Received June 23, 2014, accepted July 8, 2014, date of publication July 17, 2014, date of current version August 12, 2014.

*Digital Object Identifier 10.1109/ACCESS.2014.2340372*

# GPU-Based Acceleration for Interior Tomography

**RUI LIU[1,2], YAN LUO[3], AND HENGYONG YU[1,2]**

[1]Department of Biomedical Engineering, Wake Forest University Health Sciences, Winston-Salem, NC 27157, USA
[2]Virginia Tech-Wake Forest University School of Biomedical Engineering and Sciences, Winston-Salem, NC 27157, USA
[3]Department of Electrical and Computer Engineering, University of Massachusetts, Lowell, MA 01854, USA

Corresponding author: H. Yu (hengyong-yu@ieee.org)

This work was supported in part by the National Science Foundation (NSF) CAREER under Award CBET-1149679 and in part by NSF Collaborative under Grant DMS-1210967.

**ABSTRACT** The compressive sensing (CS) theory shows that real signals can be exactly recovered from very few samplings. Inspired by the CS theory, the interior problem in computed tomography is proved uniquely solvable by minimizing the region-of-interest's total variation if the imaging object is piecewise constant or polynomial. This is called CS-based interior tomography. However, the CS-based algorithms require high computational cost due to their iterative nature. In this paper, a graphics processing unit (GPU)-based parallel computing technique is applied to accelerate the CS-based interior reconstruction for practical application in both fan-beam and cone-beam geometries. Our results show that the CS-based interior tomography is able to reconstruct excellent volumetric images with GPU acceleration in a few minutes.

**INDEX TERMS** Computed tomography, compressed sensing, parallel computing, graphics processing unit, interior tomography.

## I. INTRODUCTION

The x-ray computed tomography (CT) has been an indispensable imaging modality relying on multiple x-ray projections of the subject to reconstruct a two dimensional (2D) or three dimensional (3D) distribution of the attenuation coefficients within the subject [1]. Although it is proud of high spatial and temporal resolution [2], CT radiation accounts for a large portion of the ionizing radiation to the population that cannot be underestimated. The patients undergo a CT scan were estimated to be 60 million in 2002 in the United States, which occupied nearly 75% of the radiation exposure and almost 15% of the imaging procedures [3]. The widely accepted As Low As Reasonably Achievable (ALARA) principle urges the medical community to reduce the unnecessary radiation hazard as much as possible [4], [5]. Reducing the x-ray flux towards the detector and decreasing the amount of x-ray paths across the whole object supporting are two common strategies to avoid extra radiation. The former is usually achieved by controlling the operating current, voltage or exposure time but leading to high projection noise and the latter may produce few-view, limited-angle or truncated projection problems [6].

Conventional iterative reconstruction algorithms are immune to noisy projections in some extent. However, for the highly ill-posed reconstruction problem, additional regularization is necessary for unique and stable solutions. Interestingly, the compressive sensing (CS) theory shows that a real signal can be accurately recovered with an overwhelming probability from the amount of data or measurements far less than the Shannon-Nyquist sampling theorem claimed [7], [8]. The $l_0$ norm minimization is the basic paradigm of CS based signal recovering. Because of the NP-hard characteristic of the $l_0$ norm minimization, it is usually relaxed to the $l_1$ norm optimization with solid theoretical supports [9], [10]. Many algorithms have been proposed to solve the $l_1$ norm optimization problem, such as interior point method [11], gradient projection method [12] and some dedicated algorithms for the CS-based optimization [16]. In the medical imaging field, the total variation (TV, $l_1$ norm of discrete gradient transform (DGT) of an image) has been widely adopted as a regularization item. The conventional simultaneous algebraic reconstruction technique (SART) framework can be applied for CS-based image reconstruction by adding the TV regularization item [14]–[16]. The TV minimization is achievable either by using the steepest decent (SD) method or by using the soft-threshold filtering (STF) method. Many other TV minimization methods or applications were also reported, such as PICCS (prior image constrained compressed sensing) algorithm [17], CS based interior tomography in SIR (statistical iterative reconstruction) [18], and improved TV method in an ASD-POCS framework [19].

Narrowing down the x-ray beam to focus on a region-of-interest (ROI) is a representative method for dose reduction nominated as interior scan and the corresponding reconstruction is the interior problem. Because the interior problem is generally non-uniquely solvable, the interior scanning is unable to be applied for quantitative analysis based applications in clinics. However, limited by the high resolution detector size and the radiation dose reduction expectation, the interior scan is commonly desirable for many practical applications, such as cardiac CT [20] and Nano-CT [4]. Inspired by the CS theory, the interior problem has been proved uniquely and stably solvable by minimizing the ROI's TV provided that the imaging object inside the ROI is piecewise constant [21], [22]. This knowledge regularized CT reconstruction algorithm is called CS-based interior tomography and the key is to minimize the TV of the ROI inside the imaging object.

The high computational cost (arithmetic operation and memory bandwidth) of the CS-based signal recovering algorithms hinders the sequential implementations of the iterative algorithms been applied in clinics especially for cone-beam spiral CT reconstruction [16]. Both the projection and backprojection processes are categorized as single instruction multiple data (SIMD) [23] computing model. The SIMD model is quite suitable for parallelization without too much complicated communication, synchronization or mutual lock mechanisms. Initially, parallel image reconstruction algorithms were implemented on clusters [24]. Cell processors were also applied in general purpose parallel computing [25]. Very recently, Intel released the Intel MIC (Intel Many Integrated Core) architecture products [26], and software engineers can run their codes on MIC with little or no additional workload. In early years, researchers began to accelerate their algorithms with graphics processing unit (GPU) when its programming interface was published for general purpose computing. Before GPU was programmable for general purpose computing, general algorithms were dedicatedly camouflaged as graphical operations such as texture mapping for parallel acceleration. Analytical reconstruction algorithms were benefited enormously from GPU [27]. Nowadays, two groups of programming interfaces for general computing in GPU have been developed. One is based on computer graphics languages, such as OpenGL [28], CG [29], HLSL [30], *etc*. The other is dedicated for high performance GPU computing, such as CUDA [23], OpenCL [31], Brook [32], *etc*. The CUDA (Compute Unified Device Architecture) [33]–[35] is rapidly exploited in many fields including medical imaging [35]. Comparing with other general GPU computing interfaces, the CUDA has higher performance and is easier to master and more flexible. Therefore, we choose CUDA to implement the CS-based interior reconstruction in GPU. For higher performance, the shared memory, texture memory and constant memory are applied for their high bandwidth and caching mechanism in our implementation.

In this paper, the SART and ordered-subset (OS)-SART reconstruction frameworks with TV minimization are implemented in GPU computing to make the CS-based interior tomography practical. In section II, the CT imaging model is reviewed with a concise SART-type reconstruction framework and the STF method. In section III, the implementation details are given. Section IV demonstrates the numerical results in 2D fan-beam and 3D cone-beam geometries. The SD and STF based TV minimization methods are compared in terms of reconstruction accuracy and speed. In section V, we discuss some related issues and conclude this paper.

## II. ALGORITHM DESIGN

### A. IMAGING MODEL

A 2D or 3D digital image can be expressed as $\boldsymbol{f} = (f_{i,j,k}) \in \mathbb{R}^N$, where $N = I \times J$ for a 2D image and $N = I \times J \times K$ for a 3D image. $I$, $J$ and $K$ are image pixel number in length, width and height dimensions. In this paper, both $f_{i,j,k}$ and $f_n$ are applied for convenience. Therefore, a CT system can be modeled as

$$\boldsymbol{p} = \boldsymbol{A}\boldsymbol{f}. \tag{1}$$

Each component of the vector $\boldsymbol{p} \in \mathbb{R}^M$ is a measured datum, where $M$ is the total measurements (the product of the projection number and the detector cell number), and $\boldsymbol{A} \in \mathbb{R}^M \times \mathbb{R}^N$ is the system matrix. Typically, the n$^{\text{th}}$ pixel is viewed as a rectangular area with a constant value $f_n$, and the m$^{\text{th}}$ projection datum $p_m$ can be viewed as the summation of all the weighted pixel values involving the m$^{\text{th}}$ x-ray. Lots of discrete models have been proposed to calculate the entries of $\boldsymbol{A}$, such as linear interpolation method, grid method, distance-driven method [36], Siddons' method [37], area integral method [38], footprint method [39], etc. To balance the speed and accuracy, the Siddons' method is adopted as the projection model and the pixel-driven method is adopted as the backprojection model in our experiments. Siddons' algorithm takes the length of the m$^{\text{th}}$ x-ray penetrating the n$^{\text{th}}$ rectangular pixel/voxel as the element $a_{m,n}$ of the system matrix $\boldsymbol{A}$. An additive noise $\boldsymbol{e} \in \mathbb{R}^N$ is assumed, and the imaging process is finally modeled as

$$\boldsymbol{p} = \boldsymbol{A}\boldsymbol{f} + \boldsymbol{e}. \tag{2}$$

### B. SART AND OS-SART

The SART-type solution for Eq. (2) is expressed as [40]

$$f_n^{(l+1)} = f_n^{(l)} + \frac{\lambda^{(l)}}{a_{+n}} \sum_{m=1}^{M} \frac{a_{m,n}}{a_{m+}} \left(p_m - \boldsymbol{A}_m \boldsymbol{f}^{(l)}\right), \tag{3}$$

where $a_{+n} = \sum_{m=1}^{M} a_{m,n} > 0$, $a_{m+} = \sum_{n=1}^{N} a_{m,n} > 0$, $\boldsymbol{A}_m$ is the m$^{\text{th}}$ row of $\boldsymbol{A}$, $l$ is the iteration index and $0 < \lambda^{(l)} < 2$ is a free relaxation parameter. For simplicity, let $\boldsymbol{\Lambda}^{+N} \in \mathbb{R}^N \times \mathbb{R}^N$ be a diagonal matrix with $\Lambda_{n,n}^{+N} = \frac{1}{a_{+n}}$ and $\boldsymbol{\Lambda}^{M+} \in \mathbb{R}^M \times \mathbb{R}^M$ be a diagonal matrix with $\Lambda_{m,m}^{M+} = \frac{1}{a_{m+}}$, Eq. (3) is rewritten as

$$\boldsymbol{f}^{(l+1)} = \boldsymbol{f}^{(l)} + \lambda^{(l)} \boldsymbol{\Lambda}^{+N} \boldsymbol{A}^T \boldsymbol{\Lambda}^{M+} \left(\boldsymbol{p} - \boldsymbol{A}\boldsymbol{f}^{(l)}\right). \tag{4}$$

