https://doi.org/10.2352/ISSN.2470-1173.2018.15.COIMG-102

Distributed Framework for Fast Iterative CT Reconstruction from View-subsets

Venkatesh Sridhar [1], Gregery T. Buzzard [2] and Charles A. Bouman [1];
[1] School of Electrical & Computer Engineering, Purdue University, West Lafayette, IN, USA
[2] Dept. of Mathematics, Purdue University, West Lafayette, IN, USA

Abstract

Model-Based Image Reconstruction (MBIR) methods significantly enhance the quality of tomographic reconstruction in contrast to analytical techniques. However, the intensive computational time and memory required by MBIR limit its use for many practical real-time applications. But, with increasing availability of parallel computing resources, distributed MBIR algorithms can overcome this limitation on computational performance.

In this paper, we propose a novel distributed and iterative approach to Computed Tomography (CT) reconstruction based on the Multi-Agent Consensus Equilibrium (MACE) framework. We formulate CT reconstruction as a consensus optimization problem wherein the objective function, and consequently the system matrix, is split across multiple disjoint view-subsets. This produces multiple regularized sparse-view reconstruction problems that are tied together by a consensus constraint, and these problems can be solved in parallel within the MACE framework. Further, we solve each sub-problem inexactly, using only 1 full pass of the Iterative Coordinate Descent (ICD) optimization technique. Yet, our distributed approach is convergent. Finally, we validate our approach with experiments on real 2D CT data.

Introduction

Tomographic reconstruction methods can be broadly divided into two categories ; Analytical reconstruction, and, Model-based iterative reconstruction (MBIR). In comparison with analytical methods, MBIR has the advantage that the quality of reconstruction is superior even when projection data is sparse. Therefore, in X-ray imaging systems where views are very limited, or equivalently, scanning time is significantly reduced, MBIR offers a way to achieve high quality reconstruction. Similarly, in medical imaging systems where X-ray dosage is highly restricted to ensure safety of the patient, MBIR vastly outperforms analytical methods in reconstruction quality [8]. However, despite all these advantages, MBIR suffers from the downside that it is computationally expensive and much slower than analytical methods. Further, in certain reconstruction applications, it is computationally more efficient to pre-compute and store the system matrix [2]. But such an approach is often limited by available memory in the case of large-scale problems. Nevertheless, distributed MBIR algorithms can overcome the above limitations on computational time and memory. Furthermore, both the increasing availability and reducing cost of parallel computing resources has made the distributed implementation of MBIR even more viable for real-time applications.

Wang et al. in [2] gives a brief overview on the scalability of current MBIR algorithms on distributed memory systems. Most MBIR algorithms are massively scalable in the spatial domain, that is, multiple voxels in object space can be updated in parallel. While simultaneous update methods based on gradient descent optimization and its accelerated variants inherently allow such scalability, it has been recently shown that even greedy sequential update methods such as Iterative Coordinate descent (ICD) optimization can be tailored to achieve similar scalability [2]. Furthermore, greedy methods such as ICD have the advantage of very fast convergence compared to simultaneous methods. While ICD can typically achieve convergence in less than ten iterations, gradient-descent methods usually require hundreds of iterations in the absence of preconditioning [12].

Another way to scale MBIR across a large number of parallel nodes is by distributing its computation in the sinogram-view domain. More specifically, we consider the problem where the entire set of sinogram-views for a given object is partitioned among the individual nodes of a distributed memory system. However, in such a scenario where each node owns only a sparse subset of the global sinogram-views, fast MBIR algorithms that rely on ICD do not directly scale across nodes.

In this paper, we propose a distributed approach called *view-subset parallelization* that solves the above problem. The key idea of this approach is to formulate reconstruction as a *consensus optimization* problem. In such a formulation, every node renders a limited-view reconstruction from its own sparse subset of sinogram data, and through repeated centralized communication with other nodes, drives its individual reconstruction towards the true global solution.

Our above-proposed distributed approach provides a number of computational advantages. First, since each node only processes a subset of views, it must store only a small portion of the system matrix, which can dramatically reduce memory requirements. Second, by distributing the computation across many nodes, the reconstruction time can be reduced. In certain applications such as medical imaging via helical-scan CT, it is preferable to compute the forward projection operator on-the-fly due to adaptive geometrical parameters of the system, but this increases computational overhead. However, many scientific applications such as X-ray synchrotron imaging permit precomputing and storing the entire forward projection matrix. Therefore, especially for such applications, our distributed approach both significantly reduces the memory foot-print of the projection matrix on each node and also provides faster data access. Further, our approach can be an auxiliary method to push the existing limits of scalability for massively parallel imaging techniques such as [2] that exploit parallelism in the spatial domain alone.

View-subset Parallelization Approach

Figure 1 illustrates how *view-subsets* are generated. All sinogram-views of the same object slice are divided among N disjoint subsets in an interleaved manner. In this specific example, the total number of views, N_θ, is 4, and N is 2. In general, when N is sufficiently large, each subset contains a very sparse subset of all views acquired.

