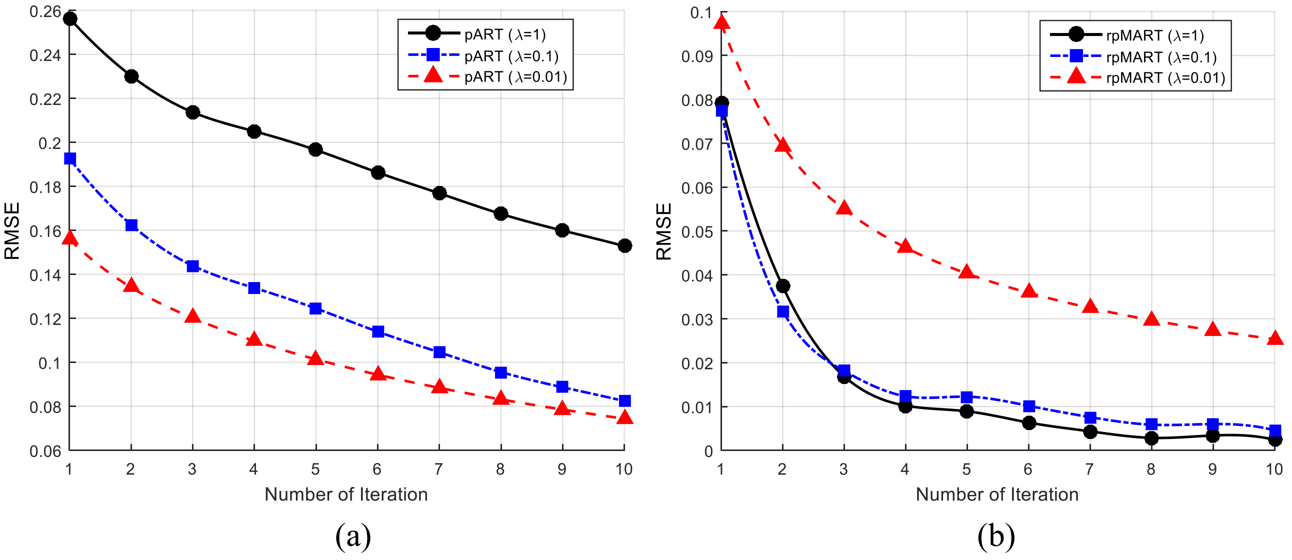

Fig. 4 – Effect of the relaxation factor (λ) on the performance of (a) pART and (b) rpMART on RMSE with respect to the number of iteration with the full projections.

As for quantitative analysis of performance evaluation, we utilize Signal to Noise Ratio (SNR), Peak Signal to Noise Ratio (PSNR), Root Mean Square Error (RMSE), and Structural Similarity Index Measure for image quality (SSIM). More details of these indices are available in [29,30].

3.2.1. *Effect of the relaxation factor on pART and rpMART*

The iterative techniques such as ART and MART depend on two parameters: namely the relaxation factor (λ) and number of iterations which control the convergence rate. The relaxation factor can be varied with a different value between 0 and 1. We have investigated the effect of this parameter at 1, 0.1, and 0.01 with respect to the number of iterations. Fig. 4 shows the effect of λ on the performance of pART and rpMART in term of RMSE with utilizing full projections data. The results show that in the case of pART, λ with a smaller value (i.e., 0.01) provides a better image quality. In the case of rpMART, better results are produced when λ has a larger value equal to 1. This difference of choosing the λ in both pART and rpMART is due to its different use in Eqs. (3) and (4) on the convergence update.

Fig. 5 illustrates the effects of the number of iterations in image reconstruction on the image quality index (SSIM) and RMSE with $\lambda = 0.01$ for pART and $\lambda = 1$ for rpMART. Increasing the iteration number leads to reduce the RMSE and increase the SSIM. This result shows that rpMART provides better SSIM and RMSE than those of pART.