The ordered subset (OS) approach accelerates the convergence of SART by one to two orders of magnitude with the cost of inducing image bias [41], [42]. The projection data is divided into $T$ disjoint subsets $B_t = \left\{ i_1^t, \ldots, i_{B(t)}^t \right\}$, where $(t = 1, 2, \ldots, T)$, is the subset index of the projection. The union of these subsets covers the whole projection set. We have $B_i \cap B_j = \emptyset \, (i \neq j, i, j \in \{1, 2, \ldots, T\})$ and $\bigcup_{t=1}^{T} B_t = \{1, 2, \ldots, M\}$. These subsets alternatively participate the iterations. The OS-SART is represented as [35]

$$f_n^{(l+1)} = f_n^{(l)} + \lambda^{(l)} \sum_{m \in B_t} \frac{a_{m,n}}{\sum_{m' \in B_k} a_{m',n}} \frac{(p_m - \boldsymbol{A}_m \boldsymbol{f}^{(l)})}{a_{m+}}, \quad (5)$$

where $t = (l \bmod T) + 1$. For every sub step in one iteration, it is suggested that the selected subset should be with the greatest possible angular distance from the previously used subset [41]. In our applications in this paper, the subset size is set as the detector cell number in one view. Meanwhile, the fast weighting technology in the FISTA (fast iterative shrinkage-thresholding algorithm) is applied to further accelerate the SART-type algorithms with a constant step size [43].

### C. CS-BASED IMAGE RECONSTRUCTION

To further reduce the projections, the CS-based signal reconstruction method is combined with the OS-SART. The paradigm for CS-based signal recovering is a constrained $l_0$ norm minimization problem defined as

$$\hat{\boldsymbol{x}} = \arg_{\boldsymbol{x}} \min \|\boldsymbol{x}\|_0, \quad s.t. \, \boldsymbol{y} = \boldsymbol{\Phi x}, \quad (6)$$

where $\boldsymbol{x}$ is a sparse signal, $\boldsymbol{y}$ is the observed data and $\boldsymbol{\Phi}$ is the sensing matrix. To address the NP-hard problem and suppress noise, Eq. (6) is usually modified as

$$\hat{\boldsymbol{x}} = \arg_{\boldsymbol{x}} \min \|\boldsymbol{x}\|_1, \quad s.t. \, \|\boldsymbol{y} - \boldsymbol{\Phi x}\|_2^2 \leq \varepsilon, \quad (7)$$

where $\varepsilon$ is the measurement error. For a sparser solution, the $l_p$ norm minimization is also investigated. Generally speaking, the smaller the $p$ is, the less measurements are needed for accurate reconstruction [44]. Actually, most of the signals in reality are far-fetched sparse, and the $l_1$ norm minimization paradigm is prohibited to be applied straightforwardly in medical image reconstruction. Usually, a sparse transform will be employed first to transform the non-sparse signal to an appropriate sparse domain. Assuming $\boldsymbol{\Psi}$ being the sparse transform, the CS-based CT image reconstruction paradigm can be finally expressed as

$$\hat{\boldsymbol{f}} = \arg_{\boldsymbol{f}} \min \|\boldsymbol{\Psi f}\|_1, \quad s.t. \, \|\boldsymbol{p} - \boldsymbol{Af}\|_2^2 \leq \varepsilon. \quad (8)$$

### D. DGT-BASED SPARSITY AND STF

The objective function $\|\boldsymbol{\Psi f}\|_1$ in Eq. (8) can be defined as $\|\boldsymbol{f}\|_{TV}$ if $\boldsymbol{\Psi}$ is the discrete gradient transform (DGT) and the image object satisfy the piecewise constant assumption, where $\|\cdot\|_{TV}$ denotes the $l_1$ norm of the DGT. The so-called Neumann condition [45] on the boundary is also assumed. With an appropriate Lagrange multiplier $\sigma$, the problem Eq. (8) can be rewritten as:

$$\hat{\boldsymbol{f}} = \arg_{\boldsymbol{f}} \min \left( \|\boldsymbol{p} - \boldsymbol{Af}\|_2^2 + \sigma \, \|\boldsymbol{f}\|_{TV} \right). \quad (9)$$

Because the two items in Eq. (9) are convex, they can be alternatively minimized to yield an accurate solution. While the OS-SART can be used to minimize $\|\boldsymbol{p} - \boldsymbol{Af}\|_2^2$, the conventional SD method can be employed to minimize the TV term. However, the SD based TV minimization sometimes over-enhances the edge regions and generates Gibbs-effects-like artifacts, and the step size needs to be carefully selected to find the minimum value and guarantee fast convergence. The STF (soft threshold filtering) is an alternative choice to minimize TV. Let $Z = \left\{ \boldsymbol{\zeta}_\gamma \right\}_{\gamma \in \Gamma}$ be a basis in $\mathbb{R}^N$. $\boldsymbol{f}$ can be linearly expressed as $\boldsymbol{f} = \sum_{\gamma \in \Gamma} \langle \boldsymbol{f}, \boldsymbol{\zeta}_\gamma \rangle \boldsymbol{\zeta}_\gamma$. Let us define an objective function with positive weights $\Omega = \left\{ \omega_\gamma \right\}_{\gamma \in \Gamma}$

$$\Xi_{\Omega,q}(\boldsymbol{f}) = \|\boldsymbol{p} - \boldsymbol{Af}\|_2^2 + \sum_{\gamma \in \Gamma} 2\omega_\gamma \left| \langle \boldsymbol{f}, \boldsymbol{\zeta}_\gamma \rangle \right|^q, \quad q \in [0, 2]. \quad (10)$$

When $q = 1$, finding the sparest solution of $\Xi_{\Omega,1}$ is equivalent to Eq.(9) for DGT [14]. To find the sparse solution of $\Xi_{\Omega,1}$, we can recursively minimize $\Xi_{\Omega,1}$ in a STF framework

$$\hat{\boldsymbol{f}}^{(l+1)} = \boldsymbol{S}_{\Omega,1} \left( \hat{\boldsymbol{f}}^{(l)} + \boldsymbol{A}^{\boldsymbol{T}} \left( \boldsymbol{p} - \boldsymbol{A} \hat{\boldsymbol{f}}^{(l)} \right) \right), \quad (11)$$

where $\hat{\boldsymbol{f}}^{(l)}$ is the intermediate image and

$$\boldsymbol{S}_{\Omega,1}(\boldsymbol{x}) = \sum\nolimits_{\gamma \in \Gamma} S_{\omega_\gamma,1} \left( \langle \boldsymbol{x}, \boldsymbol{\zeta}_\gamma \rangle \right) \boldsymbol{\zeta}_\gamma \quad (12)$$

is the functional that performs a soft threshold filtering. It has been proved by Daubechies *et al* in [15] that Eq.(11) is convergent. In Eq.(12), $S_{\omega_\gamma,1}$ is defined as

$$S_{\omega_\gamma,1}(x) = \begin{cases} x - \omega_\gamma, & if \ x \geq \omega_\gamma \\ 0, & if \ |x| < \omega_\gamma \\ x + \omega_\gamma, & if \ x \leq -\omega_\gamma. \end{cases} \quad (13)$$

However, because the DGT is non-invertible and violates the restricted isometrics property (RIP [7], [46]), the STF method is prohibited to be directly applied for TV minimization. This problem can be addressed by constructing a pseudo inverse of DGT [14], [16].

## III. GPU ACCELERATION

### A. PARALLELIZATION STRATEGY

For the CS-based interior tomography algorithm, the projection, backprojection, DGT, pseudo inverse transform, soft threshold filtering and finding the optimal threshold can be parallelized. Some constants (such as all the trigonometric values of the view angles, coordinates of the source position, aligned voxel coordinates and the detector coordinates in x, y and z directions, etc.) can be pre-calculated and symbolically mapped to constant memory to achieve higher caching efficiency. However, this symbolical linking is optional because it costs almost the same clock cycles as fetching the data from the constant memory to calculate the coordinates corresponding to the current voxel or detector cells on the fly.

The on-board device memory usually is sufficient for fan-beam reconstruction unless storing the whole system matrix $\boldsymbol{A}$. Sometimes, it is also possible to compactly store $\boldsymbol{A}$ in the device memory when the image and the projection data are not too large. We compactly stored a 180,000 × 65,536 sparse matrix $\boldsymbol{A}$ in COO (Coordinate list, which stores a list of (row, column, value) tuples) format in CPU. The system matrix is transfer to the CSR (Compressed Sparse Row) format after it is loaded into the device memory. The projection process occupies 6.23ms with cuSPARSE (NVIDIA CUDA Sparse Matrix library). Actually, the whole program spent 2.26 seconds including reading matrix to the main memory and transferring to the device memory, etc. Even though a large device memory is provided, the storing of system matrix has to be gingerly designed to be fully compacted and the computational efficiency should be kept. Assuming that the image to be reconstructed is square, if the symmetry of 2D/3D scanning is fully utilized; only the first eighth of the scanning angles needs to be considered when the scanning range is $[0, 2\pi]$. The symmetrical relationship between image indices and sinogram indices are defined as follows:

$$\begin{aligned}
(\text{i, j}) &\rightarrow (\theta, t)\\
(\text{R-j, i}) &\rightarrow (\theta + \pi/2, t)\\
(\text{R-i, R-j}) &\rightarrow (\theta + \pi, t)\\
(\text{j, R-i}) &\rightarrow (\theta + 3\pi/2, t)\\
(\text{R-j, R-i}) &\rightarrow (\pi/2 - \theta, T - t)\\
(\text{i, R-j}) &\rightarrow (\pi - \theta, T - t)\\
(\text{j, i}) &\rightarrow (3\pi/2 - \theta, T - t)\\
(\text{R-i, j}) &\rightarrow (2\pi - \theta, T - t)
\end{aligned} \tag{14}$$

where $i, j$ represents the pixel indices of the image, $\theta$ is the current projection angle that can be easily transferred to the angle index, $t$ is the index of the detector element and $T$ is the detector resolution, and $R$ is the image resolution in length or width direction. The pairs $(i, j)$ and $(\theta, t)$ satisfy the Radon transform relationship. In cone-beam geometry, not only the scanning range but also the projection symmetry on the upper and downer parts of the detector can be utilized if there is no offset. However, these symmetrical based tricks are not adopted here due to detector offset need to be considered in practical applications.