From now onward, we shall refer to the reconstruction of the object from all N_θ views as the *true reconstruction*. Figure 2 depicts the intuition behind our approach of view-subset parallelization. Each of the N sparse view-subsets can be used to render a reconstruction of its own, albeit of a relatively lower quality than the true reconstruction. We can choose any suitable sparse-view reconstruction method for the above, including those based on ICD. Intuitively, we wish to perform these N individual sparse-view reconstructions in parallel, and then somehow "fuse" them together to arrive at the true solution, as shown in Figure 2. A naive approach would be to merely average the N individual sparse-view reconstructions as an approximation to the true solution. However, such a simplistic approach would be far from the true reconstruction, especially when N is large. This is primarily because there is no consensus between the individual sparse-view reconstructions, and further, each one of them would be riddled with artifacts, particularly when dealing with real sinogram data. Therefore, a more sophisticated mathematical framework is required to both achieve parallelism across multiple view-subsets and simultaneously guarantee convergence to the true solution.

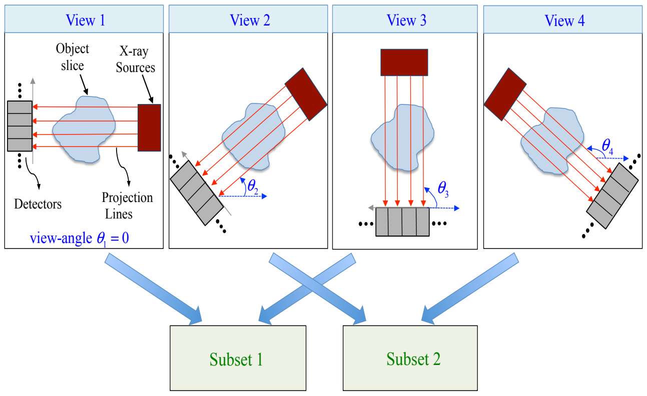


Figure 1. Generation of view-subsets. Sinogram-views are divided among multiple disjoint subsets in a round-robin fashion. In this example, 4 views of an object are split among 2 subsets.

Equation (1) describes the broad idea behind the mathematical framework needed for our distributed approach. In this equation, $x \in \mathbb{R}^n$ represents the image to be reconstructed, and $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is the convex objective function for reconstructing the image from all N_θ views acquired.

$$\text{Minimize} \;\; f(x) = f_1(x) + f_2(x) + \ldots + f_N(x), \;\; x \in \mathbb{R}^n. \quad (1)$$

In the subsequent sections of this paper, we will show that f can be expressed as a sum of N functions, where N represents the number of view-subsets, and $f_i : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, $i = 1,\ldots,N$, represents the convex objective function for reconstructing the image from sparse views of the i-th view-subset alone. This optimization problem of equation (1) can be rewritten as a *consensus optimization* problem as shown below.

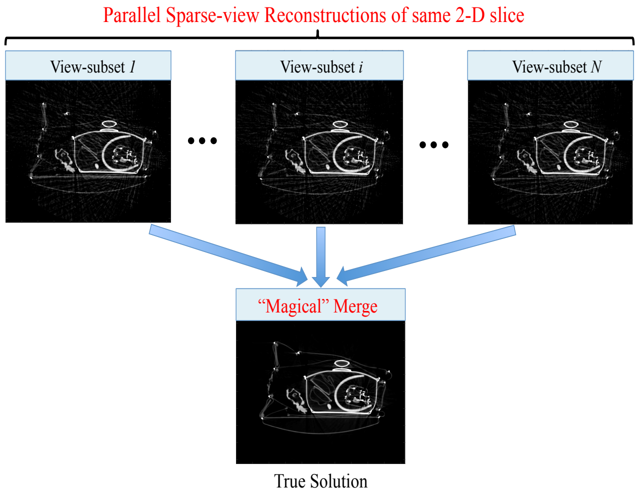


Figure 2. Intuition behind view-subset parallelization. Individual sparse-view reconstructions based on any technique are rendered in parallel from each of the N view-subsets, and then somehow combined together to obtain the true solution.

$$\text{Minimize} \sum_{i=1}^{N} f_i(x_i), \; x_i \in \mathbb{R}^n, \; \text{subject to } x_i = z, \; i = 1,\ldots,N. \quad (2)$$

The constraint $x_i = z$, $i = 1,\ldots,N$, enforces consensus among individual sparse-view reconstructions, $x_i \in \mathbb{R}^n$. The *Multi-Agent Consensus Equilibrium* (MACE) approach proposed by Buzzard et al. [1] provides a general framework wherein constrained optimization problems such as (2) and more general consensus problems can be solved in an iterative and parallel fashion. In subsequent sections of this paper, we shall discuss the MACE framework in further detail, and also propose certain modifications that allow it to be computationally efficient for our application of tomographic reconstruction.