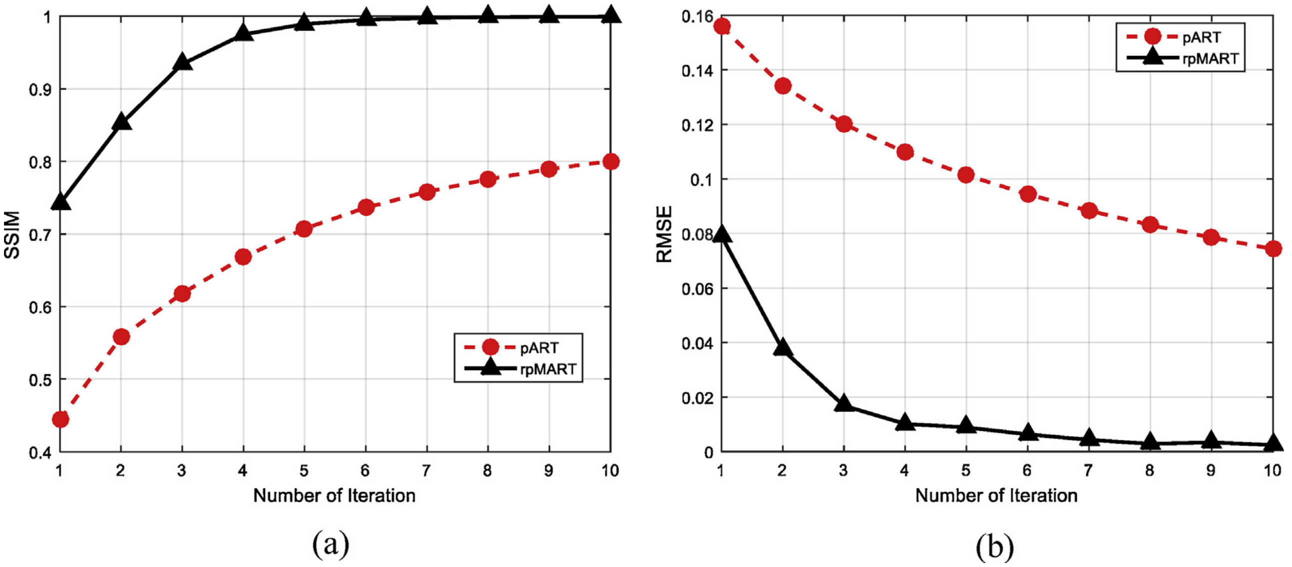

Fig. 5 – Performance of pART and rpMART in terms of (a) SSIM and (b) RMSE vs. the number of iterations with the full projections data.

3.2.2. *Quality of the reconstructed 3D image via pART, rpMART and npFDK*

The results of all reconstruction techniques using the full scan projections data of the Shepp-Logan phantom are depicted in Fig. 6. The results show the coronal, sagittal and, axial slices of the reconstructed volume. The performance of the npART and npMART is similar to pART and rpMART in term of image quality, respectively. The significant difference of these methods is in the computation time. The results of rpMART are better than pART and npFDK. The red arrows in Fig. 6 indicate the smallest object in the coronal and sagittal slices. This object is not clearly reconstructed by npFDK and pART but it is reconstructed by rpMART. Also, npFDK tends to produce noisy background which are mainly produced during the back-projection process compared with other techniques. Higher contrast between the objects and background are produced from rpMART.

To demonstrate the efficiency of the implemented methods, the line profiles from the reconstructed coronal slice are plotted in Fig. 7. rpMART produces the reconstructed life profile highly close to the original while highly preserving the edges. The behavior of the horizontal line profile of npFDK is close to the original but produces some blurring in the images. Table 1 shows the quantitative performance measures of the pART, rpMART and, npFDK. The high image quality is produced with rpMART for each slice of the reconstructed volume. The RMSE is equal 0.138, 0.074 and, 0.002 for npFDK, pART, and rpMART, respectively. Also rpMART outperforms pART and npFDK with the best image quality index of 1. The largest SNR and PSNR indices with rpMART are considered as the best quality images. Our results can be compared with those of Simon et al. [17] since they used the stationary reference case of the thorax with a full 640 projections in a 360° views (i.e., no motion as same with our case).