Typically, a sinogram is divided into adjoining sub-blocks in fan-beam case for parallel projection. Each sub-block corresponds to one thread block in CUDA and the threads in one block can be configured discretionarily. The thread block is divided into $(p_x \times p_y)$ threads, where $p_x$ addresses the detector indices and $p_y$ addresses the projection angle indices with built-in variables in CUDA. On one hand, sufficient threads are required to accurately reconstruct the image object. On the other hand, it wastes computational resource if more threads are allocated than what are needed. It is optimal that the detector cell and the projection angle numbers are divisible by $p_x$ and $p_y$, respectively. The thread number being multiple of 32 in one block is sensible because the GPU executes 32 threads as a warp simultaneously. Fully occupying 1024 threads in one block is not optimal in Fermi architecture. The maximum warps for one multiprocessor in Fermi architecture is 48 that implies 1536 threads can be executed simultaneously. If the threads block is configured with fully 1024 threads, 1/3 of the computation resource will be idle. In the Kepler architecture, the maximum warp number increases to 64 in one multiprocessor. Therefore, 2048 threads can be executed simultaneously. Without considering the backward compatible for earlier GPUs, 1024 threads fill up one threads block in Kepler architecture and 512 threads are configured in Fermi architecture.

### B. PROJECT AND BACKPROJECTION MODELS

Siddons' algorithm is adopted as the projection model. An element $a_{m,n}$ of system matrix $\boldsymbol{A}$ is computed as the length of m$^{\text{th}}$ x-ray passing through n$^{\text{th}}$ pixel/voxel for a 2D/3D image. In conventional CPU and GPU based implementations, additional memories are pre-allocated to store the parameters representing the intersections between an x-ray and the boundary box of each pixel/voxel. The pre-assigned memory should be at least $(\text{N}_x + \text{N}_y + \text{N}_z + 3) \times \text{S}_D$ bytes in cone-beam geometry, where $\text{N}_x$, $\text{N}_y$ and $\text{N}_z$ are the pixel numbers of the volume in length, width and height dimensions respectively, and $\text{S}_D$ is the data size of each voxel. This may cause trouble for the CUDA programming with GK104 chip because it is banned to dynamically allocate device memory in the executing kernel. It is required to recompile the program if the image resolution changes and it also has to be implemented with macros to allocate the constant memory for storing the parameters. Although the constant expression feature in C++11 standard can solve this problem with the key word "constexpr", this feature is not supported in Microsoft Visual Studio 2012 compiler. Therefore, a modified Siddons' algorithm without pre-assigning memory was applied [47]. This algorithm first calculates the intersections of the compact support of the 2D/3D image and the current x-ray path, which is similar to the clipping algorithm in computer graphics [48]. Beginning with the incident point, the algorithm calculates the intersection points for each pixel/voxel for its weighting coefficient $a_{i,j}$. Iteratively, the exit point of the current pixel/voxel is set as the incident point of its neighbor pixel/voxel and this process will be repeated until the current incident point is the exit point located on the boundary of the object. After the current intersection point arrives at the exit point on the compact support, the innate ordered intersection parameters are all generated and the weighting for these intersection voxels can be calculated.

The pixel/voxel driven model is adopted to implement backprojection operation. For a given projection angle, mapping a point in the world coordinate to its corresponding detector index can be simplified to a geometrical transform matrix $M$ in flat-panel detector case. This matrix can be constructed as follows. A homogeneous coordinate is assumed in our transform. The source and detector positions are both counterclockwisely rotated to make the source on the positive direction of Y axis with a rotation matrix $R_\theta$, and the source is moved to the origin of the world coordinate by multiplying a matrix $T_d$. After a perspective transform with $P_d$, the transformed object point is projected to its shadow on the

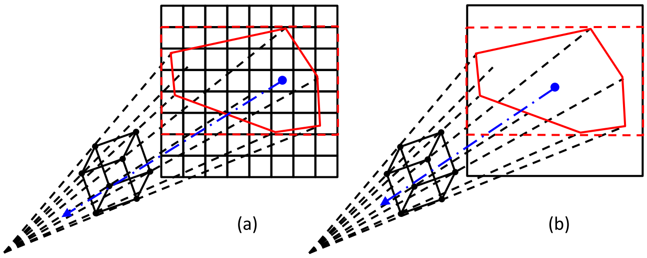


**FIGURE 1. Boundary box based backprojection process. (a) The backprojection is more accurate with higher computational cost when the detector size is small. (b) The backprojection degenerates to the pixel/voxel driven method with more computational cost when the detector size is large.**

detector. Finally, the shadow index on the detector can be easily calculated with two matrices $T_m$ and $S_u$. For concise, the offset of the detector, which can be easily integrated into the matrix multiplication in $T_m$ is not considered in this derivation. Therefore, the matrices chain can be expressed as (15) shown at the bottom of the page, where $u_x$ and $u_z$ are detector element size, $m_x$ and $m_z$ are minimum coordinate of the detector at the initial position, $D$ is the source to detector distance, $d$ is the source to iso-center distance and $\theta$ is the current view angle. Because the ray-driven projection model mismatches the pixel/voxel driven backprojection process which means the system matrix in backprojection is not exactly the transpose of the projection system matrix. This mismatch will induce artifact with the increase of iteration number. Therefore, a method similar to Li et al. [49] is adopted to make the projection and backprojection match. The projection of a voxel on a plane can only be a convex polygon such as hexagon, pentagon or quadrilateral. The ray passing through the voxel must be inside its convex polygon shadow. First, for every voxel the boundary convex polygon is calculated. Then, the rectangle boundary box is calculated from the minimum and maximum projection coordinates of the voxel. Only the detector cells inside the boundary rectangular are considered as demonstrated in Figure 1. However, if the boundary rectangular is not large enough to contain multiple detector cells, it has no advantage over the simple pixel driven method but needs more computational complexity. When the detector resolution in pitch direction is the same as or smaller than the object resolution in the height dimension, some of the pixels in different slices will never be penetrated by any cone-beam x-ray in projection process. This will result in serious artifacts. To suppress this artifact, when the detector resolution is insufficient for high resolution reconstruction, we have to compromise the backprojection model to solve the model mismatching problem. Due to the specificity of this algorithm, a single thread is mapped to the vertex of a voxel instead of a single voxel. It is readily appreciated that 8 adjacent voxels share one vertex in three directions, and the redundant coordinate computing will occur if one thread response for one voxel updating. Therefore, the shared memory is applied to minimize latency caused by reading the same data from global memory multiple times and unnecessary computing. The thread block for backprojection is allocated as (8, 8, 8) and the thread index addresses current vertex. Because only the first 7 indices in each dimension response to the voxel updating, each thread block can update 343 voxels.

### C. PARALLELIZATION OF TV MINIMIZATION

In 3D case, a general $l_p$ norm of DGT is defined as

$$L_p\left(D(f)\right) = \left(\sum_{k=1}^{K}\sum_{j=1}^{J}\sum_{i=1}^{I}\left|D\left(f_{i,j,k}\right)\right|^p\right)^{\frac{1}{p}}, \tag{16}$$

where

$$D(f_{i,j,k}) = \sqrt{(f_{i+1,j,k}-f_{i,j,k})^2+(f_{i,j+1,k}-f_{i,j,k})^2+(f_{i,j,k+1}-f_{i,j,k})^2}. \tag{17}$$

when $p = 1$, Eq.(16) degenerates to the TV. For an individual DGT value $D\left(f_{i,j,k}\right)$ at the position $(i, j, k)$, it involves four values $(f_{i+1,j,k}, f_{i,j+1,k}, f_{i,j,k+1}$ and $f_{i,j,k})$ which are sporadically stored in device global memory. Initially, this suggests each thread calculates one DGT value in position $(i, j, k)$ with a simple kernel function. Therefore, the threads configurations in fan-beam and cone-beam cases are the same as the configurations in the corresponding backprojection cases, respectively. In fact, the DGT is not intensive in arithmetic computation but in memory bandwidth. If all the data are stored in the global memory, every thread has to read 4 discontinuous data from the global memory and adjacent threads in one block have to read the same data which generate latency. Therefore, the volume is divided into overlapped subvolumes with size $8 \times 8 \times 8$. For one thread block, its indexed

$$\begin{aligned}
M &= S_u \cdot T_m \cdot P_d \cdot T_d \cdot R_\theta \\
&= \begin{bmatrix} 1/u_x & 0 & 0 \\ 0 & 1/u_z & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & -m_x \\ 0 & 1 & -m_z \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} -D & 0 & 0 & 0 \\ 0 & 0 & -D & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -d \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} (m_x \cdot \sin\theta - D\cos\theta)/u_x & -(\cos\theta \cdot m_x + D\sin\theta)/u_x & 0 & d \cdot m_x/u_x \\ m_z \cdot \sin\theta/u_z & -m_z \cdot \cos\theta/u_z & -D/u_z & d \cdot m_z/u_z \\ -\sin\theta & \cos\theta & 0 & -d \end{bmatrix},
\end{aligned} \tag{15}$$

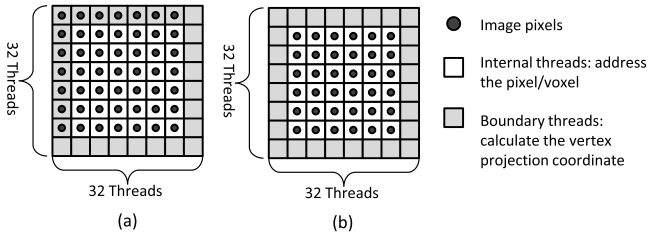

32 Threads
32 Threads
(a)
32 Threads
32 Threads
(b)
● Image pixels
□ Internal threads: address the pixel/voxel
▨ Boundary threads: calculate the vertex projection coordinate

**FIGURE 2. The thread configurations in fan-beam reconstruction. (a) represents one thread block configuration for backprojection and DGT. (b) represents one thread block configuration for TV descent direction calculation and pseudo-inverse of DGT.**

sub-volume is copied to the shared memory. In fan-beam case, similar strategies is applied, the image is divided into 32 × 32 overlapped sub-image, and for one thread block the corresponding sub-image is loaded into shared memory as shown in Figure 2. Because the value in last index of the sub image is the first index of its adjoining sub image, if the thread in each block is allocated as $(t_x, t_y)$ and the image size is $(N_x, N_y)$ in fan-beam case, the block number for DGT is $\left(\left\lfloor \frac{N_x+t_x-2}{t_x-1} \right\rfloor, \left\lfloor \frac{N_y+t_y-2}{t_y-1} \right\rfloor\right)$. Similarly, the block number in 3D case can be calculated using a similar formula.

