2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)
April 3-7, 2020, Iowa City, Iowa, USA

BLOCK AXIAL CHECKERBOARDING: A DISTRIBUTED ALGORITHM FOR HELICAL X-RAY CT RECONSTRUCTION

Naveen Murthy, Jeffrey A. Fessler

University of Michigan, Dept. of Electrical Engineering and Computer Science, Ann Arbor, MI, USA

ABSTRACT

Model-Based Iterative Reconstruction (MBIR) methods for X-ray CT provide improved image quality compared to conventional techniques like filtered backprojection (FBP), but their computational burden is undesirably high. Distributed algorithms have the potential to significantly reduce reconstruction time, but the communication overhead of existing methods has been a considerable bottleneck. This paper proposes a distributed algorithm called Block-Axial Checkerboarding (BAC) that utilizes the special structure found in helical CT geometry to reduce inter-node communication. Preliminary results using a simulated 3D helical CT scan suggest that the proposed algorithm has the potential to reduce reconstruction time in multi-node systems, depending on the balance between compute speed and communication bandwidth.

Index Terms— Distributed algorithms, Helical X-ray CT

1. INTRODUCTION

MBIR techniques for CT provide benefits such as better image quality and potential dose reduction, compared to conventional techniques like FBP. MBIR finds the image that best matches the noisy CT measurements in accordance with factors such as the system physics, measurement statistics and prior information about the image. For all their benefits, the major problem with MBIR algorithms is their high computational costs which have hindered their routine use clinically.

There have been several approaches to accelerate iterative methods for CT reconstruction [1–3]. Distributed computing is another effective tool to move beyond the constraints of working on a single node. Simple distributed techniques for CT reconstruction assume a shared memory model, where each node has access to a common image/sinogram space, thereby simplifying parallelization. Another possible distributed implementation is one where nodes do not share a global memory space [4]. That approach requires frequent synchronization of image and projection data between nodes, after *every* subset of projection views. Although reconstruction time was reduced significantly, the data communication overhead became a serious bottleneck. One approach to reduce communication is to use block-separable surrogates [5], [6].

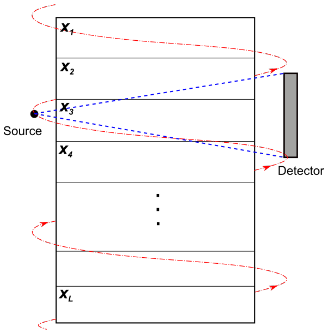


Fig. 1: Axial partition of $\boldsymbol{x}$ into blocks $\boldsymbol{x}_1, \boldsymbol{x}_2 \ldots \boldsymbol{x}_L$ such that each ray intersects at most 2 adjacent blocks.

This paper proposes a novel axial partitioning algorithm that uses the special structure in helical CT to reduce inter-node communication. The helical geometry suggests an axial partition such that every ray intersects at most 2 adjacent blocks (Fig. 1). This partition makes the problem *separable in half of the image volume*, considerably decreasing inter-node communication. Preliminary experiments are performed using a simulated helical CT scan of the XCAT phantom. Predicted wall-clock times are shown for a distributed system with multiple nodes, and these suggest that the proposed algorithm has the potential to reduce reconstruction time. This work uses a simple edge-preserving regularizer but could be extended to CNN-based regularizers.

2. PROBLEM FORMULATION

Consider the X-ray CT reconstruction problem:

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x} \succcurlyeq 0}{\operatorname{argmin}} \ \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_{\boldsymbol{W}}^2 + R(\boldsymbol{x}), \qquad (1)$$

where $\hat{\boldsymbol{x}} \in \mathbb{R}^{N_p}$ is an estimate of the unknown attenuation image ($\boldsymbol{x}$) and $\boldsymbol{y} \in \mathbb{R}^{N_d}$ is the noisy projection data. The discretized CT forward projection operator [7] is denoted by $\boldsymbol{A} \in \mathbb{R}^{N_d \times N_p}$. $\boldsymbol{W} \in \mathbb{R}^{N_d \times N_d}$ is a diagonal statistical weighting matrix. $R(\boldsymbol{x})$ is an edge-preserving regularizer, typically of the form $R(\boldsymbol{x}) = \sum_{n=1}^{N} \psi_n \left([\boldsymbol{C}\boldsymbol{x}]_n \right)$, where

Supported in part by NIH Grant U01 EB018753.

978-1-5386-9330-8/20/$31.00 ©2020 IEEE

$\boldsymbol{C} \in \mathbb{R}^{N \times N_p}$ is a stack of finite difference matrices and ψ is an edge-preserving penalty function.

