Fast Model-Based X-Ray CT Reconstruction Using Spatially Nonhomogeneous ICD Optimization

Zhou Yu, *Member, IEEE*, Jean-Baptiste Thibault, *Member, IEEE*, Charles A. Bouman, *Fellow, IEEE*, Ken D. Sauer, *Member, IEEE*, and Jiang Hsieh, *Senior Member, IEEE*

***Abstract*—Recent applications of model-based iterative reconstruction (MBIR) algorithms to multislice helical CT reconstructions have shown that MBIR can greatly improve image quality by increasing resolution as well as reducing noise and some artifacts. However, high computational cost and long reconstruction times remain as a barrier to the use of MBIR in practical applications. Among the various iterative methods that have been studied for MBIR, iterative coordinate descent (ICD) has been found to have relatively low overall computational requirements due to its fast convergence. This paper presents a fast model-based iterative reconstruction algorithm using spatially nonhomogeneous ICD (NH-ICD) optimization. The NH-ICD algorithm speeds up convergence by focusing computation where it is most needed. The NH-ICD algorithm has a mechanism that adaptively selects voxels for update. First, a voxel selection criterion VSC determines the voxels in greatest need of update. Then a voxel selection algorithm VSA selects the order of successive voxel updates based upon the need for repeated updates of some locations, while retaining characteristics for global convergence. In order to speed up each voxel update, we also propose a fast 1-D optimization algorithm that uses a quadratic substitute function to upper bound the local 1-D objective function, so that a closed form solution can be obtained rather than using a computationally expensive line search algorithm. We examine the performance of the proposed algorithm using several clinical data sets of various anatomy. The experimental results show that the proposed method accelerates the reconstructions by roughly a factor of three on average for typical 3-D multislice geometries.**

***Index Terms*—Computed tomography, coordinate descent, iterative algorithm, model based iterative reconstruction (MBIR).**

I. Introduction

RECENT applications of model based iterative reconstruction (MBIR) algorithms to multislice helical CT reconstructions have shown that MBIR can greatly improve image quality by increasing resolution as well as reducing noise and some artifacts [1]–[8]. MBIR algorithms typically work by first forming an objective function which incorporates an accurate system model [9], [10], statistical noise model [11], [12] and prior model [13]–[15]. The image is then reconstructed by computing an estimate which minimizes the resulting objective function.

A major obstacle for clinical application of MBIR is the fact that these algorithms are very computationally demanding compared to conventional reconstruction algorithms due to the more elaborate system models and the need for multiple iterations. Although accurate models are critical for MBIR to achieve high image quality, they also tend to result in objective functions that are difficult to compute and optimize. In an effort to speed up the iterative reconstruction, various hardware platforms are being considered [16], [17], and a variety of iterative methods, such as variations of expectation maximization (EM) [18], conjugate gradients (CG) [19], ordered subsets (OS) [20], and iterative coordinate descent (ICD) [21], is typically used to minimize the objective function. Each iteration of these methods can be computationally expensive since it typically requires at least one pass through a large volume of CT data, and the number of required iterations depends upon both the desired image quality and the convergence speed of the particular iterative algorithm.

Among various iterative methods that have been applied to MBIR, ICD, and similar sequential updating techniques such as OS-ICD [22], [23] and group coordinate ascent [24], [25], have been found to have relatively low overall computational requirements. The convergence behavior of the ICD algorithm has been studied in the literature [26], [27]. In particular, Bouman and Sauer's study on tomographic reconstruction using ICD showed that it has rapid convergence for high spatial frequencies and near edge pixels of the reconstruction [21]. In fact, among the optimization algorithms compared in [28], the ICD algorithm was found to have a relatively fast convergence behavior when it is initialized with the FBP reconstruction, which usually provides a good estimate of the low spatial frequency content of the image that tends to converge more slowly with the ICD algorithm. It should be noted that ICD tends to have less regular memory access than gradient based optimization methods. Therefore, depending upon the computation platform, this can negatively impact the per-iteration computation time. Nonetheless, we have found that the total computation time for ICD generally compares quite favorably to alternative methods in practical implementations.

The ICD algorithm works by decomposing the N-dimensional optimization problem into a sequence of greedy 1-D voxel updates. A full iteration of the conventional ICD algorithm then updates all the voxels in the image volume once

Manuscript received October 16, 2009; revised April 20, 2010; accepted June 17, 2010. Date of publication July 19, 2010; date of current version December 17, 2010. This work was supported by GE Healthcare. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Miles N. Wernick.

Z. Yu, J.-B. Thibault, and J. Hsieh are with GE Healthcare Technologies, Waukesha, WI 53188 USA (e-mail: zhou.yu@ge.com; jean-baptiste.thibault@med.ge.com; jiang.hsieh@med.ge.com).

C. A. Bouman is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907-0501 USA (e-mail: bouman@ecn.purdue.edu).

K. D. Sauer is with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556-5637 USA (e-mail: sauer@nd.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2010.2058811

and once only. In addition to the fast convergence speed, ICD has also been found to have a number of useful properties in model-based iterative reconstruction. First, the ICD algorithm can easily incorporate positivity constraints and non-Gaussian prior distributions; and in particular, positivity constraints can be difficult to incorporate into CG. This is important because nonquadratic regularization can substantially improve image quality, but can make optimization more difficult. Second, the ICD algorithm naturally allows for updates that vary nonuniformly across the reconstructed image. This last property has not been fully exploited so far, and provides a rich opportunity for reducing the computation of MBIR.

In this paper, we propose the nonhomogeneous ICD (NH-ICD) algorithm [29] that can substantially accelerate convergence relative to the conventional ICD algorithm for tomographic reconstruction. The NH-ICD algorithm takes advantage of the flexibility of ICD by selectively updating voxels that can benefit most from updates. Typically, the errors between the FBP initialization of ICD and the converged reconstruction are not uniformly distributed across the image. In fact, these initial errors tend to be primarily distributed around edges and other localized regions. Therefore, the convergence speed of ICD can be improved by focusing computational resources on these important locations. In order to select the order of voxel updates, we formulate a voxel selection criterion (VSC) to determine the voxels in greatest need of update. We also develop a voxel selection algorithm (VSA) that balances the need for repeated updates of some voxels with the need for more uniform updating of all voxels to guarantee global convergence.

We also propose a fast algorithm for approximately solving the 1-D optimization problem of each voxel update [30] in order to speed up the ICD algorithm. The fast 1-D update algorithm is based on the functional substitution (FS) approach [24], [31], [32], which replaces the objective function with a simplified substitute function. By carefully deriving the substitute function, the FS approach reduces computation while also guaranteeing monotone convergence of the objective function. The substitute function we propose in this paper is designed for the particular q-GGMRF [2] prior model we are using. However, it can be easily generalized to other prior models as long as the potential function in the prior model satisfies certain constraints.

In our experiments, we examine the performance of the proposed algorithms using several clinical data sets which cover a variety of anatomical locations. The experimental results show that the proposed algorithms reduce the computation time required to achieve desired image quality by approximately a factor of three on average as compared to the conventional ICD algorithm.

The paper is organized as follows. Section II provides a review of the conventional ICD algorithm for 3-D iterative reconstruction. Section III presents the spatially nonhomogeneous ICD algorithm. Section IV presents the fast 1-D optimization algorithm. Finally, in Section V we show the experimental results on clinical data cases to quantify the improvement in computation speed.

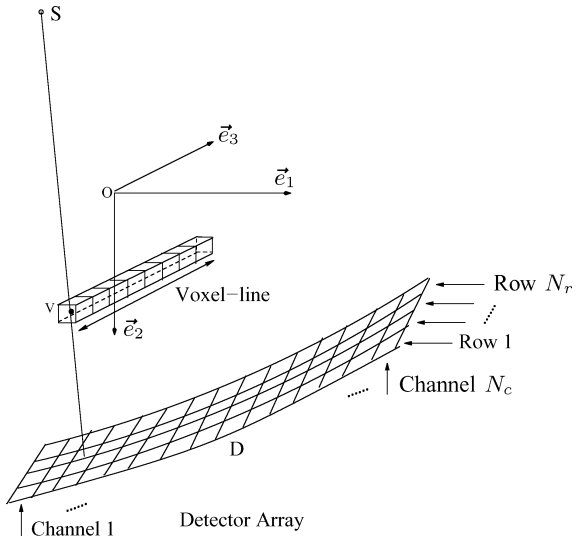


Fig. 1. Illustration of the geometry of multislice CT. S is the focus of the X-ray source, D is the detector array, in which detector cells are aligned in channels and rows. A single row of detectors forms an arc which is equidistant from S, but a single channel of detectors falls along a straight line parallel to the $\vec{e}_3$-axis. Voxels along the same (j_1, j_2) location form a voxel-line.

II. Conventional ICD Algorithm for 3-D Reconstruction

A. Statistical Model and Objective Function

In this section, we introduce the conventional ICD algorithm for reconstruction of 3-D volumes from data obtained using a multislice CT system. The corresponding cone-beam geometry is illustrated in Fig. 1 where S denotes the focus of the X-ray source, D is the detector array, and the detector channels of each row are located on an arc which is centered at S. When taking measurements in the helical scan mode, the source and detector array rotate around the patient, while the patient is simultaneously translated in the direction perpendicular to the plane of rotation. The trajectory of the source S relative to the patient forms a helical path.

To define the coordinate system for the reconstructed image volume, let $\vec{e}_1$, $\vec{e}_2$, and $\vec{e}_3$ denote the basis vectors of a right-handed coordinate system. The origin of the coordinate system is placed at the center of the rotation, also known as the "isocenter" of the scanner. As shown in Fig. 1, $\vec{e}_1$ and $\vec{e}_2$ are in the plane that is perpendicular to the axis of the helix, while $\vec{e}_1$ is pointing to the right and $\vec{e}_2$ is pointing downward. The third axis, $\vec{e}_3$, is pointing along the axis of the helix. The reconstruction is then denoted by x_j where $j = (j_1, j_2, j_3)$ is a vector index with $1 \leq j_1 \leq N_1$, $1 \leq j_2 \leq N_2$, and $1 \leq j_3 \leq N_3$ denoting the number of voxels along the three axis. For notational simplicity, we will assume that x is a vector with elements x_i indexed by $1 \leq i \leq N_1 N_2 N_3$.

The detector cells are aligned in channels and rows, as shown in Fig. 1. Each row is formed by an array of detectors which are equidistant from the source S. The detectors in each row are indexed by their channel numbers. For a given channel, the

detector cells from each row form a straight line that is parallel to the $\vec{e}_3$ axis.

The measurements from the detector array are sampled at uniformly spaced intervals in time. The full set of detector measurements sampled at a single time is known as a projection view. Therefore, the projection measurements form a 3-D array denoted by y_{i_v,i_r,i_c}, where $0 \leq i_v \leq N_v$ indexes the view, $0 \leq i_r \leq N_r$ indexes the detector row, and $0 \leq i_c \leq N_c$ indexes the detector channel. For simplicity, we use the notation y_i where $i = (i_v, i_c, i_r)$ to denote a single measurement.