3.2.3. *Effect of the limited number of projections*

Investigation of utilizing the limited amount of the whole projection data such as 1/2, 1/3, 1/4, and 1/8 of the full projection data in the reconstruction process is an issue to reduce the X-ray radiation dose and decrease the computational time. Fig. 8 shows the performance of pART, rpMART, and npFDK with respect to the number of projections that are evaluated in term of RMSE and SSIM. rpMART produced the promising results in terms of SSIM and RMSE in comparison to pART and npFDK. In the case of pART, the quality of image is reduced significantly. Also the results show that rpMART provides higher image quality at a lower amount of projections compared with other techniques. The results show that even full projections set are utilized, the better image quality is achieved using the iterative techniques against the FDK method. This due to the FDK method has only one back-projection, while the iterative methods uses multiple forward

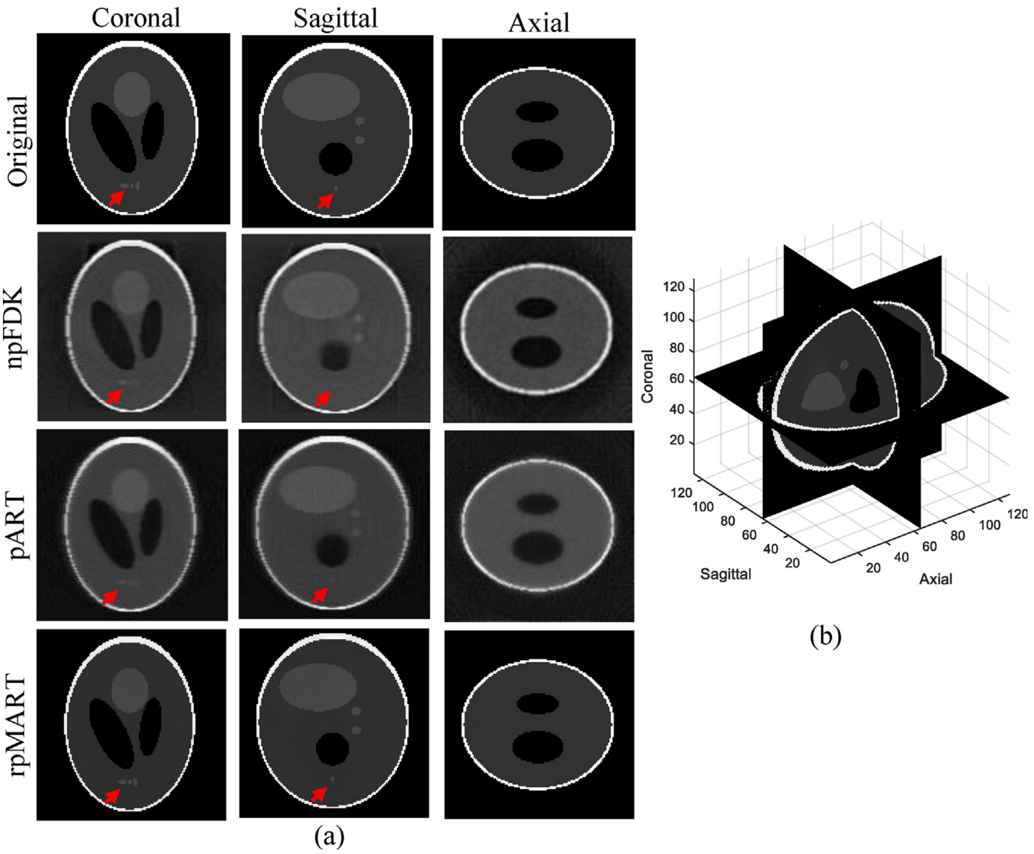

Fig. 6 – (a) The reconstructed sagittal, coronal, and transaxial images in comparison to the original from npFDK, pART, and rpMART using a full 360° projections with 10 iterations. (b) Cut views from the 3D volume data via rpMART.

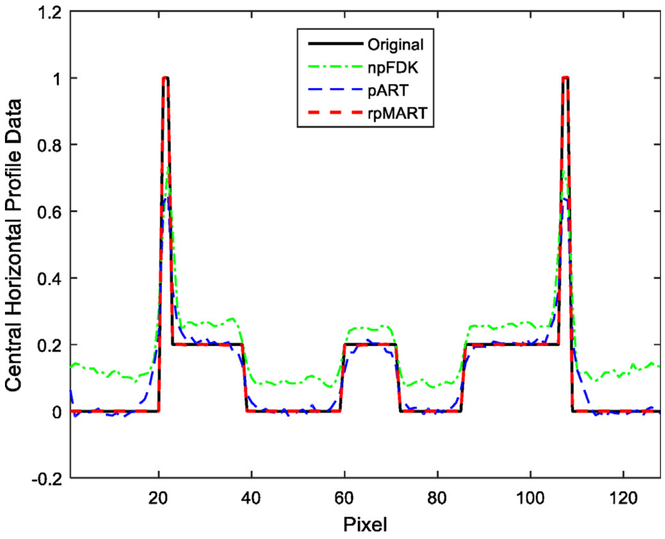

Fig. 7 – Line profiles from the reconstructed coronal slice via pART, rpMART, and npFDK compared to the line profile of the original slice.