Partitioning the Objective function among View-subsets

We express the true reconstruction, $x^* \in \mathbb{R}^n$, as the maximum-a-posteriori (MAP) estimate given by

$$x^* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} f(x), \;\; \text{where}$$
$$f(x) = -\log p(y|x) - \log p(x), \quad (3)$$

and y is the list of standard projection measurements from all N_θ views. In the above equation, $p(y|x)$ represents the likelihood model of the measurements y conditioned on the unknown image x, and $p(x)$ represents the prior model of the unknown image x. Let $k \in \{1,\ldots,N_\theta\}$ denote view-index and N_d denote number of detectors. Sauer and Bouman [3] approximate the negative log likelihood model as

$$-\log p(y|x) = \sum_{k=1}^{N_\theta} \frac{1}{2}(y_k - A_k x)^t \Lambda_k (y_k - A_k x) \tag{4}$$

where $A_k \in \mathbb{R}^{N_d \times n}$ is the forward projection matrix for the k-th view, $\Lambda_k \in \mathbb{R}^{N_d \times N_d}$ is the diagonal weighting matrix and $y_k \in \mathbb{R}^{N_d}$ is the list of projection measurements for the k-th view. The entry of the matrix A_k at index (l, j), $A_{k,l,j}$, represents the contribution of the j-th pixel to the projection measurement of the l-th detector, when the source-detector arrangement is rotated by the k-th view angle. From equations (4) and (3), the objective function for MAP estimation is

$$f(x) = \sum_{k=1}^{N_v} \frac{1}{2}\|y_k - A_k x\|^2_{\Lambda_k} + \beta h(x) \tag{5}$$

where $h(x)$ is the negative log prior model, and $\beta > 0$ is the overall regularization parameter. Typically h is chosen to be a convex spatial smoothing function that penalizes differences between neighboring pixels. This convexity of h makes f convex as well.

We can split f among N view-subsets and express it in a form similar to equation (1). We define $J_i,\ i = 1,...,N$ to be a subset of view-indices such that $J_1 \cup J_2 \cdots \cup J_N = \{1,...,N_\theta\}$ and $J_1 \cap J_2 \cdots \cap J_N = \emptyset$. We define function f_i as

$$f_i(x) = \sum_{k \in J_i} \frac{1}{2}\|y_k - A_k x\|^2_{\Lambda_k} + \frac{\beta}{N} h(x)\ . \tag{6}$$

Note that f_i represents the objective function for regularized sparse-view reconstruction, since it includes both the data fidelity fit for views indexed in subset J_i, as well as the prior model $h(x)$. From the above definition it follows that

$$f(x) = \sum_{i=1}^{N} f_i(x)\ .$$

Therefore, our reconstruction problem can reformulated as a consensus optimization problem of the form given by equation (2). We will use the MACE framework to solve this problem in a distributed and iterative fashion.

Multi-Agent Consensus Equilibrium (MACE) Framework

Buzzard et al. in [1] propose a general framework named MACE to solve consensus optimization problems such as (2). More specifically, it prescribes a system of equations which determine the consensus solution. Further, this specific system of equations that can be solved by various methods, and some of these methods are parallelizable. One particular approach in [1] to solve the MACE system of equations is closely related to consensus ADMM [11].

Before we move onto further details regarding the MACE framework, our notations and definitions for the subsequent sections of this paper are as follows:

- We define $F_i : \mathbb{R}^n \to \mathbb{R}^n$, the proximal map of a convex function $f_i : \mathbb{R}^n \to \mathbb{R}$ as

$$F_i(x) = \underset{v \in \mathbb{R}^n}{\operatorname{argmin}} \left\{ f_i(v) + \frac{1}{2\sigma^2}\|v - x\|^2 \right\}. \tag{7}$$

- We use bold notation to denote a tall vector consisting of N individual vectors, each belonging to $\mathbb{R}^n$. For example, $\mathbf{v} \in \mathbb{R}^{nN}$ denotes a tall vector, where $\mathbf{v} = [v_1^t\ v_2^t\ \cdots\ v_N^t]^t$ and $v_i \in \mathbb{R}^n,\ i = 1,...,N$.
- We use the bold notation with a "bar" superscript to denote the averaging operation on a tall vector belonging to $\mathbb{R}^{nN}$. For example, we define $\bar{\mathbf{v}} = (1/N)\sum_{i=1}^{N} v_i$.
- We use the non-bold notation with a "hat" superscript to denote a vertical stacking operation on a vector belonging to $\mathbb{R}^n$. For example, we define $\hat{x} = [x^t\ x^t\ \cdots\ x^t]^t$, where $x \in \mathbb{R}^n$ and $\hat{x} \in \mathbb{R}^{nN}$.

The MACE framework states that determining x^*, solution to optimization problem of equation (2), is equivalent to finding a pair $(x^*, \mathbf{u}^*) \in \mathbb{R}^n \times \mathbb{R}^{nN}$ that satisfies the conditions

$$F_i(x^* + u_i^*) = x^*,\ i = 1, \ldots, N, \tag{8}$$

$$\bar{\mathbf{u}}^* = 0. \tag{9}$$

The above conditions can be rewritten more compactly as

$$\mathbf{F}(\mathbf{v}^*) = \mathbf{G}(\mathbf{v}^*), \tag{10}$$

where $\mathbf{v}^* = \hat{x}^* + \mathbf{u}^*$, and maps $\mathbf{F}, \mathbf{G} : \mathbb{R}^{nN} \to \mathbb{R}^{nN}$ are defined as

$$\mathbf{F}(\mathbf{v}) = \begin{pmatrix} F_1(v_1) \\ \vdots \\ F_N(v_N) \end{pmatrix} \text{ and } \mathbf{G}(\mathbf{v}) = \begin{pmatrix} \bar{\mathbf{v}} \\ \vdots \\ \bar{\mathbf{v}} \end{pmatrix}, \tag{11}$$

for any $\mathbf{v} \in \mathbb{R}^{nN}$. Importantly, equation (10) specifies the system of equations for the MACE framework, whose solution $\mathbf{v}^*$ can give us the consensus solution x^*, since $\bar{\mathbf{v}}^* = x^*$. From equation (11), note that the N individual proximal maps within $\mathbf{F}$ can be evaluated in parallel, and so we interpret $\mathbf{F}$ as a parallel proximal map operator. On the other hand, $\mathbf{G}$ can be interpreted as a centralized merge & broadcast operator that inputs $v_i \in \mathbb{R}^n,\ i = 1,...,N$ from N different nodes and then broadcasts back their average.