Finding an optimal threshold after DGT is an indispensable step to accelerate the convergence of the STF, which can be achieved by dichotomy in GPU. The Thrust library is applied for this implementation [48]. We first copied the discrete gradient image $\hat{\boldsymbol{d}}^{(l)}$ to $\hat{\boldsymbol{d}}_c^{(l)}$ and then $\hat{\boldsymbol{d}}_c^{(l)}$ is ascending sorted. The prior knowledge of the intermediate discrete gradient image can be estimated from the roughly reconstructed image by the classical FBP approach. The soft-threshold filtering is applied with respect to each component according to Eq. (13). The pseudo inverse transform will be applied to inversely transform the image from the DGT to image domain, which is implemented in GPU according to Eqs. (3.8) to (3.11) in [14]. The memory bandwidth problem can also be solved by the shared memory similar to the DGT implementation. Different from the DGT transform, the inverse threshold filtering needs more shared memory. In 2D case, the thread block is also configured with size (32, 32). Considering that the inverse transform in position $(i, j)$ relates to all its adjacency pixels, only 30 × 30 pixels in the center of the sub image can be calculated by the current thread block as in Figure 2. Therefore the block number should be $\left(\left\lfloor \frac{N_x+t_x-3}{t_x-2} \right\rfloor, \left\lfloor \frac{N_y+t_y-3}{t_y-2} \right\rfloor\right)$ if the image resolution is $N_x \times N_y$ and the threads configuration in one block is $(t_x, t_y)$. The block number for 3D pseudo inverse transform can be calculated in a similar fashion.

### D. OVERALL PSEUDO-CODES

Combining the OS-SART and STF, the CS-based interior tomography can be implemented as the following pseudo-codes:

```
S0: Initializing $l = 0$, $\hat{\boldsymbol{f}}^{(l)} = 0$ and estimating $\omega_{\gamma_0}$;
S1: Updating $l \leftarrow l + 1$ and performing OS-SART to
    update $\hat{\boldsymbol{f}}^{(l)}$;
S2: Performing DGT from $\hat{\boldsymbol{f}}^{(l)}$ to $\hat{\boldsymbol{d}}^{(l)}$ corresponding
    to $\langle \hat{\boldsymbol{f}}, \boldsymbol{\zeta}_\gamma \rangle$ in Eq.(13);
S3: Performing soft-threshold filtering from $\hat{\boldsymbol{d}}^{(l)}$ to $\tilde{\boldsymbol{d}}^{(l)}$
    using Eq. (14);
S4: Performing pseudo inverse DGT to obtain $\tilde{\boldsymbol{f}}^{(l)}$; [14]
S5: Updating $\hat{\boldsymbol{f}}^{(l)} \leftarrow \tilde{\boldsymbol{f}}^{(l)}$;
S6: If the stopping criteria are met, output the result;
    otherwise go to S1.
```

## IV. NUMERICAL EXPERIMENTS

### A. PLATFORM CONFIGURATION AND GEOMETRY PARAMETERS

The SART and OS-SART algorithms are implemented in GPU for both fan-beam and cone-beam geometries with CUDA/C++ language. For the CS-based interior tomography, the TV regularization is applied with the aforementioned SD and STF methods. All the experiments are tested on a high-performance workstation configured as follows. Two Intel Xeon CPUs are configured with core clock frequency 3.10GHz. Each CPU contains 16 cores. The memory size is 32GBs. The operating system is Microsoft Windows 7 (64-bits Professional version). For GPU computing, NVIDIA Tesla K10 is used including 2 GK104s. Each GPU contains 1536 CUDA cores, and the GPU clock frequency is 745MHz. The device memory clock frequency is 2500MHz and the bus width is 256 bit. The device memory for each GPU is 3GB.

For cone-beam reconstruction, the system geometry is configured as in Table 1. The system geometry in fan-beam reconstruction is the same as the central slice parameters in cone-beam geometry. In table 1, CATCE means that the value will be different in different groups of experiments and it will be clarified in the experiments. In the numerical simulations, Poisson noise is assumed [51] and the photon number for each detector element is $10^4$.

### B. RECONSTRUCTION TIME COMPARISON

Because the CUDA kernels return immediately after they are called, the general timer function cannot be applied to test the performance of a kernel. We imitated the examples in CUDA SDK to test the reconstruction time in fan-beam and cone-beam cases. Visual Profiler 6.0 is also applied for more detailed analysis.

The OS-SART algorithm for fan-beam reconstruction with STF based TV minimization is tested in double floating precision. To investigate the relationships among image resolution, view number and the GPU based reconstruction speedup, the image resolution varies from $256^2$ to $2048^2$, the detector resolution varies from 300 to 2400, and the view number changes from 17 to 360. The total iteration is 50. As summarized in Table 2, when the view number is smaller, the speed up is more significant. Larger view numbers or smaller subset sizes indicate more projection and backprojection calls in one sub-loop. When the view number increases, the projection and backprojection change back and forth frequently especially when the subsets are the same size as that of one

**TABLE 1.** Cone-beam reconstruction geometry configuration for numerical simulations and clinical dataset.

| | | Numerical Simulations | | Clinical Dataset |
|---|---|---|---|---|
| Source to detector distance | | 1000mm | | 947mm |
| Source to iso distance | | 850mm | | 539mm |
| Detector Type | | Equidistant | | Equiangular |
| Scan range | | 0 to 360 degrees | | -267 to 93 degrees |
| View number | | CATCE | | 2200 |
| Detector Parameters | Size | $340^2\text{mm}^2$ | Arc | 0.959285172 |
| | | | Height | 65.5296mm |
| | Entries per row | CATCE | | 888 |
| | Truncation ratio | 50% | | 43.7% |
| | Row Number | CATCE | | 64 |
| | Offset | NONE | | 0.767925mm |
| Image Parameters | Size | $200^3\text{mm}^3$ | | $500^2 \times 40\text{mm}^3$ |
| | Resolution | CATCE | | $512^2 \times 64$ |
| | Slices interval | CATCE | | 0.625mm |
| | | | | |

**TABLE 2.** The computational cost for STF-based TV minimization in an OS-SART framework with a modified Shepp-Logan phantom. A comparison is presented between CPU (Intel Xeon 3.1GHz, single core usage) and GPU (NVIDIA, Tesla K10, single GPU usage) after 50 iterations.

| view number | | 17 | | | 21 | | | 72 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| image size | detector size | time(s) | | Speedup factor(≈) | time(s) | | Speedup factors(≈) | time(s) | | Speedup factors(≈) |
| | | CPU | GPU | | CPU | GPU | | CPU | GPU | |
| $256^2$ | 300 | 19.8 | 1.1 | 18 | 20.5 | 1.2 | 17 | 33.8 | 3.6 | 9 |
| $512^2$ | 600 | 70.9 | 2.5 | 28 | 77.6 | 3.0 | 26 | 149.4 | 9.3 | 16 |
| $1024^2$ | 1200 | 257.7 | 7.0 | 37 | 278.3 | 8.4 | 33 | 556.3 | 26.8 | 21 |
| $2048^2$ | 2400 | 1066.2 | 23.4 | 46 | 1133.7 | 28.3 | 40 | 2368.4 | 90.2 | 26 |

| view number | | 180 | | | 360 | | |
|---|---|---|---|---|---|---|---|
| image size | detector size | time(s) | | Speedup factors(≈) | time(s) | | Speedup factors(≈) |
| | | CPU | GPU | | CPU | GPU | |
| $256^2$ | 300 | 63.5 | 8.7 | 7 | 120.9 | 17.1 | 7 |
| $512^2$ | 600 | 320.8 | 22.7 | 14 | 633.2 | 45.1 | 14 |
| $1024^2$ | 1200 | 1176.5 | 65.6 | 18 | 2396.8 | 130.6 | 18 |
| $2048^2$ | 2400 | 4769.9 | 220.8 | 22 | 9083.9 | 439.1 | 21 |

projection view. Meanwhile, the GPU-computing is suitable for CS-based interior reconstruction especially when only a few projections are available. With the improvement of the image resolution, the speedup is more obvious. This is because when the image is small, the projection and backprojection generally are more bandwidth intensive instead of computing intensive.

The SD-based TV minimization in the OS-SART framework is also tested to compare with the STF-based algorithm with image resolution $1024^2$ and $2048^2$ in fan-beam geometry. The bar chart in Figure 3 shows the speedup results. The reconstruction time between the OS-SART and the other two TV regularization based algorithms are minimal, and the STF-based TV minimization algorithm runs a little bit faster than the SD-based TV minimization.

A single-floating-precision $512^3$ modified 3D Shepp-Logan phantom is applied to test the speedup performance in cone-beam geometry with the OS-SART algorithm plus TV regularization. A detector resolution of $600^2$ and 100 views are applied. The computational cost for the STF-based and the SD-based TV minimization in the OS-SART framework are respectively 1665.46 and 1791.33 seconds after 85 iterations. For one iteration step, the computational costs in different algorithms are listed in Table 3. The sum of projection, backprojection and regularization time is not exactly the same as the total time because other time is needed for data loading from disk, data transfer, functions calling and so on. The computational cost is relatively small to find the optimal threshold $\omega_{\gamma_0}$ in the OS-SART with STF-based TV minimization. With more detailed analysis, the GPU utilization for the projection and backprojection steps are both 100.0%, the utilization to find the optimal threshold $\omega_{\gamma_0}$ including filtering is 100.0%, while the occupation rate for DGT is only 75%.