3. PREVIOUS DISTRIBUTED APPROACH

For the baseline comparison, we use a distributed implementation without a shared memory space [4]. The image update uses the OS-RLALM algorithm [8]. This baseline implementation is referred to as Std-OS-RLALM henceforth. This algorithm distributes forward projection of the image among multiple nodes (parallelized by projection views). The results of forward projection are broadcast to every other node (all-to-all broadcast), since every node needs the whole sinogram residual to compute the gradient of the data-fit term.

The steps of backprojection, regularization and image update are parallelized across multiple nodes without requiring data synchronization between each of the steps. Each node then broadcasts its portion of the updated image to every other node (all-to-all broadcast) and we move on to the forward projection of the next subset of views. The method must transfer a large amount of data (essentially one image-sized variable and one subset-sized projection variable), for *every* subset.

The major drawback with this distributed implementation is the large communication overhead, especially for more nodes and subsets. To combat these issues, we propose an algorithm that drastically reduces the communication requirements, by partitioning the image volume appropriately.

4. PROPOSED BLOCK AXIAL CHECKERBOARDING (BAC) ALGORITHM

This section shows that, under appropriate axial partitions of the image, the cost function is completely separable in half the image volume. Exploiting this separability, we propose an algorithm that alternatingly updates half of the image slices (keeping the other half fixed), and vice versa, that reduces inter-node communication in a distributed system.

4.1. Partial separability of datafit term

Consider an axial partition of the image volume $\boldsymbol{x}$ into L blocks, i.e., $\boldsymbol{x} = [\boldsymbol{x}_1 \; \boldsymbol{x}_2 \; \dots \; \boldsymbol{x}_L]^T$, where each block is a collection of axial slices. Let the forward projection matrix $\boldsymbol{A}$ be correspondingly partitioned into blocks of columns, as $\boldsymbol{A} = [\boldsymbol{A}_1 \; \boldsymbol{A}_2 \; \dots \; \boldsymbol{A}_L]$. Let $P_l \triangleq \{k \; : \; [\boldsymbol{A}_l \, \boldsymbol{x}_l]_k \neq 0\}$, i.e., P_l is the set of all rays that intersect axial block $\boldsymbol{x}_l$.

Using the banded structure of the system matrix $\boldsymbol{A}$ for helical CT reconstruction (shown using $\boldsymbol{A}'\boldsymbol{A}$ in Fig. 2), we axially partition the image volume such that *every ray intersects at most 2 adjacent blocks*. The number of slices in each axial block depends on factors such as cone-beam angle and field-of-view. Without loss of generality, let us consider equally sized axial blocks. Fig. 1 illustrates this partition, and results in the condition:

$$P_l \cap P_m = \emptyset, \qquad \forall l, m : |l - m| > 1. \tag{2}$$

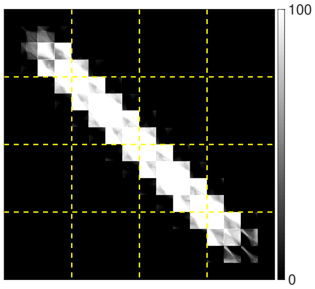


Fig. 2: Image showing the banded structure of the matrix $\boldsymbol{A}'\boldsymbol{A}$ (yellow lines indicate the axial block boundaries).

Claim: Keeping the odd-numbered blocks of $\boldsymbol{x}$ fixed, the data-fit term in (1) becomes separable in each of the even-numbered blocks of $\boldsymbol{x}$.

Proof: Let $D(\boldsymbol{x}) = \sum_{i=1}^{N_d} h_i([\boldsymbol{A}\boldsymbol{x}]_i)$ represent the data-fit term, where $h_i(t) = \frac{1}{2}(y_i - t)^2$ for a PWLS formulation. Expanding $D(\boldsymbol{x})$ in terms of the axial blocks, we obtain $D(\boldsymbol{x}) = \sum_{i=1}^{N_d} h_i \left(\left[\sum_{k=1}^{L} \boldsymbol{A}_k \boldsymbol{x}_k \right]_i \right)$.