Buzzard et al. in [1] suggest various methods for solving the MACE system of equations specified by equation (10). One particular method from [1] that we will use in this paper, is reformulating equation (10) as determining the fixed-point of a specific map $\mathbf{T} : \mathbb{R}^{nN} \to \mathbb{R}^{nN}$. More specifically, if we define $\mathbf{w}^*$ as $\mathbf{w}^* = \hat{x}^* - \mathbf{u}^*$, or equivalently, $\mathbf{w}^* = (2\mathbf{G} - \mathbf{I})\mathbf{v}^*$ since $\bar{\mathbf{u}}^* = 0$, then $\mathbf{w}^*$ is the fixed-point of the map $\mathbf{T} \triangleq (2\mathbf{F} - \mathbf{I})(2\mathbf{G} - \mathbf{I})$. Once we determine $\mathbf{w}^*$, we can subsequently compute x^*, since $\bar{\mathbf{w}}^* = x^*$.

It can be shown that the aforestated map $\mathbf{T}$ is non-expansive, and so one particular way to determine its fixed-point $\mathbf{w}^*$ is to use the *Mann iteration* technique. In this technique, we begin with any $\mathbf{w}^{(0)} \in \mathbb{R}^{nN}$, an initial guess of the fixed-point $\mathbf{w}^*$, and then each iteration is a weighted averaging of the form

$$\mathbf{w}^{(k+1)} = \rho \mathbf{T}\mathbf{w}^{(k)} + (1 - \rho)\mathbf{w}^{(k)}, \tag{12}$$

where $k \geq 0$ and $\rho \in (0, 1)$ is a weighting parameter that can even be changed every iteration subject to certain conditions . As k increases, $\mathbf{w}^{(k)}$ converges to $\mathbf{w}^*$ irrespective of the intialization $\mathbf{w}^{(0)}$.

Importantly, note that the Mann iteration is parallelizable for our specific $\mathbf{T}$ because $\mathbf{F}$ is a parallel operator, and, even though

$\mathbf{G}$ is a centralized operator, its computational overhead is significantly lower than that of $\mathbf{F}$. Therefore, this fixed-point algorithm provides us a distributed and iterative approach to solving the consensus optimization problem of equation (2). However, in the next section of this paper, we explain why this approach is too computationally expensive and also propose a solution to this problem.

Partial Update MACE (PUMACE) Framework

The MACE approach described in the previous section of this paper is in general computationally expensive and not practical for many applications. This is because each proximal map $F_i = F_i(\cdot\,;\sigma) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ defined by equation (7) requires iterative optimization on its own, and so, operator $\mathbf{F}$ has high computational overhead. Consequently, the MACE approach contains a large number of nested iterations and and requires many inner iterations to converge to the consensus solution.

To overcome this issue, we propose a slightly modified version of the MACE approach called Partial Update MACE (PUMACE). The main idea here is to replace each proximal map, $F_i(\cdot\;;\sigma)$, with an approximate version that is quicker to solve. Our approximation is based on using a highly reduced number of iterations to evaluate the proximal map, even though this may not yield the fully converged result for this map. We accordingly name this type of approximation the partial-update (PU) approximation. In theory, since f_i in equation (2) is convex, its proximal map $F_i(\cdot\,;\sigma)$ should ultimately converge to a unique global minimum irrespective of the initial state. However, when the number of iterations is very limited, the choice of initial state determines the solution. So further, the PU approximation is also dependent on the initial state. We use the notation $\tilde{F}_i(\cdot\,;\sigma,X_i) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ to denote the PU approximation of the proximal map $F_i(\cdot\,;\sigma)$, where $X_i \in \mathbb{R}^n$ specifies the initial state.

For any $\mathbf{v},\mathbf{X} \in \mathbb{R}^{nN}$, we define the parallel operator $\tilde{\mathbf{F}} : \mathbb{R}^{nN} \rightarrow \mathbb{R}^{nN}$ as

$$\tilde{\mathbf{F}}(\mathbf{v};\sigma,\mathbf{X}) = \begin{pmatrix} \tilde{F}_1(v_1;\sigma,X_1) \\ \vdots \\ \tilde{F}_N(v_N;\sigma,X_N) \end{pmatrix}.$$

So, $\tilde{\mathbf{F}}(\mathbf{v};\sigma,\mathbf{X})$ gives us an approximate solution, or *partial update solution* to $\mathbf{F}(\mathbf{v};\sigma)$, and this solution is dependent on the initial state, $\mathbf{X}$.