Table 1 – Quantitative performance measurements for pART, rpMART, and npFDK with the full projection data after 10 iterations.

Slice	Reconstruction method	Measurements			
		RMSE	SNR (dB)	PSNR (dB)	SSIM
Coronal	npFDK	0.138	−9.77	17.20	0.391
	pART	0.074	4.30	22.58	0.800
	rpMART	0.002	11.37	52.41	0.9995
Sagittal	npFDK	0.151	−2.85	16.41	0.492
	pART	0.074	4.14	22.56	0.817
	rpMART	0.003	7.74	49.74	0.9992
Axial	npFDK	0.127	0.163	17.90	0.355
	pART	0.077	3.94	22.30	0.722
	rpMART	0.001	11.81	59.38	0.9997

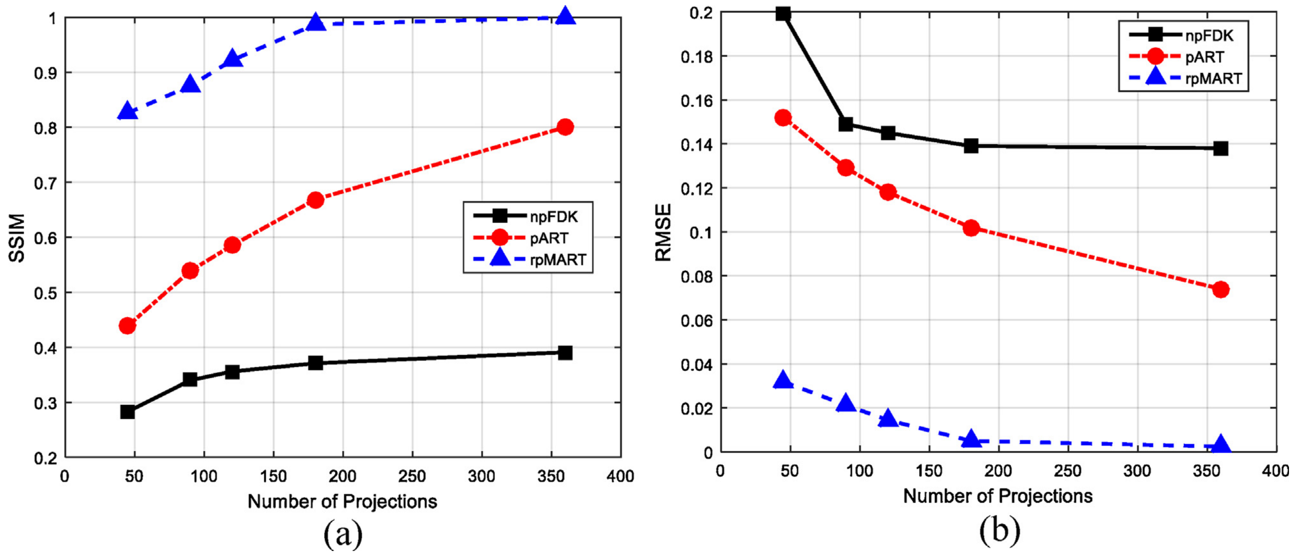

Fig. 8 – Performance of pART, rpMART and npFDK in terms of (a) SSIM and (b) RMSE vs. different number of projections data.