We consider the image x and the data y as random vectors, and our goal is to reconstruct the image by computing the maximum *a posteriori* (MAP) estimate given by

$$\hat{x} = \arg\min_{x \geq 0} \{-\log p(y|x) - \log p(x)\} \tag{1}$$

where $p(y|x)$ is the conditional distribution of y given x, $p(x)$ is the prior distribution of x, and $x \geq 0$ denotes that each voxel must be nonnegative. We can use a Taylor series expansion to approximate the log likelihood term using a quadratic function [21], [33], resulting in

$$\log p(y|x) \approx -\frac{1}{2}(y - Ax)^T D(y - Ax) + f(y) \tag{2}$$

where A is the forward system matrix, D is a diagonal weighing matrix and $f(y)$ is a function which depends upon measurement data only. The ith diagonal entry of the matrix D, denoted by d_i, is inversely proportional to an estimate of the variance in the measurement y_i [12], [21], [33]. We use the photon count measurement λ_i to estimate the variance in y_i. In theory, the relationship between y_i and λ_i is given by

$$y_i = \ln \frac{\lambda_T}{\lambda_i} \tag{3}$$

where λ_T is the expected photon count where there is no object present. We model λ_i as the sum of a Poisson random variable with mean μ_i and the electronic noise with mean zero and variance σ_n^2. Therefore, we can derive the variance of y_i as [12]

$$\sigma_{y_i}^2 \approx \frac{\mu_i + \sigma_n^2}{\mu_i^2}. \tag{4}$$

Since λ_i is an unbiased estimation of μ_i, we have

$$d_i = \frac{\lambda_i^2}{\lambda_i + \sigma_n^2} \tag{5}$$

where σ_n^2 can be experimentally estimated.

We use a distance driven (DD) forward model [9] for the calculation of A [2]. We choose the DD forward model mainly because it is relatively fast to compute, and it has been shown to produce images free of visible aliasing artifacts introduced by the forward model [9]. To forward project one voxel using the DD model, we first "flatten" the voxel to a rectangle as shown in Fig. 2(a). Then we compute the projection of the four boundaries of this flattened voxel onto the detector array. The projection is approximated as a rectangle as shown in Fig. 2(b), and can be specified by its width W, length L and the center location (δ_c, δ_r) in the detector coordinate system. Consequently, the location and the size of the rectangular footprint can be separately computed in the $(\vec{e}_1, \vec{e}_2)$ plane and along the $\vec{e}_3$ axis. Fig. 2(c) illustrates the computation of the footprint in the $(\vec{e}_1, \vec{e}_2)$ plane, wherein the flattened voxel is shown as the horizontal dashed line, and the parameters of the footprint, W and δ_c, can be computed in this plane. Similarly, the parameters δ_r and L can be computed in the plane that is perpendicular to the $(\vec{e}_1, \vec{e}_2)$ plane as shown in Fig. 2(d). Therefore, the forward model can be calculated separately in the $(\vec{e}_1, \vec{e}_2)$ plane and the $(\vec{e}_1, \vec{e}_3)$ plane. Later we will discuss how to use the separability to efficiently calculate the forward projection of a line of voxels that are parallel to the $\vec{e}_3$ axis.

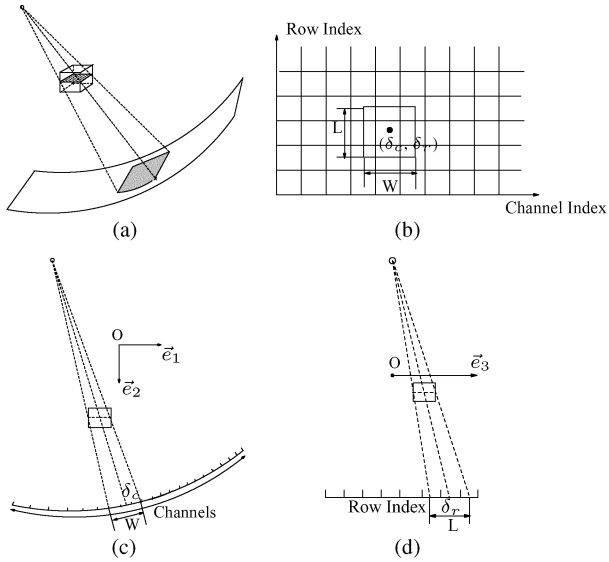


Fig. 2. Illustration of the geometric calculation of the distance driven forward model. (a) The shaded area on the detector shows the footprint of the voxel. To simplify the calculation, we "flatten" the voxel to a rectangle and then project its boundaries onto the detector array. (b) Shows the footprint of a voxel on the detector array, in which the grid represents the detector cells. The footprint can be approximated by a rectangle specified by the parameters W, L and (δ_c, δ_r). The geometric calculation in the $(\vec{e}_1, \vec{e}_2)$ plane is illustrated in (c). Flattened voxel is illustrated as the horizontal dashed line. The parameters W and δ_c can be computed by projecting the dashed line onto the detector array. (d) Geometric calculation along $\vec{e}_3$ is illustrated.

We use a Markov random field (MRF) as our image prior model with the form

$$-\log p(x) = \sum_{\{j,k\} \in \Omega} w_{jk} \rho(x_j - x_k) \tag{6}$$

where Ω is the set of all the neighboring voxel pairs, w_{jk} are fixed weights, and $\rho(\cdot)$ is a symmetric potential function. The potential function considered in this paper is a nonquadratic potential function with the form

$$\rho(\Delta) = \frac{|\Delta|^p}{1 + |\Delta/c|^{p-q}} \tag{7}$$

with $1 < q \leq p \leq 2$ [2]. We refer to MRF prior models which uses this potential function as the q-generalized Gaussian Markov random field (q-GGMRF). Fig. 3 shows the plots of the potential function, and its derivative, also known as the influence function. The potential function of (7) is strictly convex when $1 < q \leq p \leq 2$ [2]. Strict convexity of the potential function is important because it ensures that there is a unique solution to the optimization problem and that the MAP reconstruction is

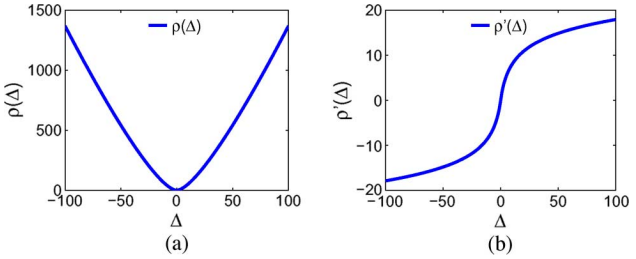


Fig. 3. (a) Shows the potential function $\rho(\Delta)$. (b) Shows the influence function $\rho'(\Delta)$ for a choice of parameters with $p = 2$, $q = 1.2$ and $c = 10$.

UpdateVoxel(j,x,e){

$\tilde{x}_j \leftarrow x_j$

$A_{*j} \leftarrow Compute$

$\theta_1 = \sum_{i=1}^{N} d_i A_{ij} e_i$

$\theta_2 = \sum_{i=1}^{N} d_i A_{ij}^2$

$x_j \leftarrow \arg\min_{u \geq 0} \left\{ \theta_1 u + \frac{\theta_2 (u - \tilde{x}_j)^2}{2} + \sum_{k \in \mathcal{N}_j} w_{jk} \rho(u - x_k) \right\}$

$e \leftarrow e + A_{*,j}(x_j - \tilde{x}_j)$

}

Fig. 4. Pseudocode for one voxel update, it comprises four steps: first, compute the column of forward projecting matrix; second, derive the parameters of the 1-D objective function; third, solve 1-D optimization problem; and fourth, update voxel and error sinogram.

a continuous function of the data [14]. We have found that the parameters $p = 2$, $q = 1.2$ and $c = 10$ Hounsfield units (HU) work well in practice [2]. The value $p = 2$ produces an approximately quadratic function for $|\Delta| \ll c$. This helps to preserve detail in low contrast regions such as soft tissue, and the value of $q = 1.2$ produces an approximately generalized-Gaussian prior [14] when $|\Delta| \gg c$. This helps preserve edges in high contrast regions such as interfaces between bone and soft tissue.

Applying the approximation of (2) and the prior distribution of (6), the MAP reconstruction is given by the solution to the following optimization problem:

$$\hat{x} = \arg\min_{x \geq 0} \left\{ \frac{1}{2}(y - Ax)^T D(y - Ax) + \sum_{\{j,k\} \in \Omega} w_{jk} \rho(x_j - x_k) \right\} \tag{8}$$

B. ICD Algorithm

We use the ICD algorithm to solve the problem of (8). One full iteration of the ICD algorithm works by updating voxels in sequence, until every voxel has been updated exactly once. Each voxel is updated so as to globally minimize the total objective function while fixing the remaining voxels. Formally, the update of the selected voxel x_j is given by

$$x_j \leftarrow \arg\min_{x_j \geq 0} \left\{ \frac{1}{2}(y - Ax)^T D(y - Ax) + \sum_{\{j,k\} \in \Omega} w_{jk} \rho(x_j - x_k) \right\} \tag{9}$$

This update can be computed efficiently by keeping track of the residual error sinogram defined by $e = Ax - y$. To do this, we first compute the first and second derivative of the negative log-likelihood term θ_1 and θ_2 as

$$\theta_1 = \sum_{i=1}^{N} d_i A_{ij} e_i \tag{10}$$

$$\theta_2 = \sum_{i=1}^{N} d_i A_{ij}^2 \tag{11}$$

where e_i is the ith element in the error sinogram and N is the total number of measurements in the sinogram. Then, derived from (8), one can write the minimization of the 1-D objective function for x_j explicitly as follows [21]:

$$x_j \leftarrow \arg\min_{u \geq 0} \left\{ \theta_1 u + \frac{\theta_2 (u - \tilde{x}_j)^2}{2} + \sum_{k \in \mathcal{N}_j} w_{jk} \rho(u - x_k) \right\} \tag{12}$$

where $\tilde{x}_j$ is the jth voxel's value before the update and $\mathcal{N}_j$ is the set of neighboring voxels of voxel j. We can bracket the minimizer of the 1-D objective function in the interval $[u_{\min}, u_{\max}]$ given by

$$u_{\max} = \max \left\{ \frac{\theta_2 \tilde{x}_j - \theta_1}{\theta_2}, x_k | k \in \mathcal{N}_j \right\} \tag{13}$$

$$u_{\min} = \max \left\{ \min \left\{ \frac{\theta_2 \tilde{x}_j - \theta_1}{\theta_2}, x_k | k \in \mathcal{N}_j \right\}, 0 \right\}. \tag{14}$$

This is because the ML term in (12) is minimized by $u = (\theta_2 \tilde{x}_j - \theta_1)/\theta_2$, and each of the prior terms $\rho(u - x_k)$ is minimized by $u = x_k$,

The pseudocode of Fig. 4 summarizes the steps for each voxel update. The first step is to compute the elements of the forward projection matrix for voxel j, that is, A_{*j}, the jth column of A. Second, we compute θ_1 and θ_2 using (10) and (11). Third, we compute the voxel's updated value by solving the 1-D optimization problem in (12). Finally, we update the error sinogram by forward projecting the update step $x_j - \tilde{x}_j$.

The FBP reconstruction typically provides a good initial condition for the ICD algorithm. This is because the FBP generally provides an accurate estimate of the low frequency components of the reconstruction. Higher frequency edge and texture details are generally not as accurate in the FBP images, but the ICD algorithm is known to have rapid convergence at high frequencies [21].

Each voxel update of ICD requires the computation of the 1-D minimization of (12). This update can be done using half-interval search, which is simple and robust, but relatively slow because it requires multiple steps to reach the desired precision of solution. Moreover, the number of steps required for a given precision may vary between voxel updates. We, therefore, propose a fast 1-D minimization algorithm for ICD in Section IV.