Algorithm 1 PUMACE algorithm

```
1:  Initialize:
2:  $\mathbf{w}^{(0)} \leftarrow$ any value $\in \mathbb{R}^{nN}$
3:  $\mathbf{X}^{(0)} \leftarrow \mathbf{G}(\mathbf{w}^{(0)})$
4:  $k \leftarrow 0$                                                  ▷ Partial update index
5:  while not converged do
6:      $\mathbf{v} \leftarrow (2\mathbf{G}-\mathbf{I})\,\mathbf{w}^{(k)}$
7:      $\mathbf{X}^{(k+1)} \leftarrow \tilde{\mathbf{F}}\left(\mathbf{v};\sigma,\mathbf{X}^{(k)}\right)$           ▷ Partial update solution
8:      $\mathbf{w}^{(k+1)} \leftarrow 2\mathbf{X}^{(k+1)} - \mathbf{v}$                   ▷ $\approx (2\mathbf{F}-\mathbf{I})(2\mathbf{G}-\mathbf{I})\mathbf{w}^{(k)}$
9:      $\mathbf{w}^{(k+1)} \leftarrow \rho\mathbf{w}^{(k+1)} + (1-\rho)\,\mathbf{w}^{(k)}$          ▷ Mann iteration
10:     $k \leftarrow k+1$
11: end while
12: Solution:
13: $x^* \leftarrow \bar{\mathbf{w}}^{(k)}$                                  ▷ Consenus solution
```

Detailed pseudo-code for the PUMACE algorithm is shown in Algorithm 1. Here, k represents the discrete time-step, alternately referred to as partial update (PU) index. Note that the PUMACE framework allows any choice of optimization technique to compute the PU solution. Importantly, in this paper, we use only 1 pass of the Iterative Coordinate Descent (ICD) optimization method [5] to compute the partial update solution. In this specific case, unlike the MACE approach, the PUMACE approach does not contain nested iterations of optimization.

Importantly, note that in line 7 of Algorithm 1, the result of the current partial update is used as the initial state for the next partial update. This is significant, because a good initial state drives the partial update solution, $\tilde{\mathbf{F}}(\mathbf{v};\sigma,\mathbf{X}^{(k)})$, towards the fully converged solution, $\mathbf{F}(\mathbf{v};\sigma)$. In practice, this occurs predominantly as k increases. Also, it is worth noting that the equilibrium conditions specified by equation (8) or (10) require $\hat{x}^* = \mathbf{F}(\mathbf{v}^*;\sigma)$, or equivalently, $\hat{x}^* = \tilde{\mathbf{F}}(\mathbf{v}^*;\sigma,\hat{x}^*)$ for the partial update case, where x^* is the consensus solution, and $\mathbf{v}^* = \hat{x}^* + \mathbf{u}^*$. Since $\mathbf{v}^* = (2\mathbf{G}-\mathbf{I})\mathbf{w}^*$, [1] the above condition can be more succinctly summarized as

$$\hat{x}^* = \tilde{\mathbf{F}}((2\mathbf{G}-\mathbf{I})\mathbf{w}^*;\sigma,\hat{x}^*).$$

This intuitively justifies the use of $\mathbf{X}^{(k)}$ as the initial state for the $(k+1)$-th partial update in line 7 of the pseudo-code. Further, note that in our specific application of PUMACE to CT reconstruction, such a way of picking the initial state for each partial update is key to fast implementation of operator $\tilde{\mathbf{F}}$. This is because ICD optimization [5] requires maintaining the residual sinogram as a state vector, and our approach allows us to do so without needing to compute a forward projection prior to each partial update.

In Algorithm 1, lines 8 and 9 together are analogous to iteratively evaluating the fixed point of the non-expansive map $\mathbf{T} : \mathbb{R}^{nN} \rightarrow \mathbb{R}^{nN}$, where $\mathbf{T} = (2\mathbf{F}-\mathbf{I})(2\mathbf{G}-\mathbf{I})$. However, since we use $\tilde{\mathbf{F}}$ in place of $\mathbf{F}$ in line 7, we may not obtain the exact fixed-point of the non-expansive map $\mathbf{T}$. Nonetheless, we can prove that for strictly, convex quadratic problems PUMACE converges to the true consensus solution. The proof for the above is beyond the scope of this brief paper and we shall provide it in a future publication. In the experiments section of this paper, we apply the PUMACE approach to CT reconstruction, wherein the objective function is convex but not strictly quadratic as a result of incorporating non-quadratic prior models in equation (5). Even so, we verify that the final PUMACE solution is near the desired fixed-point, and consequently, we approximate the true consensus solution.

Keeping in mind our application of PUMACE approach to CT reconstruction, there are certain key operations that provide us an opportunity to improvise. These key operations are: the merge operation $\mathbf{G}$, the partial sparse-view reconstruction operation $\tilde{\mathbf{F}}$, and the Mann update operation. In order to keep this paper brief, we shall discuss variants of the PUMACE approach based on improvising these key operations in a future publication.

Experimental Results

We present results on two real sinogram datasets acquired by parallel-beam CT modality. The first dataset is from a syn-

[1] Note that from definition of $\mathbf{v}^*$ and $\mathbf{w}^*$ in terms of x^* and $\mathbf{u}^*$, it follows that both $\mathbf{w}^* = (2\mathbf{G}-\mathbf{I})\mathbf{v}^*$ and $\mathbf{v}^* = (2\mathbf{G}-\mathbf{I})\mathbf{w}^*$ hold since $\bar{\mathbf{u}}^* = 0$.

chrotron tomography scan of Iron Hydroxide microstructure, and the second dataset is a security scan of checked-in baggage. The parameters of the datasets are listed in the table below.