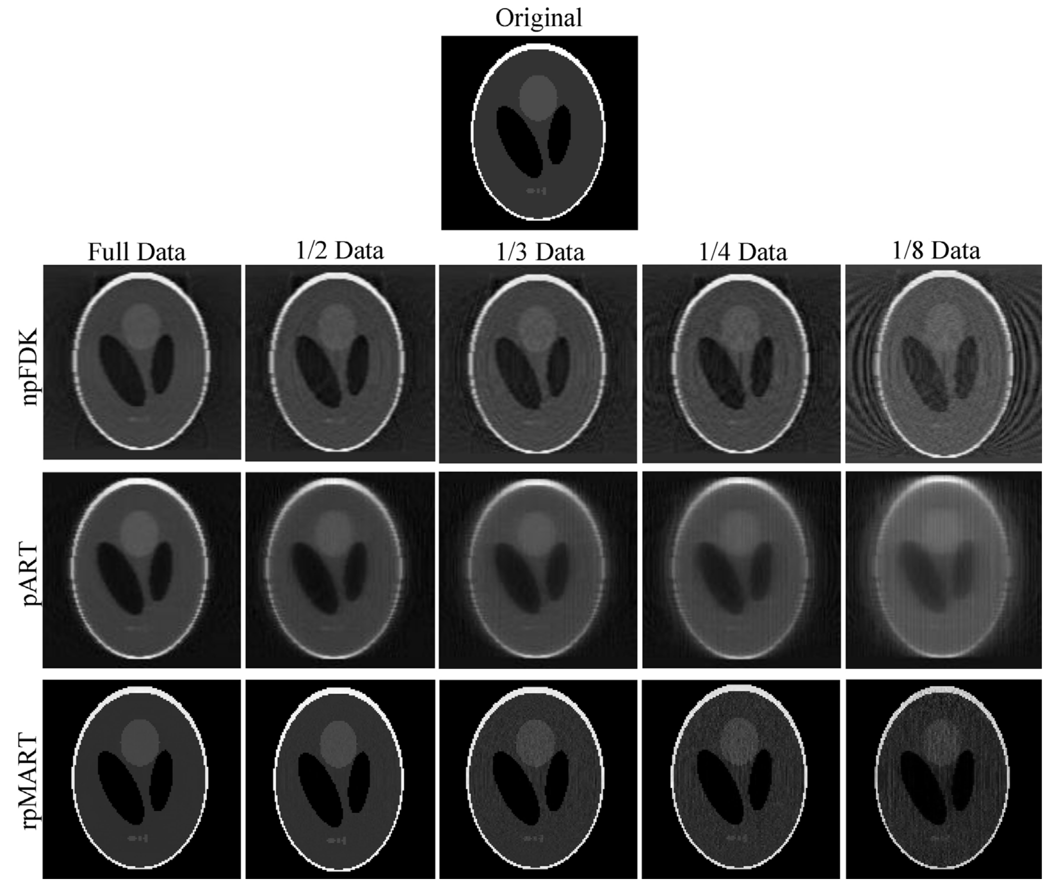

Fig. 9 – Effect of the limited number of projections data (360, 180, 120, 90, 45) with npFDK, pART and rpMART methods using 10 iterations.

and backward projections. Clearly from these two plots in Fig. 8, slightly changes are occurred on SSIM and RMSE from the full to 1/4 projections data in the case of rpMART and npFDK, but the significant differences are shown with the 1/8 of the full data.

For example, in the case of using the 1/8 projections data, rpMART provides higher image quality index of 0.83 than pART and npFDK methods which are 0.44 and 0.28, respectively.

It is shown in Fig. 9 where the improved image quality is achieved with 360 projections in comparison to the less