In each iteration of the conventional ICD algorithm, each voxel is updated once and only once. However, the order of voxel updates may vary with each iteration. We follow two rules in selecting the order of voxel updates. First, entire lines of voxels along the $\vec{e}_3$ axis are updated in sequence. As shown in Fig. 1, we refer to a line of voxels that falls at the same (j_1, j_2)

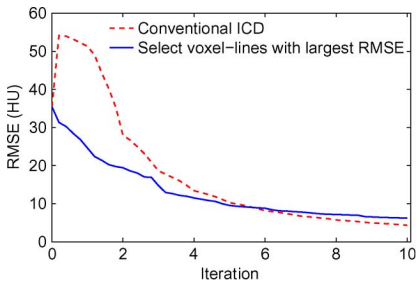


Fig. 5. Illustrates the fact that it is not necessarily good to use the convergence error as the only criterion for selection. The solid line shows the algorithm which always updates the voxel-line with the largest convergence error. However, in later iterations the convergence of this algorithm is actually slower than that of the conventional ICD algorithm.

position in the $(\vec{e}_1, \vec{e}_2)$ plane as a "voxel-line." The voxels in the voxel-line share the same geometry calculation in the $(\vec{e}_1, \vec{e}_2)$ plane as illustrated in Fig. 2(c), so updating all the voxels along a voxel-line saves computation [29]. In addition, updating the voxels sequentially helps to reduce memory bandwidth requirements. Second, we update the voxel-lines (j_1, j_2) in a random order, so that each voxel-line is updated once per iteration, and the order of selection is randomized with a uniform distribution [34]. For 2-D reconstruction, we compared this random selection method with another method which selects the pixels by raster scan order, and the experimental results indicated that selecting pixels in random order provides significantly faster convergence than raster order as the correlation between successive updates is reduced [35]. Therefore, the random update order is typically used in the conventional ICD algorithm.

III. Spatially Nonhomogeneous ICD (NH-ICD) Algorithm

The basic idea behind the spatially NH-ICD algorithm is updating some voxel-lines more frequently than others. The NH-ICD algorithm is motivated by the fact that the convergence error, which we define as the error between the current value and the fully converged value of a voxel, is not uniformly distributed across a reconstructed image. In fact, the convergence error tends to be distributed primarily near edges. However, the conventional ICD algorithm does not exploit this nonuniform distribution of error because each voxel must be updated exactly once per iteration. Therefore, we propose the NH-ICD algorithm to improve the convergence speed of ICD by focusing computational resources on the voxel-lines which can benefit most from updates.

In order to implement the NH-ICD algorithm, one must determine an ordering of the voxel updates which yields fast convergence to the MAP estimate. Ideally, it would be best to select the update ordering that results in the fastest overall convergence. However, determining this optimum ordering is very difficult since each voxel update can affect the result of subsequent updates.

In order to illustrate the difficulty in selecting the best update ordering, consider the plot of Fig. 5 which shows the root mean squared error (RMSE)[1] convergence for two different algorithms on a typical 3-D helical scan multislice CT data set. The dotted line shows the RMSE convergence of conventional ICD while the solid line shows the convergence of a nonhomogeneous update method[2] that always selects the voxel-line with the greatest mean squared error (MSE). Notice that the greedy selection method actually has slower convergence than conventional ICD. This is because fast convergence also requires that some voxels with lower MSE be updated, but these updates can be less frequent. Moreover, even if it worked well, this greedy selection method can not be practically implemented because it depends upon the knowledge of the converged MAP reconstruction to compute the MSE of each voxel-line.

With this example in mind, our nonhomogeneous ICD algorithm will be based on two concepts. First, we will compute a VSC for each voxel-line. The VSC will be used to determine which voxel-lines are in greatest need of update. Second, at a higher level, we will also need a VSA. The VSA will be designed to balance the need for repeated updates of some voxel-lines with the need for more uniform updating of all voxels lines. By balancing these two goals, we will be able to avoid the slow convergence shown in Fig. 5.

A. VSC

In this work, we choose the VSC to be related to the absolute sum of the update magnitudes along a voxel-line at its last visit. Intuitively, if a voxel-line had large updates, then it is likely that the voxels are far from their converged values, and could benefit from more frequent selection. Fig. 6 shows empirically that this conjecture is true. Fig. 6(a) shows an image of the 5% of voxel-lines whose updates were largest in the first iteration of the conventional ICD and Fig. 6(b) shows the 5% of voxel-lines with the largest MSE after the first iteration. The fact that the two images are highly correlated suggests that one can predict the RMSE using the update magnitude.

The total update magnitude is stored in a 2-D array corresponding to all the voxel-lines. The array is referred to as the update magnitude map, denoted by $u(j_1, j_2)$. The function $u(j_1, j_2)$ is initialized to zero, and with each ICD update of a voxel-line, the array is updated using the relation

$$u(j_1, j_2) \leftarrow \sum_{j_3=1}^{N_3} |x(j_1, j_2, j_3) - \tilde{x}(j_1, j_2, j_3)| \qquad (15)$$

where N_3 denotes the total number of voxels on the voxel-line, and the values $\tilde{x}(j_1, j_2, j_3)$ and $x(j_1, j_2, j_3)$ denote the values of voxels before and after the update, respectively. Because one full ICD iteration is required for all of u to take on nonzero values, the VSC is not available until after the first full ICD update. In Section III-C, we introduce the interleaved NH-ICD algorithm, which is designed to overcome this limitation. The

[1]The RMSE is computed between the current values and fully converged values of the voxels. We generate the fully converged images by running the reconstruction for a large number of iterations.

[2]For this algorithm, we define one iteration to be one "equit" as will be defined later in (25).

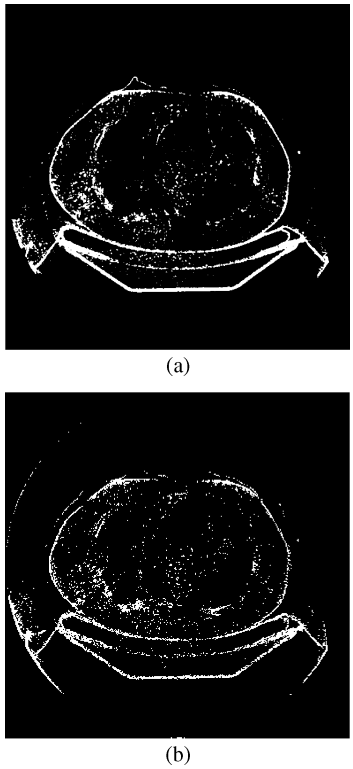


Fig. 6. Correlation between the update magnitude and the true convergence errors at the end of the first iteration of conventional ICD for a clinical body case. (a) Shows the top 5% voxel-lines with largest update magnitude. (b) Shows the top 5% voxel-lines with largest convergence error.

values of the VSC, denoted by $r(j_1, j_2)$, are then computed by applying a 2-D low pass filter to u so that

$$r(j_1, j_2) \leftarrow \sum_{s=-2}^{2} \sum_{w=-2}^{2} u(j_1 - s, j_2 - w) h(s, w) \qquad (16)$$

where the filter kernel h is a 5 by 5 Hamming window. We have found empirically that adjacent voxels have strong correlation in their MSE. Therefore, we use the low-pass filter h to encourage updates of local neighborhoods and to reduce the variation in the VSC.

B. VSA

In Fig. 5, we saw that an excessive focus on a small number of voxels can actually slow down the convergence. Therefore, an effective VSA must balance the need for repeated updating of some voxels with the need for improvement of the remainder of the image. In order to achieve this goal, we will incorporate two subprocedures in the VSA. The nonhomogeneous subprocedure selects voxel-lines that have large VSC values, and updates them frequently. Alternatively, the homogeneous subprocedures update all voxel-lines uniformly. By alternating between these two subprocedures, we can accelerate the convergence of the voxels with large VSC values while ensuring that all voxel-lines and their VSC values are updated.

Fig. 7(a) shows a flow diagram for the VSA. The algorithm starts with a homogeneous subprocedure, in which each voxel-line is updated exactly once, in a randomized order. This first homogeneous subprocedure ensures that the values of the update

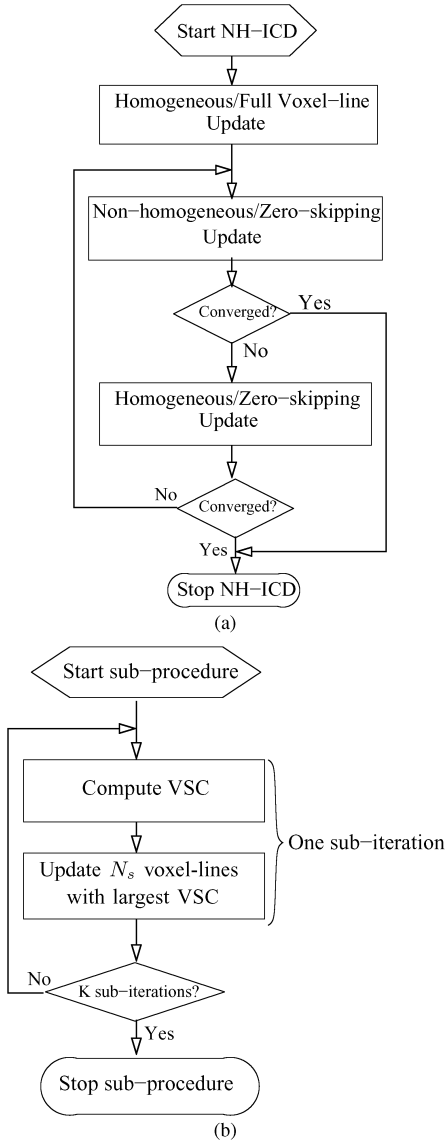


Fig. 7. Block diagram for the VSA used in NH-ICD. (a) Illustrates the top level VSA. (b) Shows the nonhomogeneous subprocedure.

magnitude map, u, are all initialized, and that the VSC can be computed for each voxel-line. Once the first homogeneous subprocedure is completed, the NH-ICD algorithm iterates between a nonhomogeneous subprocedure and a homogeneous subprocedure, and these two steps are repeated until the desired level of convergence is achieved.

When updating a voxel-line, we sometimes ignore zero-valued voxels in a process we call "zero-skipping." We have found that typically many voxels in air converge to a value of zero due to the positivity constraint, and can be skipped with little effect on the reconstructed image. Thus, if a voxel and its neighbors all have value zero, we skip the ICD update of that voxel. As shown in Fig. 7(a), zero-skipping is applied to all subprocedures except the first homogeneous subprocedure, in which we need to initialize the VSC.

Fig. 7(b) shows the flow diagram of the nonhomogeneous subprocedure. The subprocedure is composed of K subiterations. In each subiteration, first the values of the VSC are com-

```
Nonhomogeneous_Subprocedure(x, e, u, N_h){
  K ← ⌊λN_h / γN_1N_2N_3⌋
  for k = 1 to K do
    r ← ComputeVSC(u)      /* Using Equation (16) */
    L ← {(j_1, j_2)|r(j_1, j_2) ≥ T_γ}
    for each (j_1, j_2) ∈ L do
      (x, u) ←UpdateVoxelLine(j_1, j_2, x, e)
    end for
  end for
  return (x, u)
}
```

Fig. 8. Pseudocode specification of the nonhomogeneous subprocedure. Each iteration of the outer "for" loop represents a subiteration of the nonhomogeneous subprocedure. In each subiteration, the VSC is first computed by the function ComputeVSC(u), and then a set L is formed that contains γ fraction of voxel-lines with largest VSC values. The selected voxel-lines in L are then updated in a randomized order.

```
InterleavedNH-ICD(y) {
  x ← FBP images
  e ← Ax − y
  Partition voxel-lines into subsets S_0, S_1, S_2, and S_3
  for g = 0 to 3 do
      /* Perform partial homogeneous sub-procedure */
    for each (j_1, j_2) ∈ S_g do
      (x, u) ← UpdateVoxelLine(j_1, j_2, x, e)
    end for
      /* Perform partial non-homogeneous sub-procedure */
    N_h ← (1/4)N_1N_2N_3
    (x, u) ←Nonhomogeneous_Subprocedure(x, e, u, N_h)
  end for
  repeat
    (x, u, N_h) ← Homogeneous_Subprocedure(x, e)
    (x, u) ← Nonhomogeneous_Subprocedure(x, e, u, N_h)
  until Image converged to desired level
}
```

Fig. 9. Pseudocode of the complete NH-ICD algorithm. Notice that the NH-ICD algorithm starts by first running the interleaved subprocedures. The interleaved subprocedures start with a partial homogeneous subprocedure that updates a subset of voxel-lines, and then it is followed by a partial nonhomogeneous subprocedure with a limited number of subiterations. After updating all subsets, the algorithm then alternates between full homogeneous and nonhomogeneous subprocedures.

puted from the current update magnitude map using (16). Next a fraction of the voxel-lines is updated by selecting the N_s voxel-lines with the largest values of the VSC in $r(j_1, j_2)$. Once these N_s voxels-lines are updated, the VSC is recomputed in the beginning of the next subiteration, and this process is repeated K times.