Dataset Parameters

Dataset	# Views	# Detectors	Image Size
Iron Hydroxide	225	1024	1024×1024
Baggage Scan	720	1024	512×512

For all experiments in this section, the prior model is a Q-Generalized Gaussian Markov Random Field (Q-GGMRF) prior [9] that can preserve both low contrast and high contrast image features.

We benchmark the convergence of our distributed view-subset approach using a serial baseline approach. This serial baseline approach is conventional ICD reconstruction using the complete set of views. The distributed view-subset approach is based on the PUMACE framework, where each partial update is computed using only 1 pass of ICD optimization.

We measure computational time for reconstruction in units of *equits* rather than machine time. For the case of the serial baseline algorithm, we define 1 equit to be the time taken when all pixels within the region of interest (ROI) are updated in a single pass of ICD [10]. In contrast, our distributed approach utilizes N parallel processes, each owning a sparse subset of views. So more generally, we define an equit as

$$\#\text{equits} = \frac{\sum_{i=1}^{N} \text{Number of pixel updates by } i\text{-th process}}{N \times \text{Number of pixels in ROI}}$$

In practice, for any given partial update, each of the N parallel processes updates roughly the same number of pixels within the ROI. Importantly, note that under this approximation, the above definition of 1 equit does not vary with N. Therefore, our definition is consistent across both serial and distributed approaches. Further, importantly note that in the distributed approach, a single pixel update is roughly N times faster than that of the serial approach that uses all views. Consequently, if both the distributed and serial approaches take the same number of equits to converge, then the distributed approach is faster than the serial approach by a factor of N.

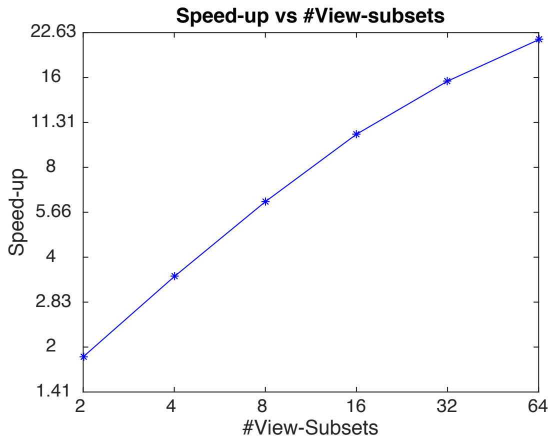
Additionally, we use the procedure of zero-skipping [10] for our ICD updates, and so, in a single pass of ICD optimization, we do not necessarily update all pixels in the ROI. In other words, the number of equits per pass of ICD can be fractional.

To define our metric of normalized RMSE (NRMSE). We define the NRMSE of an image $x \in \mathbb{R}^n$ that is benchmarked against a reference $x^* \in \mathbb{R}^n$ as

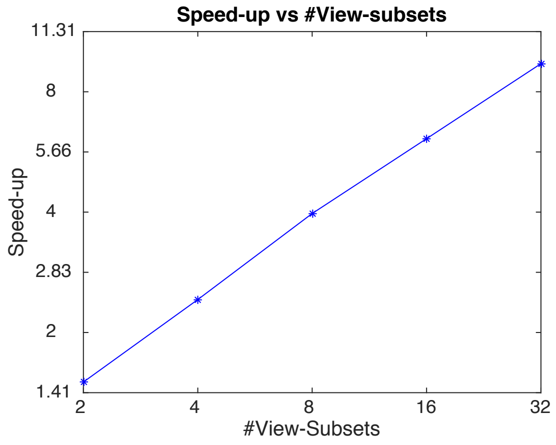
$$\text{NRMSE}(x, x^*) = \frac{\text{RMSE}(x, x^*)}{\text{Mean pixel value of } x^* \text{ within ROI}}.$$

We use the fully converged reconstruction from the baseline approach as reference x^*. We state that our distributed approach achieves convergence, if the merged result $\bar{\mathbf{w}}^{(k)}$ in line (13) of Algorithm 1 satisfies the condition $\text{NRMSE}(\bar{\mathbf{w}}^{(k)}, x^*) < T$, where T is a certain threshold. In our experiments we set $T = 4\%$ for the Iron Hydroxide Dataset, and $T = 5\%$ for the Baggage Scan dataset. We measure speed-up of our distributed approach with N view-subsets, or equivalently N distributed nodes, as

$$\text{Speed-up}(N) = N \times \frac{\#\text{equits for serial approach to converge}}{\#\text{equits for distributed approach to converge}}.$$