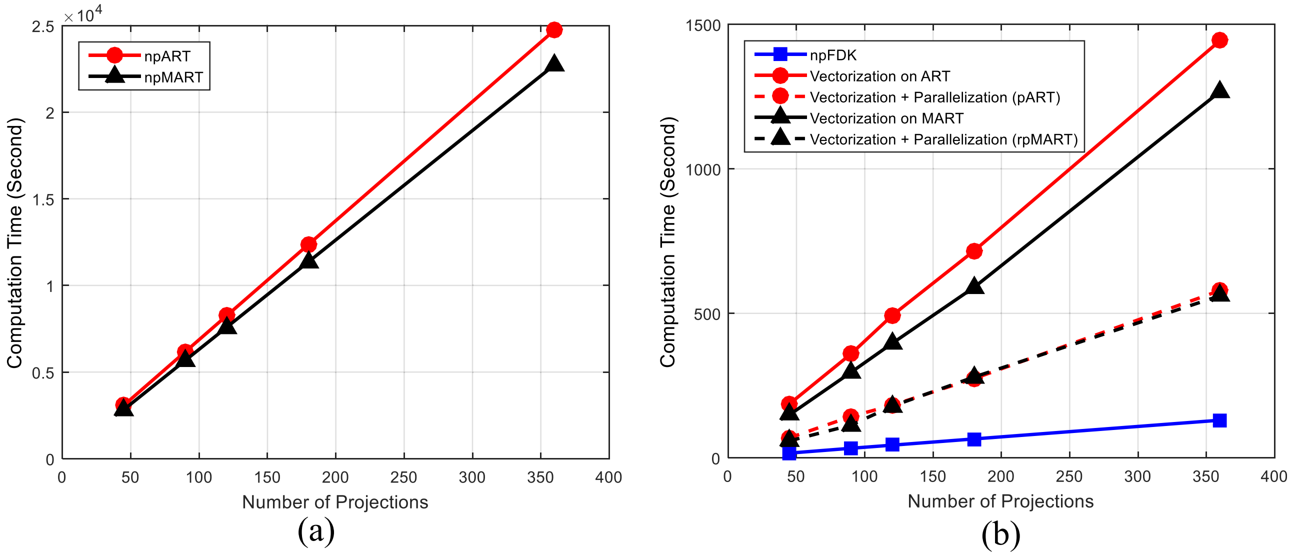

Fig. 10 – Computation time for (a) non-parallel npART and npMART, and (b) parallel computing pART and rpMART (i.e., vectorization and parallelization) against npFDK.

Table 2 – The computation time of the parallel and non-parallel ART and MART.

Methods	Computation time per one projection (s)		
	Non-parallel	After vectorization	After parallel computing
ART	68.7	4.01	1.61
MART	63.14	3.52	1.56

number of projections via npFDK, pART, or rpMART. Also, we clearly observe that rpMART provides higher images quality compared with pART and npFDK as shown in Fig. 9. The results via rpMART also significantly preserve the edges even at the lower amount of projection data are utilized. In contrast, pART and npFDK fail when utilizing the limited amount of projection data due to the produced images are blurred and deformed as visible in Fig. 9.

3.2.4. *Effect of the parallel computing*

The computation time of non-parallel iterative methods take too much time which are 6.87 h for npART and 6.314 h for npMART for the full projections data. Fig. 10(a) represents the computation time of npART and npMART at different number of projections. Considering the full projections data, vectorization process reduces the computation time to 1443.6 and 1267.2 s by pART and rpMART, repectively. Meanwhile, the partitioning parallelization process also contributed to reduce the computation time by pART from 1443.6 to 580 s and from 1267.2 to 560 s in the case of rpMART as shown in Fig. 10(b). Thus, the computation time of pART is significantly reduced with 17.13 and 2.49 times via both vectorization and partitioning parallelization processes, respectively. In the case of rpMART, the computation time is reduced with 17.94 and 2.26 times via both parallel computing processes, respectively. Although the partitioning parallelization process reduces the computation time by more than half using multi-core processing, major time reduction is occurred by vectorization process. Therefore, the total time reduction via the proposed parallel computing processes is 42.64 and 40.59 times for pART and rpMART, respectively. As shown in Fig. 10(b), npFDK has the lowest time computation against pART and rpMART with the full projections data, but it still has a limitation in the image quality as we mentioned above. The impact of our proposed parallel computing algorithm on the computation time of both the pART and rpMART methods are very significant and make these methods as techniques of choice in clinical applications. The computation time per one projection of data for the non-parallel npART and npMART compared with the parallel techniques is reported in Table 2. This comparison shows the amount of reduction in the computation time via the parallelized versions of pART and rpMART against the non-parallelized ART and MART. In the partitioning parallelization process, the ideal theoretical limit for speeding up, using only four processor cores, must not exceed 4 times. Our explanation above clarifies that the speed up due to the partitioning parallelization process contributes only by 2.49 or 2.26 times in pART and rpMART which are both clearly less than 4 times. Finally, the performance of the computation time via the proposed rapid Algebraic Reconstruction Techniques against some other previous works is summarized in Table 3. Our work shows its capability to speed up the reconstruction process.