The number of voxel-lines visited in each subiteration is equal to the total number of voxel-lines in the image multiplied by a factor $0 \leq \gamma < 1$, that is, $N_s = \gamma N_1 N_2$. The number of subiterations is then computed by $K = \lfloor \lambda N_h / \gamma N_1 N_2 N_3 \rfloor$, where $\gamma N_1 N_2 N_3$ is the number of voxels updated in each subiteration, N_h is the number of voxels updated in the previous homogeneous subprocedure, and $\lambda > 0$ is a user selected parameter. N_h may be less than $N_1 N_2 N_3$ due to zero-skipping. We typically use the values $\gamma = 0.05$ and $\lambda = 1$, which we have found to result in consistently fast convergence.

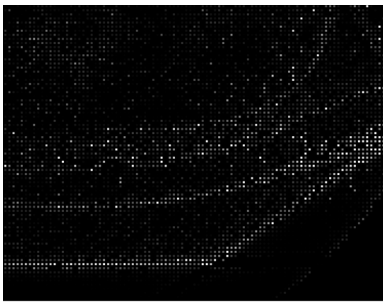
The pseudocode in Fig. 8 specifies the nonhomogeneous subprocedure in detail. In the beginning of the subprocedure, we compute the number of subiterations that need to be performed. In each subiteration, the function ComputeVSC(u) first computes the VSC using (16). Then, we form a set L that contains all the voxel-lines (j_1, j_2) with $r(j_1, j_2) \geq T_\gamma$, where T_γ is the threshold for the γ fraction of voxel-lines with the largest VSC values. Next, the voxel-lines in L are updated in a randomized order using the UpdateVoxelLine(j_1, j_2, x, e, u) function. The function UpdateVoxelLine(j_1, j_2, x, e, u) updates the voxels on the selected voxel-line (j_1, j_2) in sequence and also computes $u(j_1, j_2)$ using (15). In the next subiteration, the refreshed update magnitude map u is used to compute the VSC. A voxel-line can be updated at most once in one subiteration, but if it produces a large update magnitude, it may be selected in subsequent subiterations. Therefore, a voxel-line can be updated as many as K times during a single application of the nonhomogeneous subprocedure.

C. Interleaved NH-ICD

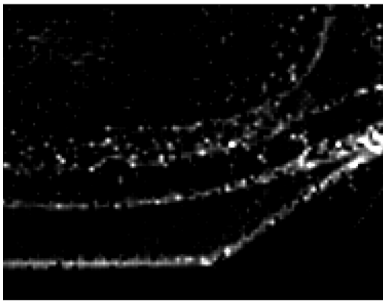
Interleaved NH-ICD is intended to allow nonhomogeneous subprocedures before the completion of the first full homogeneous update, to exploit as early as possible the knowledge of the locations where updates are most needed. To do this, we update a densely interleaved set of voxel-lines, so that a value for VSC can be computed at all voxel locations. This allows the first nonhomogeneous subprocedure to be run after a fraction of an iteration.

The pseudocode in Fig. 9 specifies the interleaved NH-ICD algorithm. Before the reconstruction starts, we partition the set of all voxel-lines into four interleaved subsets, $S_0 = \{(j_1, j_2) | j_1 = 2p, j_2 = 2q, \}$, $S_1 = \{(j_1, j_2) | j_1 = 2p+1, j_2 = 2q\}$, $S_2 = \{(j_1, j_2) | j_1 = 2p, j_2 = 2q+1\}$ and $S_3 = \{(j_1, j_2) | j_1 = 2p+1, j_2 = 2q+1\}$, in which p and q are positive integers. The interleaved NH-ICD algorithm starts by performing a homogeneous update of only the voxel-lines in S_0. The partial nonhomogeneous subprocedure then updates the voxel-lines with the largest VSC. The number of voxel updates performed in this partial nonhomogeneous subprocedure is proportional to the number in the previous partial homogeneous procedure. This process is repeated four times until each subset has been updated once, after which the NH-ICD algorithm alternates between full homogeneous and nonhomogeneous subprocedures until convergence is achieved.

In the partial nonhomogeneous subprocedure, we compute the VSC for each voxel-line using partially initialized update magnitude maps. Therefore, the updates in the partial nonhomogeneous subprocedure are not limited to the subset S_0. Since the voxel-lines are uniformly sampled in the partial homogeneous subprocedures, the filtering step of (16) can be viewed as a simple interpolation which fills in the values in the VSC for the voxel-lines that have not yet been updated. Fig. 10(a) shows the update magnitude map after the first partial homogeneous subprocedure, in which a quarter of voxel-lines have been updated. The interpolated VSC is shown in Fig. 10(b).



(a)



(b)

Fig. 10. Shows part of the update magnitude map after the first partial homogeneous subprocedure in (a), and the corresponding VSC in (b). The filtering step of equation (16) fills in the values in the VSC for the voxel-lines that have not been updated yet.

IV. Fast 1-D Optimization Algorithm

The ICD algorithm requires solving the 1-D optimization problem in (12) for each voxel update. Due to the complex form of the potential function $\rho(\Delta)$ in (7), this optimization step can account for a substantial fraction of the total computation time in an ICD iteration. In this section, we propose a fast algorithm for approximately solving the 1-D optimization problem of (12). This fast 1-D update algorithm is based on a functional substitution (FS) approach which uses a simplified upper bound to replace the true objective function [24], [30]–[32]. Importantly, this FS approach reduces computation while also guaranteeing monotone convergence of the objective function.

We form the substitute function by replacing each function $\rho(u - x_k)$ in the terms of (12) with a new function $f_{jk}(u)$. The function $f_{jk}(u)$ is chosen to have a simple quadratic form

$$f_{jk}(u) = a_{jk}u^2 + b_{jk}u + c_{jk} \tag{17}$$

so that it is easy to minimize, and the coefficients a_{jk}, b_{jk} and c_{jk} are chosen to meet the following two important constraints:

$$f_{jk}(\tilde{x}_j) = \rho(\tilde{x}_j - x_k) \tag{18}$$

$$f_{jk}(u) \geq \rho(u - x_k), \quad \forall u \in [u_{\min}, u_{\max}] \tag{19}$$

where $\tilde{x}_j$ is the value of the jth voxel before it is updated.

The motivation behind these two constraints is illustrated in Fig. 11. At the value $u = \tilde{x}_j$, the true function and substitute function are equal, but for all other values of $u \in [u_{\min}, u_{\max}]$ the substitute function, $f_{jk}(u)$, is greater than the true function. If the functions are continuously differentiable, and $\tilde{x}_j \in (u_{\min}, u_{\max})$, this must also imply that the functions are tangent

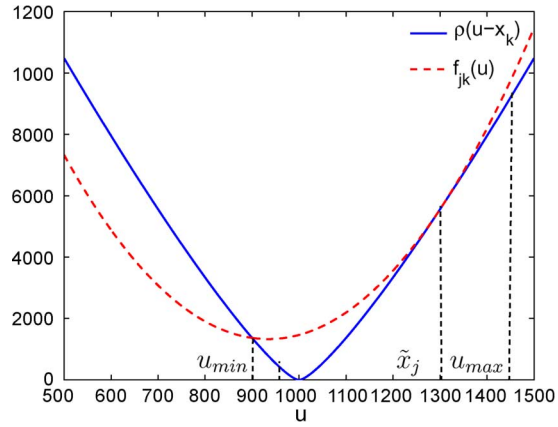


Fig. 11. Substitute function equals the true objective function at $u = \bar{x}_j$ and upper bounds the objective function on the interval $[u_{\min}, u_{\max}]$. Therefore, when the substitute function is minimized, the underlying objective function is guaranteed to be reduced.

$$
\begin{array}{l}
(a_{jk}, b_{jk}, c_{jk}) \leftarrow \text{ComputeParameters}(\tilde{x}_j, x_k, u_{min}, u_{max}) \\
\{ \\
\quad \Delta_0 \leftarrow \tilde{x}_j - x_k \\
\quad \Delta_{min} \leftarrow u_{min} - x_k \\
\quad \Delta_{max} \leftarrow u_{max} - x_k \\
\quad T \leftarrow \begin{cases} -\Delta_0, & if\ |\Delta_0| \leq \min\{|\Delta_{min}|, |\Delta_{max}|\} \\ \Delta_{min}, & if\ |\Delta_{min}| \leq \min\{|\Delta_0|, |\Delta_{max}|\} \\ \Delta_{max}, & if\ |\Delta_{max}| \leq \min\{|\Delta_0|, |\Delta_{min}|\} \end{cases} \\
\quad a \leftarrow \begin{cases} \frac{\rho(T)-\rho(\Delta_0)}{(T-\Delta_0)^2} - \frac{\rho'(\Delta_0)}{T-\Delta_0} & \text{If } \Delta_0 \neq 0 \\ \frac{\rho''(0)}{2} & \text{If } \Delta_0 = 0 \end{cases} \\
\quad b \leftarrow \rho'(\Delta_0) - 2a\tilde{x}_j \\
\quad c \leftarrow \rho(\Delta_0) - a\tilde{x}_j^2 - b\tilde{x}_j \\
\quad \textbf{return}\ \ (a, b, c) \\
\}
\end{array}
\tag{21}
$$

Fig. 12. Algorithm for computing the parameters of f_{jk}.

to each other at $u = \tilde{x}_j$, so that their derivatives must also be equal

$$f'_{jk}(\tilde{x}_j) = \rho'(\tilde{x}_j - x_k). \tag{20}$$

Our objective is to determine values of the coefficients a, b, and c which ensure that the constraints of (18) and (19) are satisfied. In fact, we can achieve this goal by computing coefficients using the algorithm specified in Fig. 12. The following theorem, proved in Appendix A, guarantees that for a broad class of potential functions, the coefficients computed by this algorithm, satisfy the conditions of (18) and (19).

Theorem 1: If $\rho(\Delta)$ is continuously differentiable and satisfies the following conditions:

1) $\rho(\Delta)$ is an even function;
2) $\rho(\Delta)$ is strictly convex;
3) $\rho'(\Delta)$ is strictly concave for $\Delta > 0$ and strictly convex for $\Delta \leq 0$;
4) $\rho''(0)$ exists.

The parameters a, b, and c are computed according to the algorithm given in Fig. 12, then the conditions in (18) and (19) hold true. Similarly to [32], we can show the given a is optimal in the sense that it is the smallest number satisfying these conditions.

The four conditions of Theorem 1 are satisfied by a variety of potential functions, among them the q-GGMRF prior used in this paper with $p = 2$ as illustrated in Fig. 3. Moreover, if we replace equation (21) in the algorithm with $T \leftarrow -\Delta_0$, then the computed coefficients guarantee that $f_{jk}(u) \geq \rho(u)$ on $(-\infty, \infty)$. This result is useful if the iterative algorithm being used does not allow us to easily find the interval $[u_{\min}, u_{\max}]$ for each voxel.