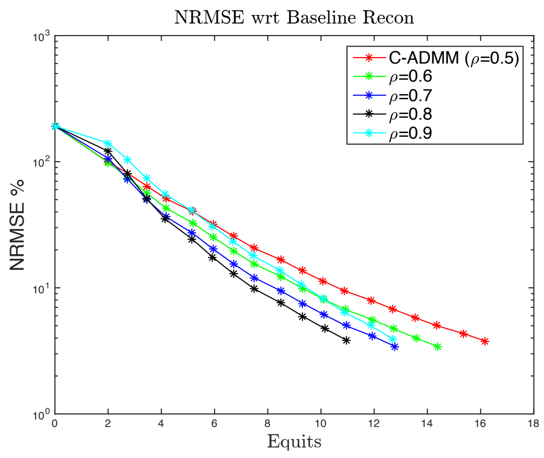
(a) Baggage scan data set



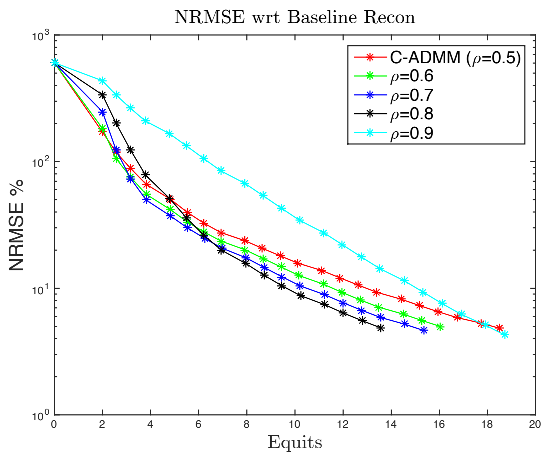
(b) Iron Hydroxide data set

Figure 3. *Scalability of the view-subset parallelization approach using PUMACE vs. the number of view-subsets. Our approach is convergent to the true reconstruction even when #views per subset is very sparse. As # view-subsets increases, speed-up increases, but efficiency decreases.*

Figure 3 illustrates the scalability of our approach. We verify the convergence of our distributed approach with increasing number of view-subsets, N, and further, plot Speed-up(N) vs N as shown in Figure 3. First, note that we achieve convergence even at high values of N, or, equivalently when the views per subset are very sparse. For example, in the case of the Iron Hyroxide dataset, we achieve convergence even when $N = 32$, or equivalently, only 7 views per subset. Similarly, in the case of the Baggage Scan dataset, we achieve convergence even when $N = 64$, or equivalently, only 11 views per subset. Second, note that while increasing N provides us higher speed-ups, the parallel efficiency reduces. For example, in the case of baggage scan dataset, we achieve a speed-up of approximately 21 when $N = 64$, while in the case of the Iron Hydroxide dataset, we achieve a speed-up of approximately 9 when $N = 32$.



(a) Iron Hydroxide data set ($N = 8$)

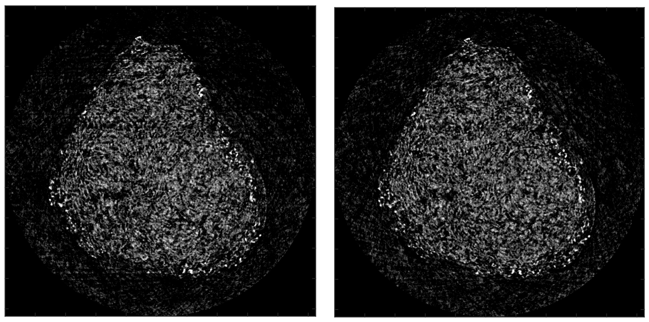


(b) Baggage scan data set ($N = 16$)

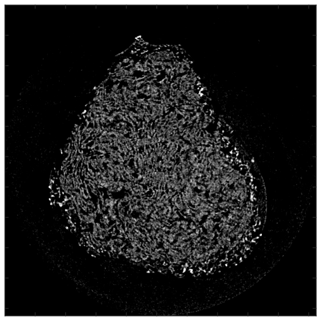
Figure 4. *PUMACE: Effect of Mann iteration parameter ρ on rate of convergence.*

Figure 4 illustrates the effect of Mann update parameter ρ on the rate of convergence of our distributed approach using PUMACE. In sub-figures (a) and (b), we sweep ρ through 5 different values from 0.5 to 0.9. Further, for a given value of ρ, we plot the NRMSE of the iteratively reconstructed image after each partial update (PU), until convergence is achieved. For both datasets, $\rho = 0.8$ seems to be the optimal choice of ρ. This optimal value of ρ produces significantly faster convergence than the case $\rho = 0.5$, which is the default relaxation used by most of the conventional proximal splitting methods including consensus-ADMM [1].

Figures 5 and 6 show reconstructions of the individual view-subsets, w_i^*, $1 \le i \le N$, and the merged consensus result $\bar{\mathbf{w}}^*$ at convergence. Note that while the the individual sparse-view reconstructions have conspicuous artifacts, no such artifacts are present in the consensus result. This is because the artifacts in each w_i^*, represent its disagreement with the consensus solution. Consensus optimization resolves these disagreements, and so, the merged consensus result is the true solution that is devoid of the abovestated artifacts. The above notion is summarized mathematically by equation (9).