4. Conclusion

We have presented a rapid version of MART via parallel computing (i.e., rpMART) in this paper and presented its performance against several conventional reconstruction techniques for CBCT. The results indicate that the reconstructed volume produced from rpMART shows the best image quality and lower errors at much reduced computation time. Even with the reduced amount of projection data, rpMART exhibits a better performance compared with pART and npFDK. Since the reduced projection data indicates the less scan time: this also implies the reduction of radiation doses. Hence, the results of this work provide a better way of image reconstruction with higher quality, fast computation time, and less projection data is required to reduce the radiation dose. The presented rpMART could be a better image reconstruction technique for CBCT in clinical applications. We believe that our proposed vectorization process plays a significant role in speeding up the reconstruction computation time. Also the parallelization via partitioning provided in this work shows the ability of our proposed vectorization process to work with multi-processors. Further works will involve adapting our vectorization process of rpMART with GPU processors in order to achieve much faster approach. Also its applicability with real data will be investigated for clinical applications in the future.

Table 3 – Comparison of the performance of our proposed rpMART against previous works (the computation time is calculated for one iteration with one projection).

Reference	Reconstruction method	Hardware specs	Computation time (s)
Qiu et al. [8]	ART	CPU, 64 bit 3.33 GHz Linux, 32 GB RAM	4.44
Fan and Xie [9]	ART	CPU NVIDIA Tesla C1060 GPU, 768 processor cores	40 2
Proposed technique	pART rpMART	CPU, 64 bit 2.2 GHz, 6 GB RAM, four processor cores	1.61 1.56

Algorithm 1. Algorithm of the rpMART

```
1:  Input: projection data
2:  Output: 3D reconstructed volume X
3:  3D volume initialization
4:  For all iterations k, until convergence
5:    Parfor projection angle $\theta_{1\to 4}$ // parallelization process for each partition
6:      For each 2D detector element that the ray hits // convert it into one vector
7:        Determine only the voxels that each ray passes through ($x_{m:1\to L}$)
8:        Calculate projection $\left(\sum_{m=1}^{L} w_{im} x_m^{(k)}\right)$ // (see Fig. 1)
9:        Correction/ update $:= \frac{\text{measured projection}\,(P_i)}{\sum_{m=1}^{L} w_{im} x_m^{(k)}}$
10:       // Back projection (for each voxel j : 1 → n)
11:       $x_j^{(k+1)} := x_j^{(k)}$; if $w_{ij} = 0$,   // vectorization process (line 8:10)
12:       $x_j^{(k+1)} := x_m^{(k)} * (\text{update})^{\lambda}$; if $w_{ij} \neq 0$ and $j = m$
13:     End
14:   End
15: End
```

Conflict of interest statement

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgement

This work was supported by a Grant from Kyung Hee University in 2017 (KHU-20170183).

REFERENCES

[1] Scarfe WC, Farman AG, Sukovic P. Clinical applications of cone-beam computed tomography in dental practice. J Can Dent Assoc 2006;72(1):75–80.

[2] Xiao S, Bresler Y, Munson DC. Fast Feldkamp algorithm for cone-beam computer tomography. Proceedings of the IEEE International Conference on Image Processing; 2003. p. 819–22.

[3] Feldkamp L, Davis L, Kress J. Practical cone-beam algorithm. J Opt Soc Am A 1984;1(6):612.

[4] Mueller K, Yagel R, Wheller J. Fast implementations of algebraic methods for three-dimensional reconstruction from cone-beam data. IEEE Trans Med Imag 1999;18 (6):538–48.

[5] Avilés JE. Multiplicative algebraic reconstruction techniques for 3D cone beam CT of the breast.Engineering Physics Thesis Yucatan, Mexico: Faculty of Engineering, Autonoma University; 2004.

[6] Cengiz K, Kamasak M. Comparison of algebraic reconstruction algorithms for tomosynthesis.MSc. Thesis Turkey: Computer Engineering Department, Faculty of Engineering, Istanbul Technical University; 2014.