Consider the update of an arbitrary even-numbered block $\boldsymbol{x}_l$. Let $\tilde{\boldsymbol{x}}_{l-1}$ and $\tilde{\boldsymbol{x}}_{l+1}$ be the fixed values of the adjacent image blocks $l-1$ and $l+1$. To obtain the terms of $D(\boldsymbol{x})$ relevant to the block $\boldsymbol{x}_l$, we consider only the rays in P_l. Rewriting P_l as the union of 3 disjoint sets (from (2)), we obtain the block-specific data-fit term $D_l(\boldsymbol{x}_l)$ as

$$\begin{aligned} D_l(\boldsymbol{x}_l) = & \sum_{i \in (P_l \cap P_{l-1})} h_i([\boldsymbol{A}_l \boldsymbol{x}_l + \boldsymbol{A}_{l-1} \tilde{\boldsymbol{x}}_{l-1}]_i) \\ & + \sum_{i \in (P_l \setminus (P_{l-1} \cup P_{l+1}))} h_i([\boldsymbol{A}_l \boldsymbol{x}_l]_i) \\ & + \sum_{i \in (P_l \cap P_{l+1})} h_i([\boldsymbol{A}_l \boldsymbol{x}_l + \boldsymbol{A}_{l+1} \tilde{\boldsymbol{x}}_{l+1}]_i) \\ = & \sum_{i \in P_l} h_i([\boldsymbol{A}_l \boldsymbol{x}_l \; + \; \tilde{\boldsymbol{b}}_l]_i), \end{aligned} \tag{3}$$

where $\tilde{\boldsymbol{b}}_l$, which we call the data-fit offset, is a fixed vector given by $\tilde{\boldsymbol{b}}_l = [\vec{0} \; \dots \; (\boldsymbol{A}_{l-1} \tilde{\boldsymbol{x}}_{l-1}) \; \vec{0} \; (\boldsymbol{A}_{l+1} \tilde{\boldsymbol{x}}_{l+1}) \; \dots \; \vec{0}]^T$. The zeros in $\tilde{\boldsymbol{b}}_l$ are shown for notational simplicity, and an efficient implementation transfers *only the relevant forward projection results* from the adjacent blocks $l-1$ and $l+1$. A key property of our approach is that it is *never necessary* to transfer complete sinogram data between nodes. In the PWLS case, the datafit term simplifies to

$$D_l(\boldsymbol{x}_l) = \frac{1}{2} \left\| \tilde{\boldsymbol{y}}_l - \boldsymbol{A}_l \boldsymbol{x}_l \right\|_{\boldsymbol{W}}^2, \qquad l = 2, 4, \dots, L, \tag{4}$$

where $\tilde{\boldsymbol{y}}_l = \boldsymbol{y} - \tilde{\boldsymbol{b}}_l$. Similarly, the data-fit term is completely separable in the odd-numbered blocks (when the even blocks are fixed), and is of a similar form as (4).

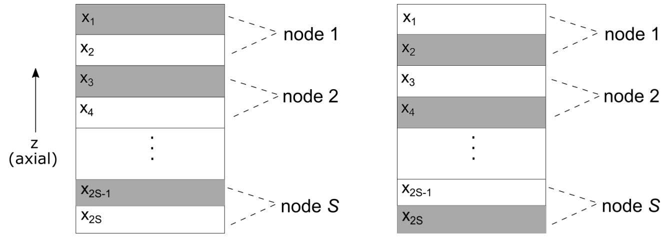


Fig. 3: Alternating updates of odd and even blocks (blocks in grey are simultaneously updated).

4.2. Partial separability of regularizer term

Using a similar proof as in Section 4.1, we obtain a block-specific regularizer term $R_l(\boldsymbol{x}_l)$ for the update of block l. We show that $R_l(\boldsymbol{x}_l)$ requires the last slice from block $l-1$ and the first slice from block $l+1$. Intuitively, this makes sense, since we are regularizing across block boundaries.

The overall cost function for updating the lth block (when alternate blocks are kept fixed) is given by

$$f_l(\boldsymbol{x}_l) = D_l(\boldsymbol{x}_l) + R_l(\boldsymbol{x}_l), \qquad l = 1, 3, 5, \ldots, L-1, \\ \text{or} \quad l = 2, 4, 6, \ldots, L. \qquad (5)$$

4.3. Implementation

Consider a distributed system of S nodes. Each node could consist of CPU cores or single/multiple GPUs [9, 10] or a combination of both. As shown in Fig. 3, we axially partition the image volume into $2S$ blocks and each node works on 2 adjacent blocks (one odd and one even block). Using more nodes S potentially speeds up reconstruction, though the block size needs to be large enough to satisfy the requirement that every ray intersect at most 2 adjacent axial blocks.