If we replace each function $\rho(u - x_k)$ in (12) with $f_{jk}(u)$, then instead of minimizing the original objective function, we minimize the substitute function on the interval $[u_{\min}, u_{\max}]$ as shown in the following:

$$x_j \leftarrow \underset{u\in[u_{\min},u_{\max}]}{\arg\min} \left\{ \theta_1 u + \frac{\theta_2(u-\tilde{x}_j)^2}{2} + \sum_{k\in\mathcal{N}_j} w_{jk} f_{jk}(u) \right\} = \text{clip}\{u^*, [u_{\min}, u_{\max}]\} \tag{22}$$

where u^* is the global minimizer of the quadratic substitute function given by

$$u^* = \frac{-\sum_{k\in\mathcal{N}_j} w_{jk} b_k + \theta_2 \tilde{x}_j - \theta_1}{2\sum_{k\in\mathcal{N}_j} w_{jk} a_{jk} + \theta_2} \tag{23}$$

and the function $\text{clip}\{u^*, [u_{\min}, u_{\max}]\}$ clips the variable u^* to either $u_{\min}$ or $u_{\max}$ if it falls out of the interval $[u_{\min}, u_{\max}]$.

The update given by (22) tends to be conservative. Therefore, we use an over-relaxation method to encourage larger update steps in order to improve the convergence speed of the algorithm. Using over-relaxation, we compute the update value of x_j by

$$x_j \leftarrow \text{clip}\{\tilde{x}_j + \alpha(u^* - \tilde{x}_j), [u_{\min}, u_{\max}]\} \tag{24}$$

where $1 < \alpha < 2$. Since the substitute function is quadratic, values of α in the range of (0,2) are still guaranteed to strictly decrease the substitute function's value, and therefore also the true cost function's value.

We summarize the proposed 1-D optimization algorithm in the pseudocode in Fig. 13. First, we compute $u_{\min}$ and $u_{\max}$. Second, the parameters a_{jk} and b_{jk} are computed in the for loop, for each voxel pair (j, k) in the neighborhood. Third the global minimizer u^* of the substitute function is computed using the closed form formula. Finally, we find the update value of x_j by over-relaxing the solution using (24), and then clip the solution if necessary.

```
FSMethod(θ1, θ2, x, j){
  x̃_j ← x_j
  u_max ← max{(θ2 x̃_j − θ1)/θ2, x_k | k ∈ N_j}
  u_min ← max{min{(θ2 x̃_j − θ1)/θ2, x_k | k ∈ N_j}, 0}
  for k ∈ N_j do
    (a_jk, b_jk, c_jk) ← ComputeParameters(x̃_j, x_k, u_min, u_max)
  end for
  u* ← (−Σ_{k∈N_j} w_jk b_jk + θ2 x̃_j − θ1) / (2 Σ_{k∈N_j} w_jk a_jk + θ2)
  x_j ← clip{x̃_j + α(u* − x̃_j), [u_min, u_max]}
  return x_j
}
```

Fig. 13. Pseudocode for functional substitution (FS) method. The algorithm computes the parameters of the substitute function that upper bounds the true objective function on the interval $[u_{\min}, u_{\max}]$. We then find the minimum u^* of the substitute function using a closed form formula. The final solution is over-relaxed by a factor α and then clipped to the interval $[u_{\min}, u_{\max}]$.

V. Experimental Results

In this section, we apply the NH-ICD algorithm to clinical reconstructions. The data are acquired from a multislice GE Lightspeed VCT scanner. All axial reconstructed images are of size 512×512 with each slice having thickness of 0.625 mm. We use three clinical data sets that cover different anatomy: a abdomen scan of 95 slices in 700 mm field of view (FOV) with normalized pitch 1.375, a head scan of 155 slices in 480 mm FOV with pitch 1, and an abdomen scan of 123 slices in 500 mm FOV with normalized pitch 1.375. Fig. 14 shows a single axial slice from data set 1 which has been reconstructed using FBP and the conventional ICD algorithm with 10 iterations. In our objective function, we choose w_{jk} to be inversely proportional to the distance between voxels j and k. We adjust the scale of w_{jk} to achieve a balance between resolution and noise in the reconstruction. We implemented all algorithms on a standard 2.0 GHz clock rate 8 core Intel processor workstation running the Linux operating system. The algorithm was parallelized so that each core was responsible for updating a sequence of slices along the $\vec{e}_3$ axis. All the cores simultaneously work on the same voxel-line. Once a voxel-line is selected, we distribute voxel updates onto each core. Moreover, we guarantee the voxels being updated in parallel are far apart so that they do not share any sinogram data and can be updated independently.

We first investigated the computational cost reduction associated with the functional substitution update algorithm described in Section IV. In this experiment, we use a constant over-relaxation factor $\alpha = 1.5$ for the functional substitution method. The functional substitution method performs only one update for each voxel. The half-interval method performs multiple iterations until the search interval is less than 1 HU. In Table I, the first row compares the average computation time of the 1-D optimization using the functional substitution and the half interval methods for a single voxel. The second row compares the total computation time required for updating one voxel on a voxel-line, which includes the time required for the forward model calculation as well as reading and updating the sinogram data. The computation time is measured by averaging all the voxel update times from 10 iterations of the conventional ICD algorithm. The results show that the functional substitution algorithm on average reduces the computation time of the 1-D optimization by 85%. Consequently, the total computation time per voxel update is reduced by approximately 20% on this computer.

Since the two algorithms do not reach same numerical solution at each voxel update, we would also compare the convergence speed. The convergence speed of functional substitution and half-interval methods as well as other algorithms discussed

(a)

(b)

Fig. 14. Single slice of the data set 1. (a) Shows the FBP reconstruction. (b) Shows the conventional ICD reconstruction with 10 iterations. The FBP reconstruction is used as the initial estimate for the iterative reconstruction.

TABLE I
TABLE COMPARING THE AVERAGE COMPUTATION TIME BETWEEN HALF-INTERVAL SEARCH AND FUNCTIONAL SUBSTITUTION METHOD FOR A SINGLE VOXEL ON A VOXEL-LINE

Method	Half-interval Search	Functional Substitution
1D Optimization	0.043 ms	0.00645 ms
Voxel Update	0.15 ms	0.12 ms

in this paper will be compared in the following. The results show that the functional substitution method also improves the convergence speed of the ICD algorithm.

Next, we compare the speed of the following five algorithms:

1) ICD/HI—conventional ICD using half interval 1-D optimization;
2) ICD/FS—vonventional ICD using functional substitution 1-D optimization;
3) ICD/FS/zero skipping—the ICD/FS algorithm with zero skipping;
4) NH-ICD—the NH-ICD algorithm using functional substitution and zero-skipping with $\gamma = 0.05$ and $\lambda = 1$;
5) NH-ICD/Interleave—the NH-ICD algorithm with interleaving.

In order to compare the speed of convergence for each of these methods, we need measures of both convergence and computation. We employ two measures of convergence. The first measure is the value of the MAP cost function being minimized. The second measure is the RMSE difference between the current image and its converged value after 50 iterations of the conventional ICD algorithm. We also use two measures of computation. The first measure, called equivalent iteration or "**equit**," is based on the total number of voxel updates and is defined as

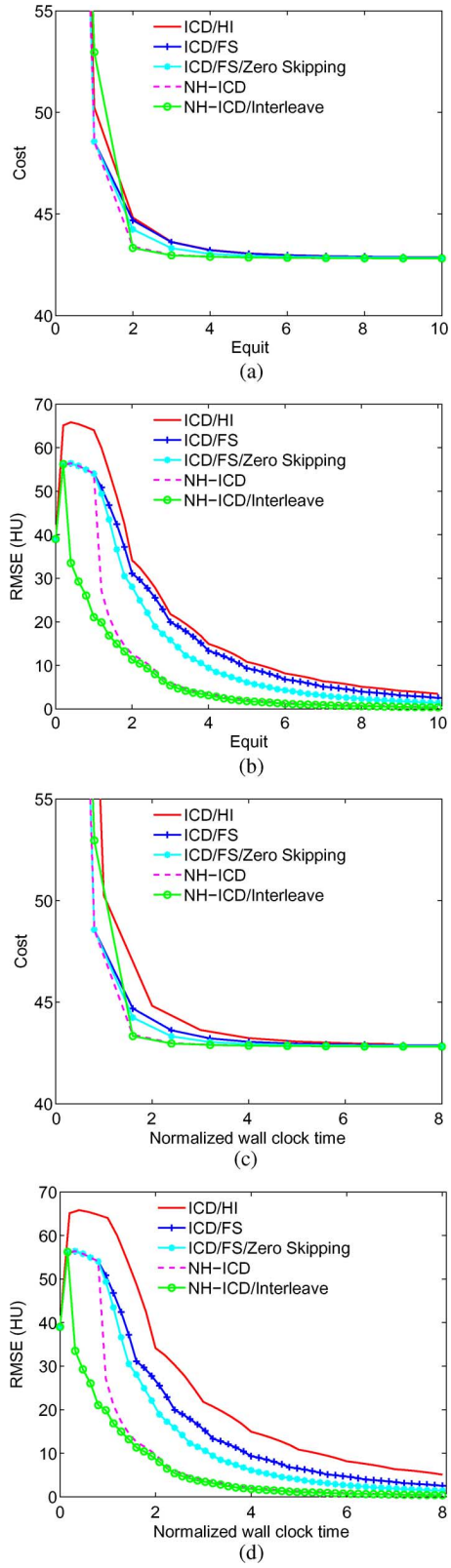


Fig. 15. Comparison of the convergence speed of different algorithms. The cost function and RMSE are computed by averaging from all three clinical data reconstructions. (a) and (b) Show the convergence of the cost function and RMSE versus equits. (c) and (d) Show the convergence of the cost function and RMSE versus normalized wall clock time. The results show that the interleaved NH-ICD algorithm significantly improves the speed of the reconstruction.

$$\text{equit} = \frac{\text{number of voxel updates}}{\text{total number of voxels in the FOV}}. \quad (25)$$

By this definition, one conventional ICD iteration requires one equit of computation. The subprocedures in NH-ICD generally require less than one equit due to zero-skipping. Also, one equit of NH-ICD may update some voxels multiple times whereas other voxels are not visited. The second measure of computation is the normalized wall clock time, which is computed as the actual wall clock time divided by the computation time required for a single full iteration of ICD/HI. The convergence plots are based on the averaged cost function and RMSE of all three clinical cases. To compute the averages, we evaluated the cost function every equit and the RMSE every 0.2 equit for each data set. We then averaged the cost function and RMSE values at the same number of equits from the three data sets to form the aggregate convergence plot.

Fig. 15(a) and (b) shows the convergence of the averaged cost function and RMSE versus equits. The convergence plots show that all the algorithms converge to the same asymptotic cost function and RMSE, which is expected for this convex optimization problem. Generally, we found that RMSE is a better indicator of the visual convergence of the reconstruction than the cost function. For example, we found that images with similar RMSE typically have similar visual quality, while reconstructions with close cost function values might have very different visual appearance. First note that the convergence speed in equits of ICD/FS is consistently faster than ICD/HI. Perhaps this is surprising since ICD/HI computes the exact ICD update for each voxel. However, ICD/HI initially suffers from overshoot due to the greedy nature of the algorithm, and ICD/FS avoids this pitfall by effectively reducing the size of changes in these initial updates. The NH-ICD algorithm dramatically reduces the RMSE in the first nonhomogeneous subprocedure, which occurs roughly in between equits 1 and 2, and maintains a fast convergence speed afterward. The interleaved NH-ICD algorithm provides the fastest convergence speed in terms of RMSE. Especially after the first equit, the RMSE of the interleaved NH-ICD algorithm is significantly smaller than that of all other algorithms. Although the asymptotic convergence speed of interleaved NH-ICD is similar to the NH-ICD algorithm, the interleaved NH-ICD algorithm has the advantage of eliminating the overshoots in the early stage of the reconstruction.

Fig. 15(c) and (d) shows the plots of cost function and RMSE versus the normalized wall clock time in order to compare the overall speed improvement contributed by both the fast voxel update and fast convergence speed. For example, to achieve an RMSE under 5 HU, it takes on average eight iterations for the conventional ICD algorithm, while the computational cost for the interleaved NH-ICD algorithm is equivalent to only 2.5 iterations of conventional ICD. Therefore, the plots show that the proposed interleaved NH-ICD algorithm improves the reconstruction speed by approximately a factor of 3.