[7] Fu J, Hu X, Velroyen A, Bech M, Jiang M, Pfeiffer F. 3D algebraic iterative reconstruction for cone-beam X-ray differential phase-contrast computed tomography. PLOS ONE 2015;10(3). http://dx.doi.org/10.1371/journal.pone.0117502

[8] Qiu W, Tong J, Mitchell C, Marchant T, Spencer P, Moore C, et al. New iterative cone beam CT reconstruction software: parameter optimisation and convergence study. Comput Methods Programs Biomed 2010;100(2):166–74.

[9] Fan Z, Xie Y. A block-wise approximate parallel implementation for ART algorithm on CUDA-enabled GPU. Bio-Med Mater Eng 2015;26(s1):S1027–35.

[10] Zeng G. Medical image reconstruction. Beijing: Higher Education Press; 2010.

[11] Siddon R. Fast calculation of the exact radiological path for a three-dimensional CT array. Med Phys 1985;12(2):252.

[12] Jacobs F, Sundermann E, Sutter BD, Lemahieu I. A fast algorithm to calculate the exact radiological path through a pixel or voxel space. J Comput Inf Technol 1998;6:89–94.

[13] Nguyen V, Lee S. Parallelizing a matched pair of ray-tracing projector and backprojector for iterative cone-beam CT reconstruction. IEEE Trans Nucl Sci 2015;62(1):171–81.

[14] Buzug T. Computed tomography. Berlin: Springer; 2008.

[15] Scherl H. Evaluation of state-of-the-art hardware architectures for fast cone-beam CT reconstruction. Wiesbaden: Vieweg+Teubner Verlag; 2011.

[16] Mori S, Endo M, Komatsu S, Kandatsu S, Yashiro T, Baba M. A combination-weighted Feldkamp-based reconstruction algorithm for cone-beam CT. Phys Med Biol 2006;51(16):3953–65.

[17] Simon R, David S, Laurent D. Comparison of analytic and algebraic methods for motion-compensated cone-beam CT reconstruction of the thorax. IEEE Trans Med Imag 2009;28 (10):1513–25.

[18] Xu F, Mueller K. Real-time 3D computed tomographic reconstruction using commodity graphics hardware. Phys Med Biol 2007;52(12):3405–19.

[19] Rezvani N, Aruliah D, Jackson K, Moseley D, Siewerdsen J. OSCaR: an open-source cone-beam CT reconstruction tool for imaging research. Med Phys 2007;34:2341.

[20] Chlewicki W, Badea C, Pallikarakis N. Cone based 3D reconstruction: a FDK – SART comparison for limited number of projections. Proceedings of the 9th Mediterranean Conference on Medical and Biological Engineering and Computing; 2001. p. 495–7.

[21] Mueller K, Yagel R. Rapid 3-D cone-beam reconstruction with the simultaneous algebraic reconstruction technique (SART) using 2-D texture mapping hardware. IEEE Trans Med Imag 2000;19(12):1227–37.

[22] Mian-Yi C, Yong R, Peng F, Peng H, Lu-Zhen D, Biao W, Zhi-Chao L, Jian W. Computed tomography image reconstruction from few-views data by multi-directional total variation. J Med Imag Health Inform 2015;5(2):309–16.

[23] Gordon D. Parallel ART for image reconstruction in CT using processor arrays. Int J Parallel Emerg Distrib Syst 2006;21 (5):365–80.

[24] Saha S, Tahtali M, Lambert A, Pickering M. CT reconstruction from simultaneous projections: a step towards capturing CT in One Go. Comput Methods Biomech Biomed Eng: Imag Vis 2015;1–13.

[25] Gordon R, Bender R, Herman G. Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography. J Theor Biol 1970;29 (3):471–81.

[26] Atkinson CH, Soria J. Algebraic reconstruction techniques for tomographic particle image velocimetry. Proceedings of the 16th Australasian Fluid Mechanics Conference; 2007. p. 191–8.

[27] Zeng K, Bai E, Wang G. A fast CT reconstruction scheme for a general multi-core PC. Int J Biomed Imag 2007;1–9. Article ID 29610.

[28] MATLAB version 8.5.0.197613 (R2015a). Natick, MA: The MathWorks Inc.; 2015.

[29] Loizou C, Pattichis C. Despeckle filtering algorithms and software for ultrasound imaging. San Rafael, CA: Morgan & Claypool Publishers; 2008.

[30] Wang Z, Bovik A, Sheikh H, Simoncelli E. Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 2004;13(4):600–12.