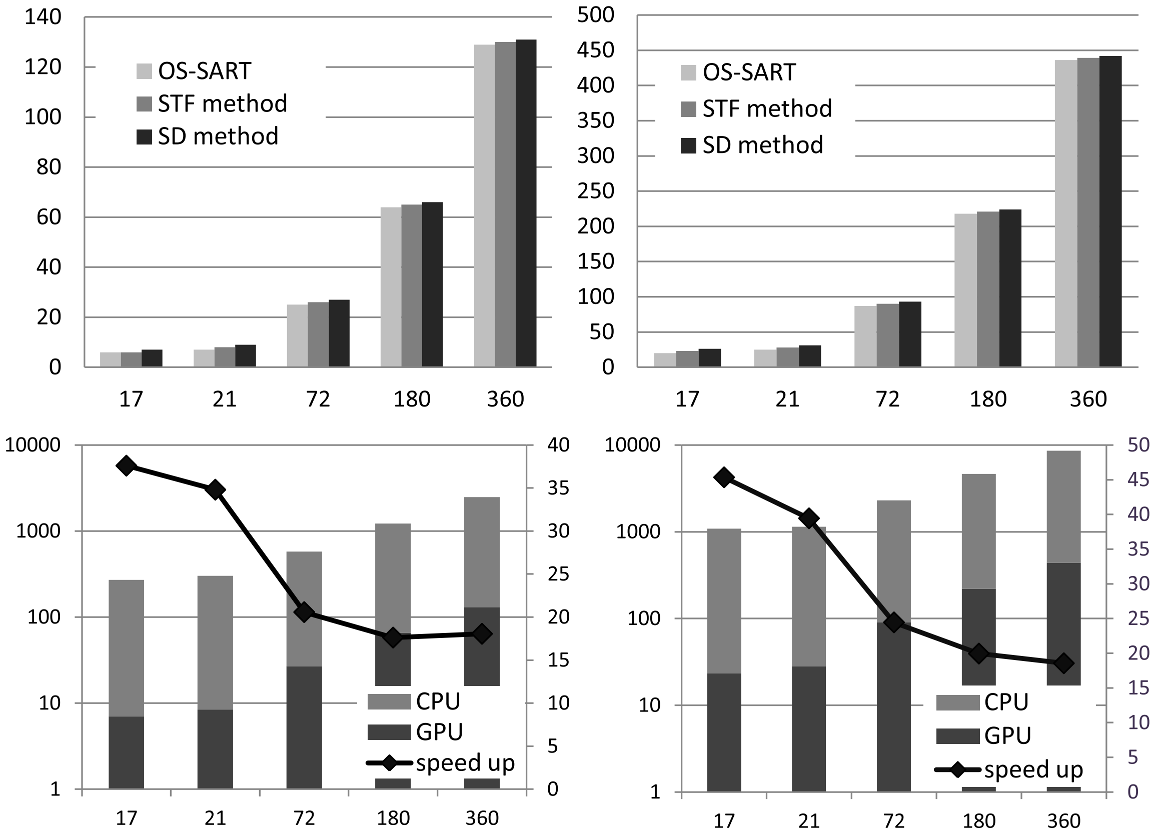


**FIGURE 3.** Bar charts of the computational cost for different GPU-based reconstruction methods in fan-beam geometry. The left column is for image size 1024 × 1024 and the right column is for image size 2048 × 2048 after 50 iterations. For the top row bar charts, the vertical axis represents the reconstruction time and the unit is second. The bottom row is the speedup comparison between CPU and GPU. The abscissa indicates different projection number used in reconstruction. The left vertical axis is the reconstruction time (in seconds, drawn in logarithmic scale), and the right vertical axis is for the speedup factor in the bottom charts.

**TABLE 3.** The computational cost for the projection, backprojection and TV minimization steps in OS-SART, OS-SART with SD-based TV minimization, and OS-SART with STF-based TV minimization.

| | | Total Time | Projection | Backprojection | Regularization | Other |
|---|---|---|---|---|---|---|
| OS-SART | Time: | 17.728s | 9.494s | 6.547s | 0s | 1.687s |
| | Ratio: | 100% | 53.55% | 36.93% | 0% | 9.52% |
| SD-based | Time: | 19.583s | 9.477s | 6.547s | 1.808s | 1.751s |
| | Ratio: | 100% | 48.39% | 33.43% | 9.23% | 8.94% |
| STF-based | Time: | 18.816s | 9.487s | 6.546s | 0.934s | 1.849s |
| | Ratio: | 100% | 50.42% | 34.79% | 4.96% | 9.83% |

A single-floating-precision $512^2 \times 256$ image volume is also reconstructed with the OS-SART. The projection geometry is in Table 1 except that the detector resolution changes to $1024 \times 512$ and the views increases to 360. An iteration number 10 is applied for this study. The total reconstruction time is 423.20 seconds. In one iteration, 42.3 seconds are needed to project the $512^2 \times 256$ image volume to $1024 \times 512 \times 360$ projections and to backproject the projections. From the performance comparison with different reconstruction scales in image resolution, detector resolution and views, we can see that the performance of boundary box based backprojection is largely influenced by the detector resolution. To further accelerate the reconstruction, the dataset is divided into two smaller ones and distributed to dual GPUs evenly. Except the same geometry configuration, both the volumetric image and the raw projections are halved. An iteration number 30 is assumed. There is no data redundancy to guarantee an accurate reconstruction since the symmetry of circular cone-beam scanning geometry. While the computational cost is 583.46 seconds for single Kepler GK104, the computational cost is reduced to 289.85 seconds with two Kepler GK104s. The GUPS (Giga-updates per second) is also tested for the

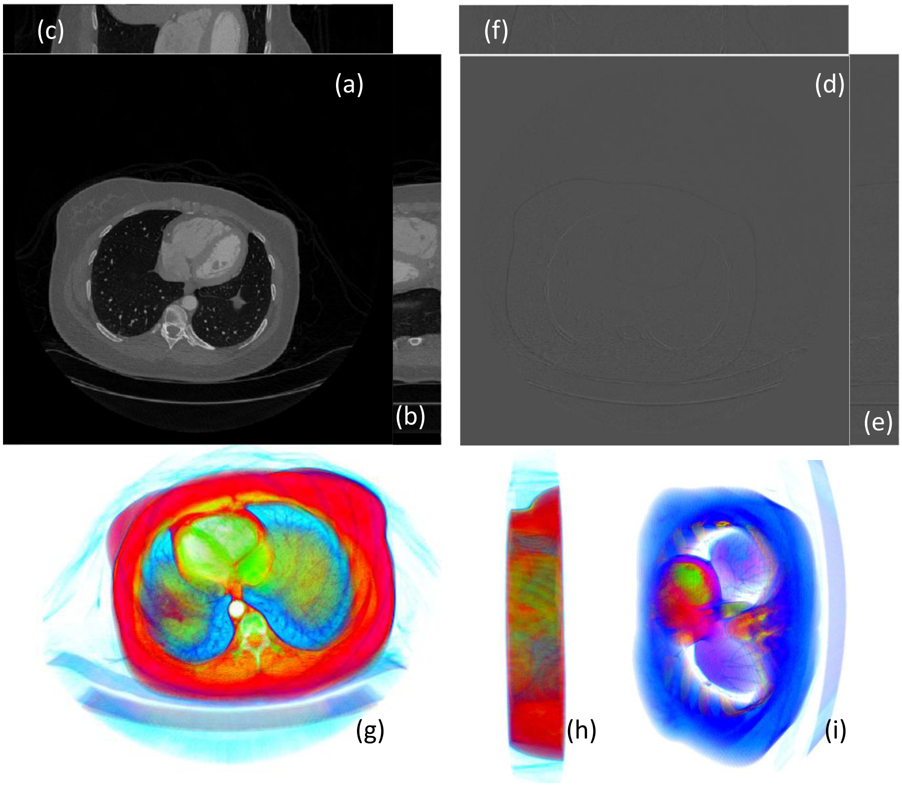


**FIGURE 4.** Image volume reconstructed from a patient data set by the OS-SART algorithm. (a)-(c) are the transverse, sagittal and coronal planes, respectively. The display window is [−1000HU, 2500HU]. (d)-(f) are the corresponding difference images of (a)-(c) related to the ones reconstructed by the FDK method in a narrow display window. (g)-(i) are three pseudo-color volumetric rendering in different observation positions, which are generated by the NVIDIA CUDA SDK under the license EULA (the NVIDIA end user license agreement).

backprojection step. On average, it takes 7.39 seconds for the backprojection step to update $512^3$ image from 360 views in once iteration. This implies that for each view it spends 20.52ms and the GUPS is 6.09.

The recently released unified memory technique in CUDA 6.0 creates a pool of managed memory that is shared between CPU and GPU. It bridges the CPU-GPU gap. This technique makes the programming convenient without considering the data transfer among devices. At the same time, it simplifies the huge volumetric reconstruction when the GPU device memory is insufficient. As the documentation suggested, the raw projection data and the volume to be reconstructed are both declared with key word "__managed__". We implemented a miniature of the projection and backprojection with unified memory technique. The performance from one view with the same geometry configuration makes us a little frustrating. Projecting a $512^2 \times 256$ volume to 1024 × 512 detector takes 214.312ms (22.5% of the time) and backprojecting the same data costs 736.69ms (77.5% of the time). It needs more complicated implementation details to accelerate the performance which will be an extension of our work.