Fig. 16 compares the reconstructions of the ICD/HI in the first and second rows and the interleaved NH-ICD in the third and fourth rows with 1, 3, 7, and 10 equits. The RMSE values of the reconstructions are also listed in the caption. In Fig. 16(a), the first iteration of ICD/HI creates overshoots which appear as salt and pepper noise and gradually disappear after several more iterations. Although most areas in the reconstruction do not change significantly after three equits, the conventional ICD algorithm still iterates on all the voxels instead of focusing on the visible errors near the edge of the patient. On the other hand, the interleaved NH-ICD algorithm visually converges much faster. For example, comparing at a fixed number of equits, interleaved

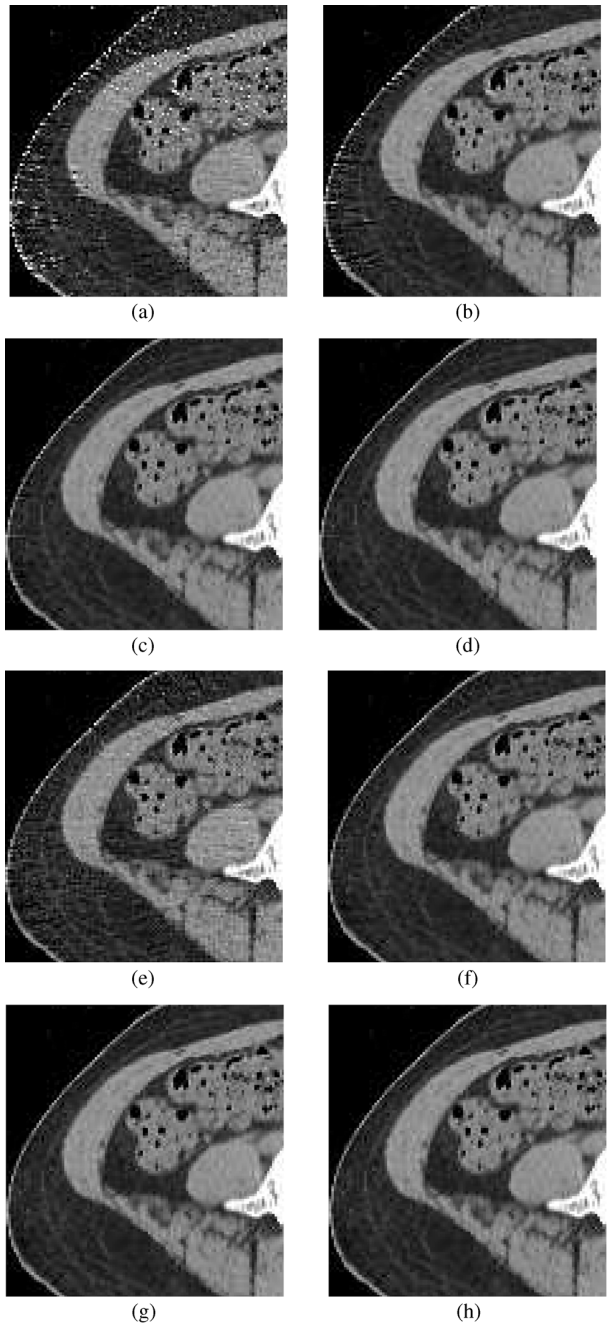


Fig. 16. First and second rows show the reconstruction using ICD/HI algorithm with (a) 1 equit, RMSE = 51.3 HU (b) 3 equits, RMSE = 18.6 HU, (c) 7 equits, RMSE = 6.7 HU, and (d) 10 equits, RMSE = 4.3 HU. The third and fourth rows show the reconstruction using Interleaved NH-ICD with (e) 1 equit, RMSE = 14.7 HU, (f) 3 equits, RMSE = 3.8 HU (g) 7 equits, RMSE = 0.91 HU and (h) 10 equits, RMSE = 0.45 HU. The images show that NH-ICD with interleaving can achieve the same visual quality of conventional ICD with significantly fewer voxel updates.

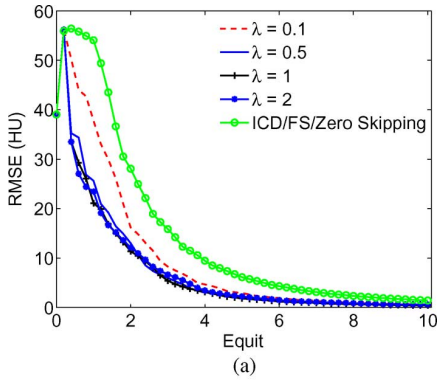


Fig. 17. Convergence plot of the interleaved NH-ICD algorithm with different choice of parameters λ and γ. In (a), we fix $\gamma = 0.05$, and vary λ. In (b), we fix $\lambda = 1$, and vary γ. The plots show that the interleaved NH-ICD algorithm is not sensitive to the choice of λ and γ.

NH-ICD reconstructions in Fig. 16(e) and (f) are visually better than the ICD/HI reconstructions in (a) and (b). Comparing at a fixed RMSE level, the interleaved NH-ICD reconstruction in (e) and (f) with one and three equits, respectively, have smaller RMSE than the ICD/HI reconstruction in (b) and (d) with three and ten equits. In this case, the interleaved NH-ICD is more than three times faster in reaching the same RMSE level.

Fig. 17 studies the impact of the VSA parameters, γ and λ, on the convergence speed of the interleaved NH-ICD. Fig. 17(a) shows the RMSE convergence of the first data set with λ varying from 0.1 to 2 and γ fixed at 0.05. Fig. 17(b) shows the RMSE convergence plots with γ varying from 0.01 to 0.50 and fixed $\lambda = 1$. The convergence plots indicate that the convergence speed of the NH-ICD algorithm is not sensitive to the choice of the parameters when we select λ from 0.5 to 2 and γ from 0.01 to 0.15. We typically use $\lambda = 1$ and $\gamma = 0.05$, which consistently performed well. Intuitively, when λ approaches 0 or γ approaches 1, the algorithm approaches the conventional ICD algorithm with zero skipping. This trend is also shown in Fig. 17(a) and (b).

Finally, we conjecture that it is possible to further improve the convergence speed of the NH-ICD algorithm by reducing the length of the voxel-lines, updating smaller segments independently. To verify, we divide the voxels with the same (i_1, i_2) coordinates into K voxel-lines that are equally spaced along the e_3 axis. The mechanism to select voxel-lines for update is exactly as described in Section III, except now we need to manage K times the number of voxel-lines and keep track of their update magnitudes. In Fig. 18, we compare the convergence speed of the NH-ICD algorithm with varying voxel-line lengths (VL). In this experiment, we use the data of case 3. Instead of reconstructing 123 slices as in previous experiments, we now reconstruct a wider coverage of 363 slices to illustrate the impact of the voxel-line length. We first consider the case with a full-length voxel-line of 363 voxels, then reduce the length by approximate factors 4 and 16, and finally consider only one voxel. First, the results confirm that, even with long voxel-lines, the NH-ICD algorithm still converges significantly faster than the conventional ICD algorithm. Second, we notice that as the length of the voxel-line decreases, the convergence speed of the NH-ICD algorithm increases. For example, by reducing the voxel-line length to 1, we can save an average of 0.7 equits over the RMSE range of 1 to 10 HU. However, using smaller voxel-lines tends to increase the computation time per voxel update. In general, one needs to choose the length of the voxel-line to achieve a balance between the convergence speed and the computational efficiency of the voxel update.

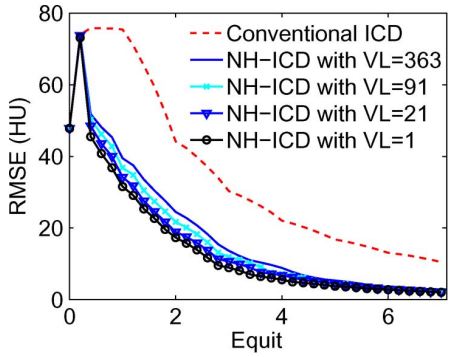


Fig. 18. In this experiment, we divide a voxel-line into smaller segments of length VL that can be updated independently. The results show that we can improve the convergence speed of NH-ICD by reducing the length of the voxel-line.

VI. Conclusion

In this paper, we have presented a spatially nonhomogeneous ICD algorithm with fast 1-D optimization. The method works by focusing computation to the most important areas of the reconstruction. The experiments on a variety of clinical data sets show that the proposed algorithm can accelerate the reconstruction by a factor of approximately three on average. This improved convergence speed may be used to either reduce computation for a fixed level of quality, or improve quality for applications with fixed computational resources.

Appendix
Proof of the Theorem

Theorem: If $\rho(\Delta)$ is continuously differentiable and satisfies the following conditions:

1) $\rho(\Delta)$ is an even function;
2) $\rho(\Delta)$ is strictly convex;
3) $\rho'(\Delta)$ is strictly concave for $\Delta > 0$ and strictly convex for $\Delta \leq 0$;
4) $\rho''(0)$ exists.

The parameters a, b, and c are computed according to the algorithm given in Fig. 12, then the conditions in (18) and (19) hold true. Moreover, a is the smallest number satisfying (18) and (19).

Proof: In order to simplify the notation, in the following we suppress the dependency of the index on j and k. To do this, we define function f_s by changing variables, that is, $f_s(u - x_k) = f_{jk}(u) = a_{jk}u^2 + b_{jk}u + c_{jk}$. Let $\Delta = u - x_k$ and $\Delta_0 = \tilde{x}_j - x_k$, then to show $f_{jk}(u)$ satisfies (18) and (19) is equivalent to show

$$f_s(\Delta_0) = \rho(\Delta_0) \tag{26}$$

$$f_s(\Delta) \geq \rho(\Delta), \quad \forall \Delta \in [\Delta_{\min}, \Delta_{\max}] \tag{27}$$

where $\Delta_{\min} = u_{\min} - x_k$ and $\Delta_{\max} = u_{\max} - x_k$.

It is easy to verify that the parameters a_{jk}, b_{jk} and c_{jk} computed using the algorithm in Fig. 12 satisfies (26) and the following equations:

$$f_s'(\Delta_0) = \rho'(\Delta_0) \tag{28}$$

$$f_s(T) = \rho(T). \tag{29}$$

In fact, we can use (26), (28) and (29) to derive the formula of a, b and c in Fig. 12.

Our objective for the remaining part of this proof is to show that, first, the inequality 27 holds, and second, a is the smallest number that satisfies (27). In order to show the inequality (27) holds, we construct the function $y_T(\Delta) = f_s(\Delta) - \rho(\Delta)$, so that we only need to show $y_T(\Delta) \geq 0$ on $[\Delta_{\min}, \Delta_{\max}]$. By the previous assumptions on $\rho(\Delta)$ and $f_s(\Delta)$, $y_T(\Delta)$ has the following properties which will be used later in this proof:

1) $y_T(\Delta_0) = y_T'(\Delta_0) = y_T(T) = 0$
2) $y_T'(\Delta)$ is strictly convex for $\Delta > 0$ and strictly concave for $\Delta \leq 0$ (since $f_s'(\Delta)$ is a linear function)

In equation (21), we have three different cases for the value of T. We now show the inequality $y_T(\Delta) \geq 0$ holds on $[\Delta_{\min}, \Delta_{\max}]$ for each case:

In the first case when $T = -\Delta_0$, let $y_0(\Delta) = f_s(\Delta) - \rho(\Delta)$. It is easy to verify that in this case $f_s(\Delta)$ is an even function. $\rho(\Delta)$ is also even, which results in $y_0(\Delta)$ as an even function and $y_0'(\Delta)$ as an odd function. By the properties of odd functions, we can derive that $y_0'(0) = 0$ and $y_0'(-\Delta_0) = -y_0'(\Delta_0) = 0$.