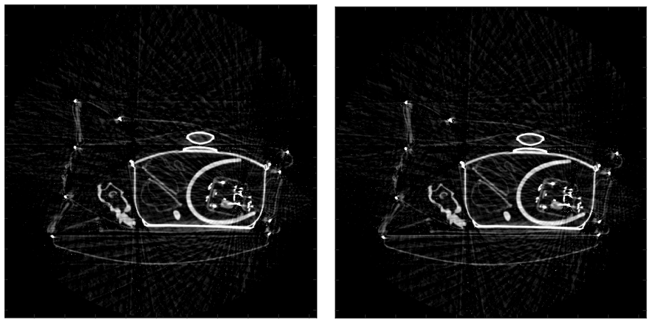


(a) w_1^* from view-subset 1 of 8 (b) w_2^* from view-subset 2 of 8

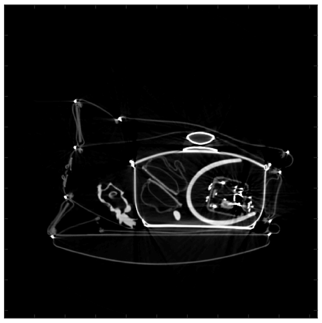


(c) $\bar{\mathbf{w}}^*$: Merged reconstruction

Figure 5. *Converged reconstruction when $N = 8$: (top) Individual sparse-view reconstructions, subsets 1 and 2. (bottom) merged consensus result.*



(a) w_1^* from view-subset 1 of 16 (b) w_2^* from view-subset 2 of 16



(c) $\bar{\mathbf{w}}^*$: Merged reconstruction

Figure 6. *Converged reconstruction when $N = 16$: (top) Individual sparse-view reconstructions, subsets 1 and 2. (bottom) merged consensus result.*

Conclusion

In this paper, we proposed a novel distributed and iterative approach to CT reconstruction called view-subset parallelization. Our approach distributes the computation of fast MBIR methods such as those based on ICD, across a large number of parallel nodes, where each node owns only a sparse subset of sinogram views and a small portion of the global system matrix. The key idea behind our approach is re-formulating the reconstruction problem as a consensus optimization problem, and we used the MACE framework to solve the same. Further, we presented a variation of MACE using partial updates to accelerate convergence of our distributed approach. Finally, we evaluated the performance of our approach through experiments with real CT data.

Acknowledgements

This work was supported in part by the U.S. Department of Homeland Security under Grant 2013-ST-061-ED000, and in part by the Showalter Trust Endowment. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security. Gregery T. Buzzard was supported in part by NSF grant DMS-1318894. We also thank Air-Force Research Labs (AFRL), Wright-Patterson Air-Force Base, Ohio, for providing us their synchrotron data of the Iron Hydroxide sample used in this work.

References

[1] Buzzard G.T., Chan S., and Bouman C.A., 'Plug-and-Play Unplugged: Optimization Free Reconstruction using Consensus Equilibrium', arXiv, 2017.

[2] Wang X., Sabne A., Sakdhnagool P., Kisner S.J., Bouman C.A., and Midkiff S.P., 'Massively Parallel 3D Image Reconstruction', ACM 2017 Supercomputing Conference (SC17).

[3] Sauer K. and Bouman C.A., 'A Local Update Strategy for Iterative Reconstruction from Projections', IEEE Transactions on Signal Processing, vol. 41 pp. 534-548, 1993.

[4] Kisner S.J., Haneda E., Bouman C.A., Skatter S., Kourinny M., Bedford S., 'Model-Based CT Reconstruction from Sparse Views', International Conference on Image Formation in X-Ray Computed Tomography, 2012.

[5] Bouman, C.A. and Sauer, K.D, 'A Unified Approach to Statistical Tomography Using Coordinate Descent Optimization', IEEE Transactions on Image Processing, vol. 5, pp. 480-492, 1996.

[6] Sreehari S., Venkatakrishnan S.V., Wohlberg B., Buzzard G.T., Drummy L.F., Simmons J.P., and Bouman C.A., 'Plug-and-Play Priors for Bright Field Electron Tomography and Sparse Interpolation,', IEEE Transactions on Computational Imaging, vol. 2, no. 4, Dec. 2016.

[7] Kisner S.J., Jin P., Bouman C.A., Sauer K., Garms W., Gable T., Oh S., Merzbacher M., and Skatter S., 'Innovative Data Weighting for Iterative Reconstruction in Helical CT Security Baggage Scanner', 47th IEEE International Carnahan Conference on Security Technology, Medellin-Colombia, October 8-11, 2013.

[8] Ollinger J.M. and Fessler J.A., 'Positron Emission Tomography', IEEE Signal Processing Magazine, Jan. 1997.

[9] Bouman C.A., Sauer K., 'A generalized Gaussian image model for edge-preserving MAP estimation', IEEE Transactions on image processing vol. 2, pp 296-310

[10] Yu Z., Thibault J.B., Bouman C.A., Sauer K.D., and Hsieh J., 'Fast Model-Based X-ray CT Reconstruction Using Spatially Non-Homogeneous ICD Optimization', IEEE Trans. on Image Processing, vol. 20, no. 1, pp. 161-175, January 2011.

[11] Boyd S., Parikh N., Chu E., Peleato B. and Eckstein J., 'Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers'

[12] De Mann B., Basu S., Thibault J.B., Hsieh J., Fessler J.A., Bouman C.A. and Sauer K., 'A Study of four Minimization Approaches for iterative reconstruction in X-ray CT', IEEE Nuclear Science Symposium Conference Record, 2005.

Author Biography

Venkatesh Sridhar received his BE (Hons.) in Electronics & Instrumentation engineering from the Birla Institute of Technology & Science, Pilani, and MS in Electrical & Computer Engineering (ECE) from Purdue University in 2014. He is working towards hid PhD in ECE at Purdue University under the guidance of Prof Charles A. Bouman. His interests are computational imaging, tomography and statistical signal processing.