Under the approval of the institutional review board of Wake Forest University Health Sciences, a clinical patient dataset is also reconstructed with the configuration in Table 1. To fully utilize all the GPU resources in our device, three GK104 chips including two Tesla K10s and one GeForce GTX 670 are all occupied. This configuration can be approximately viewed as cone-parallel geometry because the divergence angle is small enough to be ignored and the projection data can be evenly distributed to three GPUs without data redundancy. When only SART is used to reconstruct the image volume, the total computational cost for 160 iterations is 1268.46 seconds. This means 7.93 seconds are needed for one iteration on average. On the other hand, if only one core is applied in CPU implementation, 884.07 seconds are required for one iteration. Therefore, GPU implementation can accelerates the reconstruction more than 110 times in this case. The ordered subset technique is also applied to accelerate the convergence. The projection is divided into 50 subsets and each subset contains 44 views. Totally, 536.12 seconds are required for 40 iterations which guarantee a promising result as shown in Figure 4, and 13.40 seconds are needed for one iteration. This is caused by the following reason: when the subset number increases, although it can accelerate the convergence with more projection and backprojection processes to traverse all the projection views, it will slow down the execution in one loop. The interior scan is simulated by

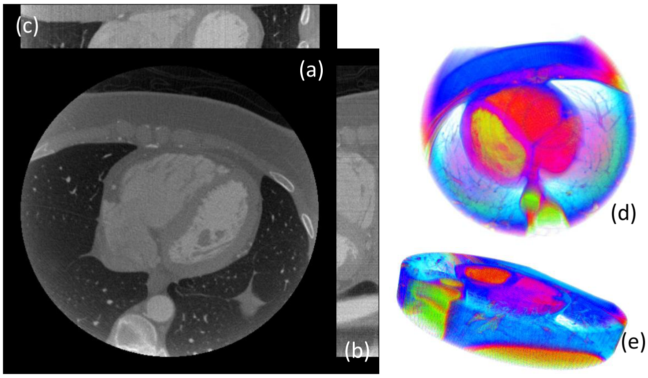


**FIGURE 5.** Image volume reconstructed from a truncated data set by the OS-SART plus TV regularization algorithm. The image matrix size is 52 × 512 × 64. (a)-(c) are the transverse, sagittal and coronal planes, respectively. The display window is [−1000HU, 2500HU]. (d)-(e) are two pseudo-color volumetric rendering in different observation positions, which are generated by the NVIDIA CUDA SDK under the license EULA (the NVIDIA end user license agreement).

truncating 43.7% of the clinical dataset as in Table 1. Because only the interior part is illuminated due to the projection truncation, the FOV is reduced to $230^2$mm$^2$. The projection is divided into 22 subsets for the OS-SART. The computational cost is 161.203 seconds in total for 20 iterations. This means that ∼2.5 minutes can give promising interior reconstruction results as shown in Figure 5.

### C. IMAGE QUALITY

First, the image quality of interior reconstruction is evaluated with a modified Shepp-Logan phantom in fan-beam geometry. The iteration number is 30. Some representative interior reconstruction results are shown in Figure 6. The images reconstructed by the OS-SART from few view projections show serious artifacts inside the internal ROI. The SD-based and STF-based TV minimizations have similar performance after decades of iterations. The slight differences between SD-based and STF-based TV minimizations can be seen from the reconstructions of highly sparse views.

Another group of reconstruction comparison is based on the clinical patient dataset. The sinogram for the central slice in Figure 5 is extracted and uniformly downsampled from 2200 to 180 views. The iterations number is also 30. The interior parts of the reconstruction results are shown in Figure 7. The RMSE (root-mean-square deviation) is calculated for the ROI indicated by the red circle in Figure 7(a) to quantitatively evaluate the image quality. While the RMSE of Figure 7(b) is 88.28HU, the RMSE of figure 7(c) is 79.81HU. The SSIM (the structural similarity index) is calculated to evaluate the similarity [52]. The SSIM for the STF-based OS-SART is 0.9140 while the SSIM for the SD-based OS-SART is 0.8723. The two quantitative measurements show that the STF-based OS-SART outperforms the SD-based OS-SART in this study.

In cone-beam geometry, a $512^3$ single floating precision modified Shepp-Logan phantom is reconstructed from 72 views. The detector resolution is $600^2$ with the simulation configuration summarized in Table 1. The reconstructed results in transverse, sagittal and coronal planes are shown in Figure 8. The SSIM on these three planes are 0.9587, 0.9850 and 0.9690, respectively in the STF-based reconstruction method. In the SD-based reconstruction method, the corresponding SSIM values are 0.9279, 0.9226 and 0.9029,

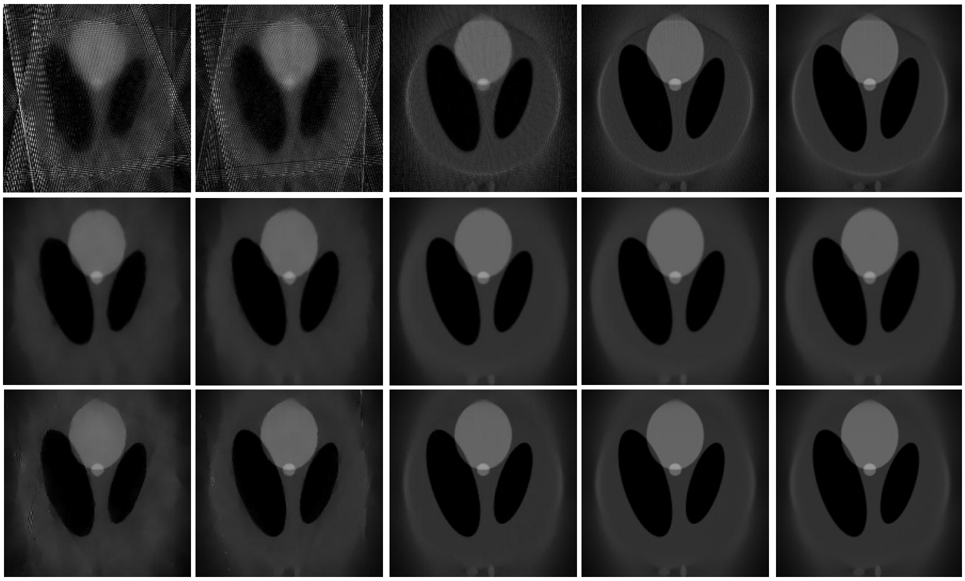


**FIGURE 6.** Representative results reconstructed from truncated local projections for a modified Shepp- Logan phantom after 30 iterations. Form left to right columns, the images are reconstructed form 17, 21, 72,180 and 360 projections, respectively. From top to bottom rows, the images are reconstructed by the OS-SART, OS-SART plus steepest descent and OS-SART plus soft-threshold filtering for TV minimization, respectively. The display window is [0, 1].

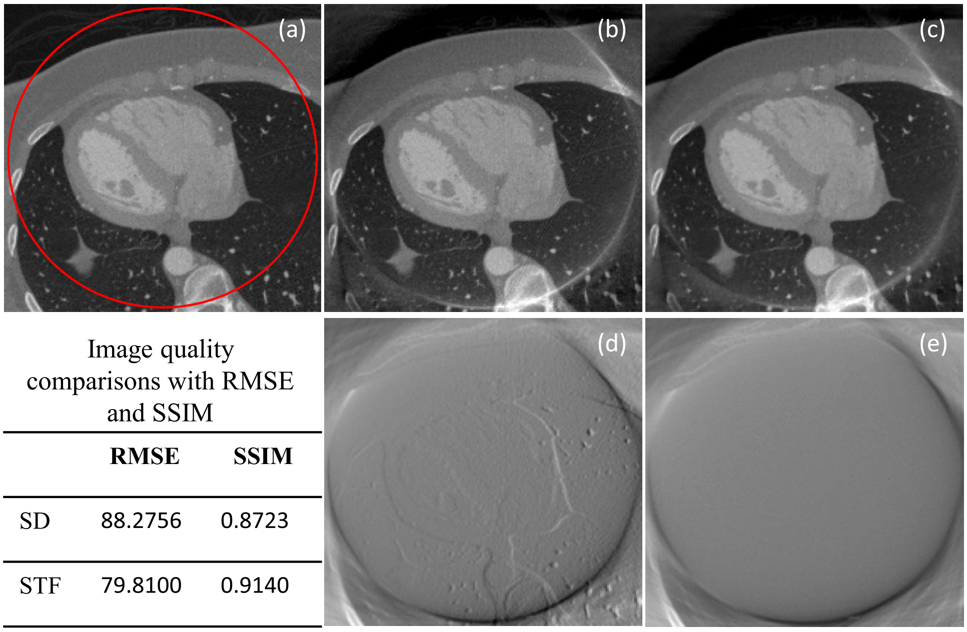


Image quality comparisons with RMSE and SSIM

| | RMSE | SSIM |
|---|---|---|
| SD | 88.2756 | 0.8723 |
| STF | 79.8100 | 0.9140 |

**FIGURE 7. Reconstructed results of a cardiac region from clinical projections. (a) is the reference image, (b) is reconstructed by the SD-based TV minimization, and (c) is reconstructed by the STF-based TV minimization. For (a) to (c), the display window is [−1000HU, 1800HU]. (d) and (e) are the corresponding difference images of (b) and (c) related to (a), respectively, and the display window is [−2400*HU*, 400*HU*].**

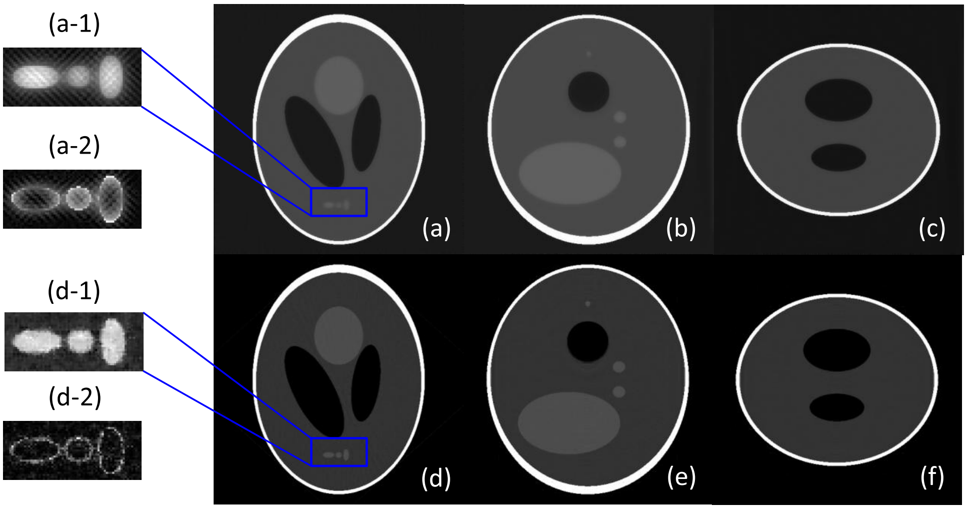


**FIGURE 8. Representative results of the OS-SART with SD-based TV minimization (the top row images) and the OS-SART with STF-based TV minimization (the bottom row images) in transverse (left column), sagittal (middle column) and coronal (right column) views. The display window is [0,1]. The subfigures (a-1) and (d-1) are the magnified parts of (a) and (d). (a-2) and (d-2) are the error images of (a-1) and (d-1) in reference to the original phantom. It can be seen that the STF-based OS-SART can keep more fine details than the SD-based OS-SART.**

respectively. Therefore, the STF-based reconstruction keeps more fine structures in the volume and the residual errors are smaller.