Let N_o be the number of outer iterations. As shown in Algorithm 1, each node simultaneously updates its odd (even) block before transferring the relevant offset terms to its adjacent node. Each of these updates is performed for multiple iterations (N_i in Algorithm 1), since the cost function is completely separable in the alternate blocks (shown in Sections 4.1 and 4.2). This lessens the need for frequent data communication between nodes, hence reducing time. The update of each individual block could be performed using any fast iterative algorithm that is suitable for 3D X-ray CT reconstruction.

4.4. Communication overhead

The potential reductions in communication overhead of the proposed algorithm (BAC) instead of the standard algorithm (Std-OS-RLALM) are three-fold: (i) Each node transfers data to only one other node (instead of an all-to-all broadcast). (ii) Data is transferred only once per outer iteration (instead of once per subset). (iii) Amount of data transferred is much smaller (only block-specific offset terms for block l). The cost of the additional forward projections required in BAC (see Algorithm 1) is offset by the potentially huge reductions in communication time between nodes.

Algorithm 1 Block Axial Checkerboarding (BAC)

```
Initialize $\boldsymbol{x}^{(0)}$ with FBP image.
Divide $\boldsymbol{x}^{(0)}$ into $2S$ axial blocks; distribute among $S$ nodes (2 blocks per node).
for $n = 0, 1, \ldots, N_o$ do
    for $s = 1, 2, \ldots, S$ simultaneously do
        Update odd-numbered block $\boldsymbol{x}_l^{(n)}$, for $l = 2s - 1$, using $N_i$ iterations of any recon algorithm.
        Transfer offset term $\boldsymbol{A}_l \boldsymbol{x}_l^{(n)}$ and the first slice of $\boldsymbol{x}_l^{(n)}$ to node $s - 1$.
        Update even-numbered block $\boldsymbol{x}_l^{(n)}$, for $l = 2s$, using $N_i$ iterations of any recon algorithm.
        Transfer offset term $\boldsymbol{A}_l \boldsymbol{x}_l^{(n)}$ and the last slice of $\boldsymbol{x}_l^{(n)}$ to node $s + 1$.
    end for
end for
```

5. EXPERIMENTAL RESULTS

We simulated a helical CT scan using a 512 × 512 × 320 volume of the XCAT phantom [11]. The helical scan consisted of 5.25 helical turns with a pitch of 63/64, resulting in a sinogram of size 444 × 32 × 5166 ([detector columns] × [rows] × [projection views]). We reconstructed the image volume on a coarser grid of size 256 × 256 × 320. For the regularizer, we have used 13 finite differences to include all 26 neighbors of a voxel. We use the Fair potential function for ψ.

Fig. 4 shows reconstructions of a central slice of the XCAT phantom. The reference image $\boldsymbol{x}^*$ was obtained after running 100 iterations of OS-RLALM (18 subsets) followed by 1200 iterations of OS-RLALM with 1 subset. The proposed algorithm BAC was run for 6 iterations, and each odd/even image block was updated using OS-RLALM (5 inner iterations, 18 subsets).

Fig. 5(a) shows the convergence plot vs. equits. The convergence metric is root mean square difference (RMSD) computed as $\sqrt{\frac{1}{|\Omega|} \sum_{i \in \Omega} |\boldsymbol{x}_i^{(k)} - \boldsymbol{x}_i^*|^2}$, where Ω is a cylindrical region-of-interest. One equit corresponds to one complete forward and backprojection, and using equits ensures a fair comparison between the competing algorithms. This plot was generated using two nodes (S = 2), each with a single TitanX (Pascal) GPU. To see the potential benefits in a realistic distributed system, we simulate the predicted wall clock times for a 4-node system, each with a single GPU (Fig.