Let us first consider $\Delta \in [0, \infty)$. There are two subcases, First, if $\Delta_0 \neq 0$, since $y_0'(\Delta)$ is strictly convex on $[0, \infty)$, $y_0'(\Delta) \leq 0$ on $(0, \Delta_0)$, and $y_0'(\Delta) \geq 0$ on (Δ_0, ∞), which is illustrated in Fig. 19(a) and (b). Therefore, for $\Delta \in (0, \Delta_0)$, we apply the fact that $y_0(\Delta_0) = 0$ to yield $y_0(\Delta) = -\int_{\Delta}^{\Delta_0} y_0'(s)ds > 0$. Similarly, for $\Delta \in (\Delta_0, \infty)$, we have $y_0(\Delta) = \int_{\Delta_0}^{\Delta} y_0'(s)ds > 0$. Therefore, $y_0(\Delta) \geq 0$ for $\Delta > 0$. Second, if $\Delta_0 = 0$, it is easy to see that in this case $f_s''(0) = \rho''(0)$, therefore, $y_0''(0) = 0$. For $\Delta \in (0, \infty)$, by convexity of $y_0'(\Delta)$, we have $y_0'(\Delta) > y_0'(0) + y_0''(0)\Delta = 0$, and, thus, $y_0(\Delta) = \int_0^{\Delta} y_0'(s)ds > 0$. Symmetrically, it can be shown that $y_0(\Delta) \geq 0$ for $\Delta \in (-\infty, 0)$.

Now, we consider the second case when $\Delta_0 > 0$ and $-\Delta_0 < T < \Delta_0$. First, we want to show that $y_T'(\Delta) \geq y_0'(\Delta)$ for $\Delta \leq \Delta_0$, where $y_0'(\Delta)$ is defined in the previous case. Let $f_{s0}(\Delta)$ be the substitute function for the previous case when $T = -\Delta_0$. Intuitively, as illustrated in Fig. 19(c), as $T \to \Delta_0$, $f_s'(\Delta)$ rotates clockwise around the fixed point $(\Delta_0, \rho(\Delta_0))$. Thus, $f_s(\Delta) \geq f_{s0}(\Delta)$ for $\Delta \leq \Delta_0$. This can be rigorously proved by contradiction as follows:

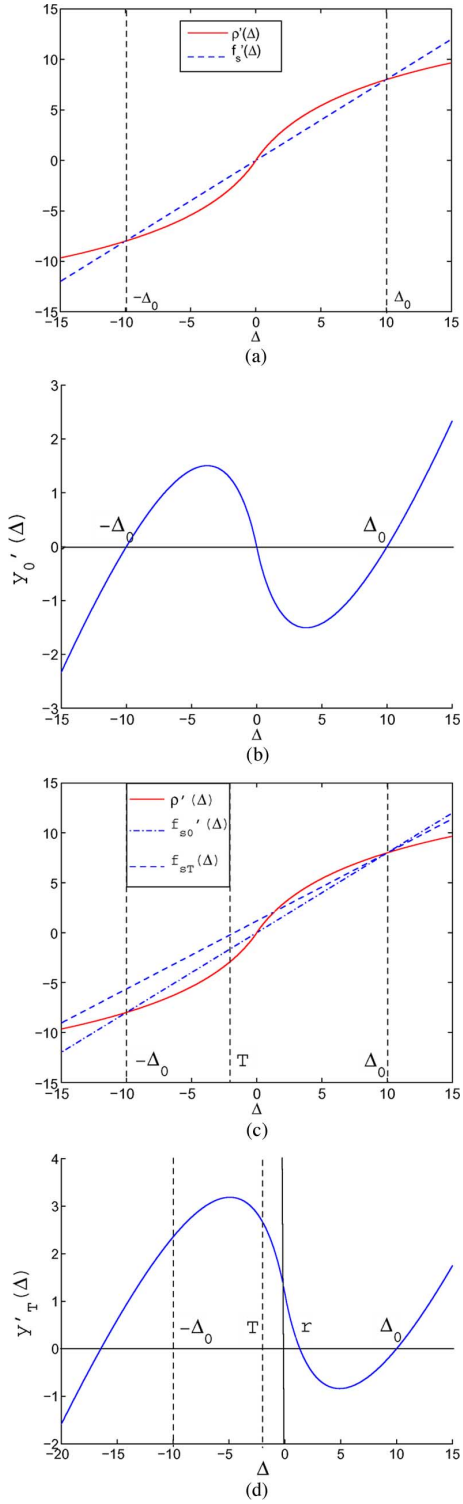


Fig. 19. Illustrates the proof for two cases. In the first case when $T = -\Delta_0$, (a) shows $\rho'(\Delta)$ and $f_s'(\Delta)$ intersects at $\Delta = -\Delta_0, 0, \Delta$. Consequently, $y_0'(\Delta)$, as shown in (b), has three roots at $\Delta = -\Delta_0, 0, \Delta_0$. In the second case when $\Delta_0 > 0, -\Delta_0 < T < \Delta_0$, (c) shows $\rho'(\Delta)$ and $f_s'(\Delta)$ also intersects at three points; and (d) shows $y_T'(\Delta)$ has three roots at $\Delta = -\Delta_0, r, \Delta_0$, where $r \in (T, \Delta_0)$.

We observe that $f'_s(\Delta_0) = f'_{s0}(\Delta_0)$, i.e., two lines intersect at $\Delta = \Delta_0$. Therefore, either $f'_{sT}(\Delta) \geq f'_{s0}(\Delta)$ or $f'_{sT}(\Delta) \leq f'_{s0}(\Delta)$ on $(-\infty, \Delta_0]$. If $f'_{sT}(\Delta) \leq f'_{s0}(\Delta)$ on $(-\infty, \Delta_0]$, then

$$\begin{aligned} y'_T(\Delta) \leq y'_0(\Delta) &\Rightarrow \int_T^{\Delta_0} y'_T(s)ds \leq \int_T^{\Delta_0} y'_0(s)ds \\ &\Rightarrow y_T(\Delta_0) - y_T(T) \leq y_0(\Delta_0) - y_0(T) \\ &\Rightarrow y_T(T) \geq y_0(T) > 0 \end{aligned}$$

which contradicts the fact that $y_T(T) = 0$. Therefore, $y'_T(\Delta) \geq y'_0(\Delta)$ for $\Delta \leq \Delta_0$. As a result, we have $y'_T(0) > 0$ and $y'_T(-\Delta_0) > 0$, which will be used in the next step. Next, as illustrated in Fig. 19(d), we want to show there is one and only one root of $y'_T(\Delta)$, denoted by r, on the interval $(0, \Delta_0)$. This can be shown by proving the following statements.

1) There is at least one root in $[T, \Delta_0)$. Since $y_T(T) = y_T(\Delta_0) = 0$, by applying the mean value theorem, there must be at least one root of $y'_T(\Delta)$.
2) The roots of $y'_T(\Delta) = 0$ in the interval $[T, \Delta_0]$ can only lie in $(0, \Delta_0)$. If $T > 0$, this is obvious. Otherwise $T \in (-\Delta_0, 0]$, since we showed earlier $y'_T(-\Delta_0) > 0$ and $y'_T(0) > 0$, any root in $[-\Delta_0, 0]$ will contradict the fact that $y'_T(\Delta)$ is concave on $(-\infty, 0]$.
3) There is only one root in $(0, \Delta_0)$. If there is more than one root in $(0, \Delta_0)$, it will contradict with the convexity of the function $y'_T(\Delta)$ in $[0, \infty)$.

In the last step, using the convexity and concavity of the function $y'_T(\Delta)$ and the fact that $y'_T(r) = 0$ where $r \in (0, \Delta_0)$, we can check that the inequality $y_T(\Delta) \geq 0$ holds on the following intervals:

1) $\Delta \in [T, r), y'_T(\Delta) \geq 0$: $y_T(\Delta) = y_T(T) + \int_T^{\Delta} y'_T(s)ds \geq 0$.
2) $\Delta \in [r, \Delta_0)$, $y'_T(\Delta) \leq 0$: $y_T(\Delta) = y_T(\Delta_0) + \int_{\Delta_0}^{\Delta} y_T(s)ds = -\int_{\Delta}^{\Delta_0} y'_T(s)ds \geq 0$.
3) $\Delta \in [\Delta_0, \infty)$, $y'_T(\Delta) \geq 0$: $y_T(\Delta) = y_T(\Delta_0) + \int_{\Delta_0}^{\Delta} y'_T(s)ds \geq 0$.

Above all, we have proved the second case.

The third case $\Delta_0 \leq 0$ is symmetric to the second case and can, therefore, be proved in the same way. Therefore, we have shown that inequality (27) holds in all three cases.

Next, we would like to show that a is the smallest number that satisfies conditions (26) and (27). Let us assume there exists $\tilde{a} < a$, $\tilde{b}$ and $\tilde{c}$ such that the substitute function denoted as $\widetilde{f}_s(\Delta)$ satisfies conditions (26) and (27). Therefore, $\widetilde{f}_s(\Delta)$ must also satisfy (28). Thus, we have

$$\begin{aligned} \widetilde{f}_s(\Delta_0) &= f_s(\Delta_0) \\ \widetilde{f}'_s(\Delta_0) &= f'_s(\Delta_0). \end{aligned}$$

Therefore, by Taylor series expansion at $\Delta = \Delta_0$, we find $f_s(\Delta) - \widetilde{f}_s(\Delta) = (a - \tilde{a})(\Delta - \Delta_0)^2$. Consequently, $f_s(\Delta) > \widetilde{f}_s(\Delta)$ for $\Delta \neq \Delta_0$. In particular, we choose $\Delta = T$, where T is given by (21). Notice that, if $\Delta_0 \neq 0$, then $T \neq \Delta_0$, thus, $\widetilde{f}_s(T) < f_s(T) = \rho(T)$. Therefore, $\widetilde{f}(\Delta)$ does not satisfy condition (27). If $\Delta_0 = 0$, let $\widetilde{y}_0(\Delta) = \widetilde{f}_s(\Delta) - \rho(\Delta)$, then $\widetilde{y}''_0(0) < y''_0(0) = 0$. Therefore, there exist $\tau > 0$ such that $\forall \Delta \in (0, \tau)$, $\widetilde{y}'(\Delta) < \widetilde{y}'(0) = 0$. Then $\widetilde{y}_0(\tau) = \int_0^{\tau} \widetilde{y}'_0(s)ds < 0$, that is, $\widetilde{f}_s(\tau) < \rho(\tau)$, which violates the condition (27). ■

REFERENCES

[1] G. Wang, H. Yu, and B. De Man, "An outlook on x-ray CT research and development," *Med. Phys.*, vol. 35, no. 3, pp. 1051–1064, 2008.

[2] J.-B. Thibault, K. Sauer, C. Bouman, and J. Hsieh, "A three-dimensional statistical approach to improved image quality for multi-slice helical CT," *Med. Phys.*, vol. 34, no. 11, pp. 4526–4544, 2007.

[3] J.-B. Thibault, K. Sauer, C. Bouman, and J. Hsieh, "Three-dimensional statistical modeling for image quality improvements in multi-slice helical CT," in *Proc. Int. Conf. Fully 3-D Reconstruct. Radiol. Nucl. Med.*, Salt Lake City, UT, Jul. 6–9, 2005, pp. 271–274.

[4] D. Pollitte, S. Yan, J. O'Sullivan, D. Snyder, and B. Whiting, "Implementation of alternating minimization algorithms for fully 3-D CT imaging," in *Proc. SPIE/IS&T Symp. Computat. Imag. II*, San Jose, CA, Jan. 17–18, 2005, vol. 5674, no. 49, pp. 362–373.