For the clinical patient dataset, it can be easily observed from Figure 4 that the differences are really small between the images reconstructed by the OS-SART and FBP algorithm. Moreover, as shown in Figure 7, the STF-based TV minimization provides better reconstruction result compared to the SD-based TV minimization for interior tomography. To validate the convergence of the STF-based OS-SART algorithm, 50 views are uniformly sampled from the sinogram of the central slice to reconstruct an image of $256^2$. Figure 9 shows the reference and seven intermediate images with respect to different iteration numbers. From images (b) to (h), the image quality improves gradually with the increase of the iteration number. However,

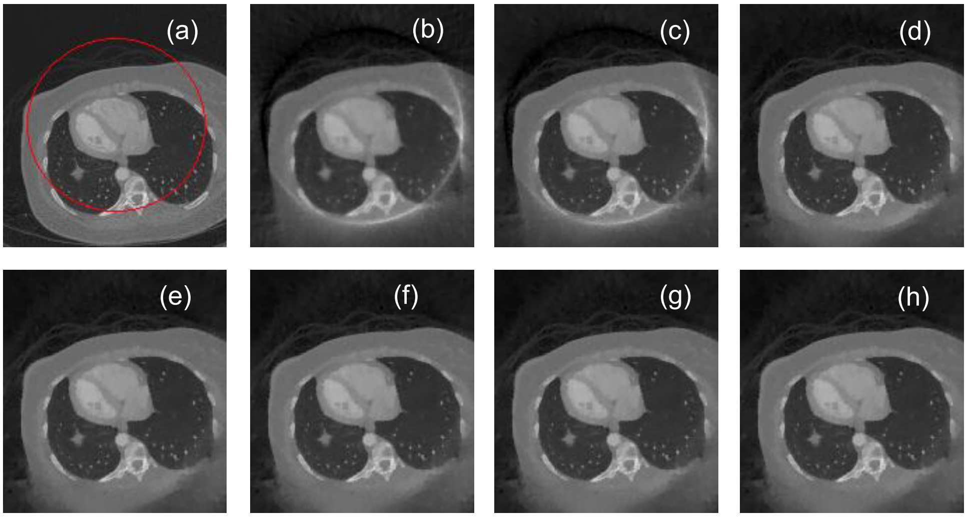


**FIGURE 9. Interior reconstruction results from 50 projections after different iterations. The image size is 256 × 256. (a) is the reference image reconstructed by the FBP method from 2200 global projections. (b) to (h) are the reconstructed images from 50 projections after 10, 20, 100, 200, 500, 1000 and 5000 iterations, respectively.**

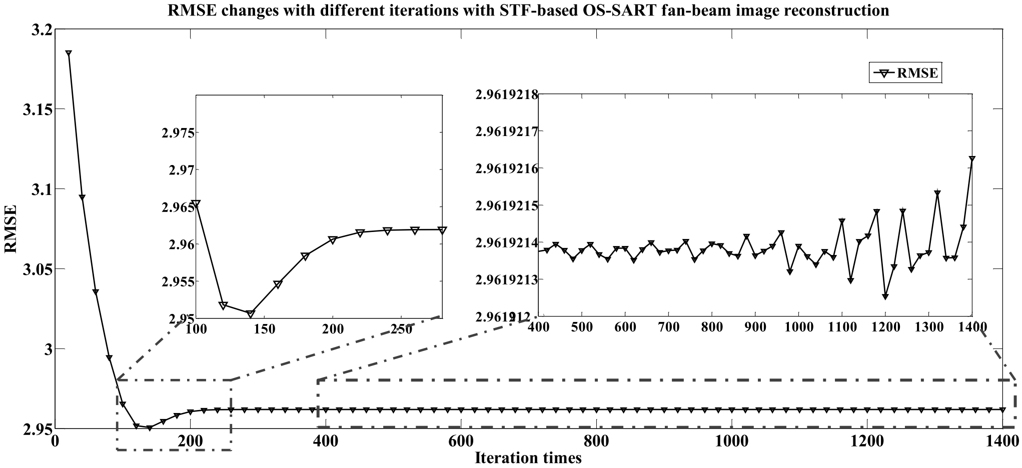


**FIGURE 10. Reconstruction error curve with respect to iterative number. The errors are computed in reference to region marked by the red circle in image (a) in Fig. 9. Two subfigures inside the main figure are the local amplifications of the main curve.**

when the iterations number is sufficient large (e.g. 200), the image quality becomes stable. From the convergence curve in Figure 10, one can see that the reconstruction error (the sum of pixel error squares) decreases rapidly in the first decades of iterations, reaches the minimizer after 130 iterations, then increases a little bit and finally becomes stable after 200 iterations.

## V. CONCLUSION

The x-ray CT is one of the most important imaging modalities for non-destructive diagnosis and image-guided intervention despite the potential radiation risks. To reduce the radiation dose, we have implemented the CS-based interior reconstruction in GPU for fan-beam and cone-beam reconstruction in this paper. The TV regularization is adopted in our work by incorporating the SD or STF methods. These two methods are both implemented and compared. To test the reconstruction performance, we performed several groups of experiments with different reconstruction parameters from simulated and real datasets in both fan-beam and cone-beam geometries.

The GPU parallel computing can be used to boost the CS-based interior tomography for practical applications. We implemented the CS-based interior tomography in GPU devices for fast reconstruction. Our experimental results show that the OS-SART with STF-based TV minimization method runs slightly faster than the SD-based TV minimization and reconstruct promising results in fan-beam geometry using one GPU for acceleration. In the cone-beam geometry experiments, the STF-based method outperforms the SD-based method for few-view projections. Comparing with the CPU-based implementation in fan-beam geometry, the speedup is higher when the views are smaller or the image resolutions are larger in cone-beam case, and the reconstruction speedup with real data is obvious. Therefore, the GPU parallelization is suitable for CS-based interior tomography especially for large-scale volumetric reconstruction. By analyzing the timeline in cone-beam reconstruction, it is found that the projection and backprojection operations dominate the reconstruction cost in the STF-based method. In the near future, we will optimize the implementation and investigate other projection and backprojection models for possible high-efficient GPU implementation to further reduce the computational cost. Other regularization methods will also be studied with implementations in GPU to further decrease the number of views for interior reconstruction. The unified memory technique will be studies to deal with the very large volumetric reconstruction cases.

It is not surprising that the STF-based OS-SART outperforms the SD-based OS-SART. The SD-based TV minimization requires tentatively choosing the parameters such as the descent steps and the descent iteration numbers. It is more likely to find the minimum solution with small steps while the convergence is inferior and the descent iteration number influences the reconstruction speed. On the other hand, the deficiency of STF is its convergence rate. To accelerate the convergence, it is necessary to choose the optimized threshold which is related to dichotomy. The searches range from several times to dozens of times. Actually, there are no too much visual differences between the results reconstructed by the SD-based OS-SART and the STF-based OS-SART. The RMSE and SSIM show that the reconstruction results are comparable with more iterations. For a clinical volumetric reconstruction, several minutes should give promising interior reconstruction results in a realistic setting.

## REFERENCES

[1] T. M. Buzug, *Computed Tomography: From Photon Statistics to Modern Cone-Beam CT*, 1st ed. New York, NY, USA: Springer-Verlag, 2008.

[2] J. Hsieh, *Computed Tomography: Principles, Design, Artifacts, and Recent Advances*, vol. 114, 2nd ed. Bellingham, WA, USA: SPIE, 2009.

[3] O. W. Linton and F. A. Mettler, "National conference on dose reduction in CT, with an emphasis on pediatric patients," *Amer. J. Roentgenol.*, vol. 181, no. 2, pp. 321–329, Aug. 2003.

[4] G. Wang, "The meaning of interior tomography," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Aug. 2011, pp. 5764–5767.

[5] J. Yang, H. Yu, W. Cong, M. Jiang, and G. Wang, "Higher-order total variation method for interior tomography," in *Proc. SPIE*, 2012, p. 85061B.

[6] Q. Xu, H. Yu, X. Mou, L. Zhang, J. Hsieh, and G. Wang, "Low-dose X-ray CT reconstruction via dictionary learning," *IEEE Trans. Med. Imag.*, vol. 31, no. 9, pp. 1682–1697, Sep. 2012.

[7] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Jan. 2006.

[8] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[9] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.

[10] D. L. Donoho, M. Elad, and V. N. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Trans. Inf. Theory*, vol. 52, no. 1, pp. 6–18, Jan. 2006.

[11] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale $l_1$-regularized least squares," *IEEE J. Sel. Top. Signal Process.*, vol. 1, no. 4, pp. 606–617, Dec. 2007.

[12] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE J. Sel. Top. Signal Process.*, vol. 1, no. 4, pp. 586–597, Dec. 2007.

[13] D. L. Donoho and Y. Tsaig, "Fast solution of $\ell_1$-norm minimization problems when the solution may be sparse," *IEEE Trans. Inf. Theory*, vol. 54, no. 11, pp. 4789–4812, Nov. 2008.

[14] H. Yu and G. Wang, "A soft-threshold filtering approach for reconstruction from a limited number of projections," *Phys. Med. Biol.*, vol. 55, no. 13, pp. 3905–3916, Jul. 2010.

[15] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, Nov. 2004.

[16] H. Yu and G. Wang, "SART-type image reconstruction from a limited number of projections with the sparsity constraint," *Int. J. Biomed. Imag.*, vol. 2010, Apr. 2010, Art. ID 934847.

[17] G.-H. Chen, J. Tang, and S. Leng, "Prior image constrained compressed sensing (PICCS): A method to accurately reconstruct dynamic CT images from highly undersampled projection data sets," *Med. Phys.*, vol. 35, no. 2, p. 660, Jan. 2008.

[18] Q. Xu, X. Mou, G. Wang, J. Sieren, E. A. Hoffman, and H. Yu, "Statistical interior tomography," *IEEE Trans. Med. Imag.*, vol. 30, no. 5, pp. 1116–1128, May 2011.

[19] L. Ritschl, F. Bergner, C. Fleischmann, and M. Kachelriess, "Improved total variation-based CT image reconstruction applied to clinical data," *Phys. Med. Biol.*, vol. 56, no. 6, pp. 1545–1561, Mar. 2011.