[5] I. Elbakri and J. Fessler, "Statistical image reconstruction for polyenergetic x-ray computed tomography," *IEEE Trans. Med. Imag.*, vol. 21, no. 2, pp. 89–99, Feb. 2002.

[6] M. Iatrou, B. De Man, and S. Basu, "A comparison between filtered backprojection, post-smoothed weighted least squares, and penalized weighted least squares for CT reconstruction," in *Proc. IEEE Nucl. Sci. Symp. Conf.*, Nov. 1, 2006, vol. 5, pp. 2845–2850.

[7] A. Ziegler, T. Köhler, and R. Proksa, "Noise and resolution in images reconstructed with FBP and OSC algorithms for CT," *Med. Phys.*, vol. 34, no. 2, pp. 585–598, 2007.

[8] S. Do, M. K. Kalra, Z. Liang, W. C. Karl, T. J. Brady, and H. Pien, "Noise properties of iterative reconstruction techniques in low-dose CT scans," in *Proc. Med. Imag. 2009: Phys. Med. Imag.*, 2009, vol. 7258, no. 1, SPIE.

[9] B. De Man and S. Basu, "Distance-driven projection and backprojection in three-dimensions," *Phys. Med. Biol.*, vol. 49, pp. 2463–2475, 2004.

[10] R. Lewitt, "Multidimensional digital image representations using generalized kaiser-bessel window functions," *J. Opt. Soc. Amer. A*, vol. 7, no. 10, pp. 1834–1846, Oct. 1990.

[11] B. Whiting and P. Massoumzadeh, "Properties of preprocessed sinogram data in x-ray computed tomography," *Med. Phys*, vol. 33, pp. 3290–3303, 2006.

[12] J.-B. Thibault, C. Bouman, K. Sauer, and J. Hsieh, "A recursive filter for noise reduction in statistical tomographic imaging," in *Proc. SPIE/IS&T Symp. Comput. Imag. IV*, San Jose, CA, Jan. 16–18, 2006, vol. 6065, p. 0X.

[13] D. Geman and G. Reynolds, "Constrained restoration and the recovery of discontinuities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 3, pp. 367–383, Mar. 1992.

[14] C. Bouman and K. Sauer, "A generalized Gaussian image model for edge-preserving MAP estimation," *IEEE Trans. Image Process.*, vol. 2, no. 3, pp. 296–310, Jul. 1993.

[15] V. Panin, G. Zeng, and G. Gullberg, "Total variation regulated EM algorithm," *IEEE Trans. Nucl. Sci.*, vol. 46, no. 6, pp. 2202–2210, Dec. 1999.

[16] K. Mueller and F. Xu, "Practical considerations for GPU-accelerated CT," in *PROC. 3rd IEEE Int. Symp. Biomed. Imag.: Nano to Macro*, Apr. 2006, pp. 1184–1187.

[17] M. Kachelrieß, M. Knaup, and O. Bockenbach, "Hyperfast parallel-beam and cone-beam backprojection using the cell general purpose hardware," *Med. Phys.*, vol. 34, no. 4, pp. 1474–1486, 2007.

[18] L. Shepp and Y. Vardi, "Maximum likelihood reconstruction for emission tomography," *IEEE Trans. Med. Imag.*, vol. MI-1, no. 2, pp. 113–122, Oct. 1982.

[19] E. Ü. Mumcuoğlu, R. Leahy, S. Cherry, and Z. Zhou, "Fast gradient-based methods for Bayesian reconstruction of transmission and emission pet images," *IEEE Trans. Med. Imag.*, vol. 13, no. 4, pp. 687–701, Dec. 1994.

[20] H. Hudson and R. Larkin, "Accelerated image reconstruction using ordered subsets of projection data," *IEEE Trans. Med. Imag.*, vol. 13, no. 4, pp. 601–609, Dec. 1994.

[21] C. Bouman and K. Sauer, "A unified approach to statistical tomography using coordinate descent optimization," *IEEE Trans. Image Process.*, vol. 5, no. 3, pp. 480–492, Mar. 1996.

[22] S.-J. Lee, "Accelerated coordinate descent methods for Bayesian reconstruction using ordered subsets of projection data," in *Proc. SPIE Conf. Math. Model., Estimation, Imag.*, Oct. 2000, vol. 4121, pp. 170–181.

[23] H. Zhu, H. Shu, J. Zhou, and L. Luo, "A weighted least squares PET image reconstruction method using iterative coordinate descent algorithms," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf.*, Oct. 2004, vol. 6, pp. 3380–3384.
[24] J. Zheng, S. Saquib, K. Sauer, and C. Bouman, "Parallelizable Bayesian tomography algorithms with rapid, guaranteed convergence," *IEEE Trans. Image Process.*, vol. 9, no. 10, pp. 1745–1759, Oct. 2000.
[25] J. Fessler, E. Ficaro, N. Clinthorne, and K. Lange, "Grouped-coordinate ascent algorithms for penalized-likelihood transmission image reconstruction," *IEEE Trans. Med. Imag.*, vol. 16, no. 2, pp. 166–175, Apr. 1997.
[26] T. Abatzoglou and B. O'Donnell, "Minimization by coordinate descent," *J. of Opt. Theory Appl.*, vol. 36, no. 2, pp. 163–174, Feb. 1982.
[27] Z. Q. Luo and P. Tseng, "On the convergence of the coordinate descent method for convex differentiable minimization," *J. Opt. Theory Appl.*, vol. 72, no. 1, pp. 7–35, Jan. 1992.
[28] B. DeMan, S. Basu, J.-B. Thibault, J. Hsieh, J. Fessler, K. Sauer, and C. Bouman, "A study of different minimization approaches for iterative reconstruction in x-ray CT," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf.*, San Juan, Puerto Rico, Oct. 23–29, 2005, vol. 5, pp. 2708–2710.
[29] Z. Yu, J.-B. Thibault, C. A. Bouman, K. D. Sauer, and J. Hsieh, "Nonhomogeneous updates for the iterative coordinate descent algorithm," in *Proc. SPIE/IS&T Symp. Comput. Imag. V*, San Jose, CA, 2007, vol. 6498, p. 1B.
[30] Z. Yu, J.-B. Thibault, K. D. Sauer, C. A. Bouman, and J. Hsieh, "Accelerated line search for coordinate descent optimization," in *Proc. IEEE Nucl. Sci. Symp.Med. Imag. Conf.*, San Diego, CA, 2006, , vol. 5, pp. 2841–2844.
[31] J. Fessler and A. Hero, "Penalized maximum-likelihood image reconstruction using space-alternating generalized EM algorithms," *IEEE Trans. Image Process.*, vol. 4, no. 10, pp. 1417–1429, Oct. 1995.
[32] H. Erdogan and J. Fessler, "Monotonic algorithms for transmission tomography," *IEEE Trans. Med. Imag.*, vol. 18, no. 9, pp. 801–814, Sep. 1999.
[33] K. Sauer and C. Bouman, "A local update strategy for iterative reconstruction from projections," *IEEE Trans. Signal Process.*, vol. 41, no. 2, Feb. 1993.
[34] J. Bowsher, M. Smith, J. Peter, and R. Jaszczak, "A comparison of OSEM and ICD for iterative reconstruction of SPECT brain images," *J. Nucl. Med.*, vol. 39, p. 79P, 1998.
[35] J.-B. Thibault, K. Sauer, and C. Bouman, "Newton-style optimization for emission tomographic estimation," *J. Electron. Imag.*, vol. 9, no. 3, pp. 269–282, 2000.

Zhou Yu (S'05–M'10) received the B.S. degree in electronics engineering from Shanghai Jiao Tong University, Shanghai, China, in 2005, and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, in 2009.

He is currently a Scientist in the CT Systems Engineering Group, GE Healthcare, Waukesha, WI. His research interests are primarily in the areas of statistical signal processing and inverse problems, with particular focus on tomographic reconstruction and numerical algorithms.

Dr. Yu is a member of the Eta Kappa Nu Honorary Society.

Jean-Baptiste Thibault (M'09) received the Diplme d'Ingnieur degree from the Ecole Suprieure d'Electricit, Gif-sur-Yvette, France, in 1998, and the M.S. and Ph.D. degrees in electrical engineering from the University of Notre Dame, South Bend, IN, in 1999 and 2006, respectively.

He is currently Senior Scientist at GE Healthcare, Waukesha, WI, where he has been awarded more than 20 patents and patent applications covering multiple aspects of the design and optimization of CT scanners. His research interests include x-ray dose management, tomographic reconstruction, and numerical optimization methods, with an emphasis on Bayesian estimation and its applications to statistical reconstruction of medical images.

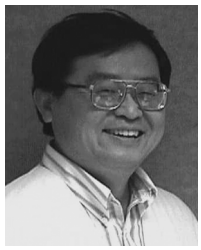
Charles A. Bouman (S'86–M'89–SM'97–F'01) received the B.S.E.E. degree from the University of Pennsylvania, Philadelphia, in 1981, the M.S. degree from the University of California, Berkeley, in 1982, and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, in 1989.

From 1982 to 1985, he was a full staff member at the Massachusetts Institute of Technology Lincoln Laboratory. In 1989, he joined the faculty of Purdue University, West Lafayette, IN, where he is a Professor with a primary appointment in the School of Electrical and Computer Engineering and a secondary appointment in the School of Biomedical Engineering. Currently, he is Co-Director of Purdue's Magnetic Resonance Imaging Facility located in Purdue's Research Park. His research focuses on the use of statistical image models, multiscale techniques, and fast algorithms in applications including tomographic reconstruction, medical imaging, and document rendering and acquisition.

Prof. Bouman is a Fellow of the IEEE, a Fellow of the American Institute for Medical and Biological Engineering (AIMBE), a Fellow of the society for Imaging Science and Technology (IS&T), a Fellow of the SPIE professional society, a recipient of IS&T's Raymond C. Bowman Award for outstanding contributions to digital imaging education and research, and a University Faculty Scholar of Purdue University. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON IMAGE PROCESSING and a member of the IEEE Biomedical Image and Signal Processing Technical Committee. He has been a member of the Steering Committee for the IEEE TRANSACTIONS ON MEDICAL IMAGING and an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING and the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. He has also been Co-Chair of the 2006 SPIE/IS&T Symposium on Electronic Imaging, Co-Chair of the SPIE/IS&T Conferences on Visual Communications and Image Processing 2000 (VCIP), a Vice President of Publications and a member of the Board of Directors for the IS&T Society, and he is the founder and Co-Chair of the SPIE/IS&T Conference on Computational Imaging.

Ken D. Sauer (S'85–M'89) received the B.S.E.E. and M.S.E.E. degrees from Purdue University, West Lafayette, IN, in 1984 and 1985, respectively, and the Ph.D. degree in electrical engineering as an AT&T Foundation Fellow from Princeton University, Princeton, NJ, in 1989.

Since 1989, he has been with the University of Notre Dame, South Bend, IN, where he currently holds the position of Associate Professor and Director of Undergraduate Studies in the Department of Electrical Engineering. His research interests are primarily within the domain of inverse problems, with particular concentration in tomography for medical diagnostic imaging and nondestructive evaluation. His work includes stochastic image modeling for Bayesian estimation, numerical methods for optimization, and general nonlinear estimation techniques. His research has been funded by GE Health Care, NASA, the National Science Foundation, Electricite de France and the State of Indiana's 21st Century Fund.



Jiang Hsieh (S'82–M'84–SM'06) received the Ph.D. degree from Illinois Institute of Technology, Chicago, IL, in 1989.

He is currently a Chief Scientist in the CT Engineering Division, GE Healthcare, Waukesha, WI, and an Adjunct Professor in the Medical Physics Department, University of Wisconsin, Madison, WI. He has over 26 years of experience in medical imaging. His research interests include tomographic reconstruction, CT image artifact reduction, signal processing, image processing, and advanced CT applications.