[20] J. Liu and Q. Hu, "Cardiac CT image reconstruction based on compressed sensing," *Proc. Eng.*, vol. 29, pp. 2235–2239, Feb. 2012.

[21] Y. Ye, H. Yu, and G. Wang, "Exact interior reconstruction with cone-beam CT," *Int. J. Biomed. Imag.*, vol. 2007, pp. 1–5, Dec. 2007, Art. ID 10693.

[22] Y. Ye, H. Yu, and G. Wang, "Exact interior reconstruction from truncated limited-angle projection data," *J. Biomed. Imag.*, vol. 2008, Jan. 2008, Art. ID 427989-1–427989-6.

[23] S. Cook, *CUDA Programming: A Developer's Guide to Parallel Computing With GPUs*. Matlock Bath, U.K.: Newnes, 2012.

[24] X. Li, J. Ni, and G. Wang, "Parallel iterative cone beam CT image reconstruction on a PC cluster," *J. X-Ray Sci. Technol.*, vol. 13, no. 2, pp. 63–72, 2005.

[25] B. Flachs *et al.*, "A streaming processing unit for a CELL processor," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers (ISSCC)*, Feb. 2005, pp. 134–135.

[26] A. Heinecke, M. Klemm, and H.-J. Bungartz, "From GPGPU to many-core: Nvidia fermi and intel many integrated core architecture," *Comput. Sci. Eng.*, vol. 14, no. 2, pp. 78–83, Mar./Apr. 2012.

[27] T. Zinßer and B. Keck, "Systematic performance optimization of cone-beam back-projection on the Kepler architecture," in *Proc. 12th Int. Meeting Fully Three-Dimensional Image Reconstruct. Radiol. Nucl. Med.*, Jul. 2013, pp. 225–228.

[28] R. S. Wright, N. Haemel, G. Sellers, and B. Lipchak, *OpenGL SuperBible: Comprehensive Tutorial and Reference*, 5th ed. Reading, MA, USA: Addison-Wesley, 2010.

[29] W. R. Mark, R. S. Glanville, K. Akeley, and M. J. Kilgard, "Cg: A system for programming graphics hardware in a C-like language," *ACM Trans. Graphics*, vol. 22, no. 3, pp. 896–907, Jul. 2003.

[30] D. Feinstein, *HLSL Development Cookbook*. Birmingham, U.K.: Packt Publishing, 2013.

[31] B. Gaster, L. Howes, D. R. Kaeli, P. Mistry, and D. Schaa, *Heterogeneous Computing With OpenCL*, 1st ed. San Mateo, CA, USA: Morgan Kaufmann, 2011.

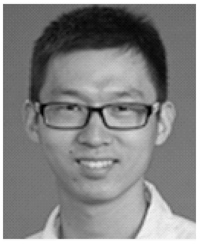
[32] I. Buck *et al.*, "Brook for GPUs: Stream computing on graphics hardware," *ACM Trans. Graphics*, vol. 23, no. 3, pp. 777–786, Aug. 2004.

[33] M. Fatica *et al.*, "High performance computing with CUDA," in *Proc. Int. Supercomput. Conf.*, Jun. 2008.

[34] M. Garland *et al.*, "Parallel computing experiences with CUDA," *Micro IEEE*, vol. 28, no. 4, pp. 13–27, Jul./Aug. 2008.

[35] D. Luebke, "CUDA: Scalable parallel programming for high-performance scientific computing," in *Proc. 5th IEEE Int. Symp. Biomed. Imag., Nano Macro (ISBI)*, May 2008, pp. 836–838.

[36] B. D. Man and S. Basu, "Distance-driven projection and backprojection in three dimensions," *Phys. Med. Biol.*, vol. 49, no. 11, pp. 2463–2475, Jun. 2004.
[37] R. L. Siddon, "Fast calculation of the exact radiological path for a three-dimensional CT array," *Med. Phys.*, vol. 12, no. 2, pp. 252–255, 1985.
[38] H. Yu and G. Wang, "Finite detector based projection model for high spatial resolution," *J. X-Ray Sci. Technol.*, vol. 20, no. 2, pp. 229–238, May 2012.
[39] Y. Long, J. A. Fessler, and J. M. Balter, "3D forward and back-projection for X-ray CT using separable footprints," *IEEE Trans. Med. Imag.*, vol. 29, no. 11, pp. 1839–1850, Nov. 2010.
[40] G. Wang and M. Jiang, "Ordered-subset simultaneous algebraic reconstruction techniques (OS-SART)," *J. X-Ray Sci. Technol.*, vol. 12, no. 3, pp. 169–177, 2004.
[41] C. Kamphuis and F. J. Beekman, "Accelerated iterative transmission CT reconstruction using an ordered subsets convex algorithm," *IEEE Trans. Med. Imag.*, vol. 17, no. 6, pp. 1101–1105, Dec. 1998.
[42] F. J. Beekma and C. Kamphuis, "Ordered subset reconstruction for X-ray CT," *Phys. Med. Biol.*, vol. 46, no. 7, pp. 1835–1844, Jul. 2001.
[43] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm with application to wavelet-based image deblurring," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2009, pp. 693–696.
[44] R. Chartrand, "Exact reconstruction of sparse signals via nonconvex minimization," *IEEE Signal Process. Lett.*, vol. 14, no. 10, pp. 707–710, Oct. 2007.
[45] A. H.-D. Cheng and D. T. Cheng, "Heritage and early history of the boundary element method," *Eng. Anal. Bound. Elements*, vol. 29, no. 3, pp. 268–302, Mar. 2005.
[46] E. J. Candes and J. Romberg, "Quantitative robust uncertainty principles and optimally sparse decompositions," *Found. Comput. Math.*, vol. 6, no. 2, pp. 227–254, Apr. 2006.
[47] F. Jacobs, E. Sundermann, B. De Sutter, M. Christiaens, and I. Lemahieu, "A fast algorithm to calculate the exact radiological path through a pixel or voxel space," *J. Comput. Inform. Technol.*, vol. 6, no. 1, pp. 89–94, 1998.
[48] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics: Principles and Practice in C*, 2nd ed. Reading, MA, USA: Addison-Wesley, 1995.
[49] N. Li, H.-X. Zhao, S.-H. Cho, J.-G. Choi, and M.-H. Kim, "A fast algorithm for voxel-based deterministic simulation of X-ray imaging," *Comput. Phys. Commun.*, vol. 178, no. 7, pp. 518–523, Apr. 2008.
[50] M. Pharr and R. Fernando, *GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation*. Reading, MA, USA: Addison-Wesley, 2005.
[51] H. Yu, Y. Ye, S. Zhao, and G. Wang, "Local ROI reconstruction via generalized FBP and BPF algorithms along more flexible curves," *Int. J. Biomed. Imag.*, vol. 2006, Dec. 2006, Art. ID 14989.
[52] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.



**RUI LIU** is currently pursuing the Ph.D. degree in biomedical engineering with the Department of Biomedical Engineering, Wake Forest University Health Sciences, Winston-Salem, NC, USA. He received the bachelor's degree in computer science and technology from South China Normal University, Guangzhou, China, in 2009, and the master's degree in computer architecture from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2012. His research interests include computed tomography and medical image processing with an emphasis on GPU acceleration for compressive sensing based interior reconstruction algorithms.

**YAN LUO** (M'05) is an Associate Professor with the Department of Electrical and Computer Engineering, University of Massachusetts Lowell, Lowell, MA, USA. He received the Ph.D. degree in computer science from the University of California at Riverside, Riverside, CA, USA, in 2005, and the B.E. and M.E. degrees from the Huazhong University of Science and Technology, Wuhan, China, in 1996 and 2000, respectively. While his research interest spans broadly computer architecture and network systems, his current research focuses on heterogeneous architecture and systems, software-defined networks, and deep learning. He has served on the program committee of numerous international conferences, and has served as a Guest Editor and referee of premier journals. He directs the Laboratory of Computer Architecture and Network Systems, which has been supported by the National Science Foundation, Intel, Raytheon/BBN, Xilinx, and Altera. He has published numerous peer-reviewed journal and conference papers with one Best Paper Award. He is a member of the Association for Computing Machinery.

**HENGYONG YU** (SM'06) is an Assistant Professor and the Director of the CT Laboratory with the Department of Biomedical Engineering, Wake Forest University Health Sciences, Winston-Salem, NC, USA. He received the bachelor's degrees in information science and technology, and computational mathematics, and the Ph.D. degree in information and communication engineering from Xi'an Jiaotong University, Xi'an, China, in 1998, 1998, and 2003, respectively. He was an Instructor and Associate Professor with the College of Communication Engineering, Hangzhou Dianzi University, Hangzhou, China, from 2003 to 2004. From 2004 to 2006, he was a Post-Doctoral Fellow and an Associate Research Scientist with the Department of Radiology, University of Iowa, Iowa City, IA, USA. From 2006 to 2010, he was a Research Scientist and the Associate Director of the CT Laboratory with the Biomedical Imaging Division, VT-WFU School of Biomedical Engineering and Sciences, Virginia Polytechnic Institute and State University, Blacksburg, VA, USA. He joined Wake Forest University Health Sciences as a Faculty Member in 2010. His interests include computed tomography and medical image processing. He has authored and co-authored more than 100 peer-reviewed journal papers with an H-index of 21 according to the Web of knowledge. He is the founding Editor-in-Chief of *JSM Biomedical Imaging Data Papers*, serves as an Editorial Board member of *Signal Processing*, *Journal of Medical Engineering, CT Theory and Applications*, *International Journal of Biomedical Engineering and Consumer Health Informatics*, and *Open Medical Imaging Journal*, and the Guest Editor of the IEEE TRANSACTIONS ON MEDICAL IMAGING, the IEEE ACCESS, and *International Journal of Biomedical Imaging*. He is a Senior Member of the IEEE Engineering in Medicine and Biology Society, and a member of the American Association of Physicists in Medicine and the Biomedical Engineering Society. He was for a recipient of the Outstanding Doctoral Dissertation Award from Xi'an Jiaotong University in 2005, and the best Natural Science Paper Award from the Association of Science and Technology of Zhejiang Province, and the NSF CAREER Award for development of CS-based interior tomography in 2012.