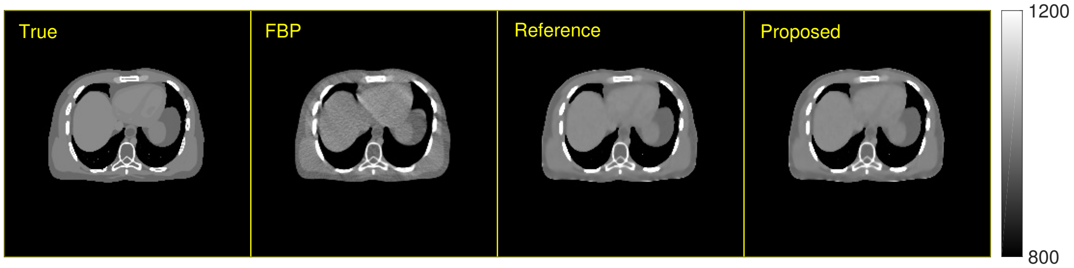
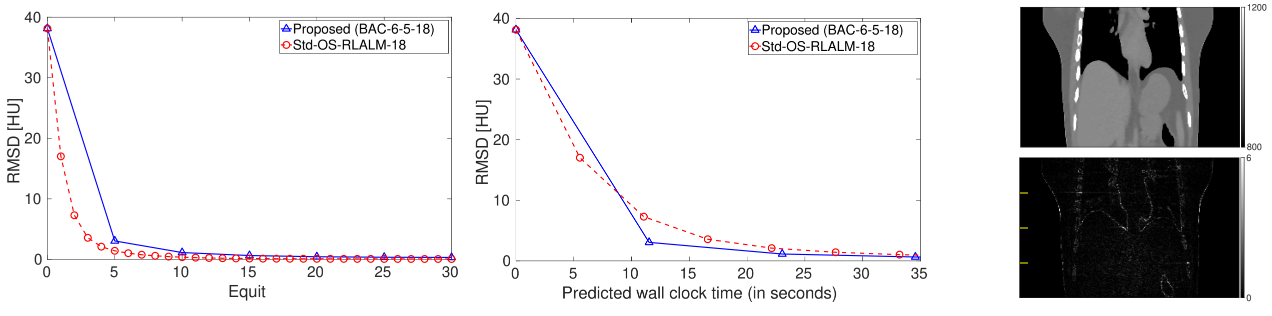


Fig. 4: Reconstruction of central transaxial slice of XCAT phantom. Images are displayed in HU (modified so that air is 0).



(a)

(b)

(c)

Fig. 5: Convergence plots (a) vs. equit and (b) vs. predicted wall clock time. (c) Coronal view of the BAC reconstructed image and its absolute difference with the reference image $\boldsymbol{x}^*$ (yellow ticks indicate axial block boundaries).

5(b)). We assume an ethernet connection between the nodes (with a maximum bandwidth of 10Gbps), and we extrapolate forward/backprojection times from a single GPU. As expected, Std-OS-RLALM requires fewer equits than BAC to reach a lower RMSD value; however, the huge communication overhead of Std-OS-RLALM suggests that the predicted recon time would be higher than BAC (see Fig. 5(b)). The improvement in recon time is predicted to be much greater when using more nodes and subsets.

6. CONCLUSION

This paper proposes a novel partitioning algorithm that utilizes the special structure found in helical CT to reduce frequent data communication between the nodes in a distributed system. Preliminary predicted wall-clock times suggest that the proposed BAC algorithm could potentially be faster than the standard Std-OS-RLALM approach. We observe negligible boundary artefacts at the block boundaries (see Fig. 5(c)), which could be further reduced using dithering [6]. Implementation on a real distributed multi-node platform is required as future work.

7. REFERENCES

[1] K. Sauer and C. Bouman, "A local update strategy for iterative reconstruction from projections," *IEEE Trans. Sig. Proc.*, vol. 41, no. 2, pp. 534–48, Feb. 1993.

[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. 4526–44, Nov. 2007.

[3] D. Kim, S. Ramani, and J. A. Fessler, "Combining ordered subsets and momentum for accelerated X-ray CT image reconstruction," *IEEE Trans. Med. Imag.*, vol. 34, no. 1, pp. 167–78, Jan. 2015.

[4] J. M. Rosen, J. Wu, T. F. Wenisch, and J. A. Fessler, "Iterative helical CT reconstruction in the cloud for ten dollars in five minutes," in *Proc. Intl. Mtg. on Fully 3D Image Recon. in Rad. and Nuc. Med*, 2013, pp. 241–4.

[5] S. Sotthivirat and J. A. Fessler, "Image recovery using partitioned-separable paraboloidal surrogate coordinate ascent algorithms," *IEEE Trans. Im. Proc.*, vol. 11, no. 3, pp. 306–17, Mar. 2002.

[6] D. Kim and J. A. Fessler, "Distributed block-separable ordered subsets for helical X-ray CT image reconstruction," in *Proc. Intl. Mtg. on Fully 3D Image Recon. in Rad. and Nuc. Med*, 2015, pp. 138–41.

[7] Y. Long, J. A. Fessler, and J. M. Balter, "3D forward and back-projection for X-ray CT using separable footprints," *IEEE Trans. Med. Imag.*, vol. 29, no. 11, pp. 1839–50, Nov. 2010.

[8] H. Nien and J. A. Fessler, "Relaxed linearized algorithms for faster X-ray CT image reconstruction," *IEEE Trans. Med. Imag.*, vol. 35, no. 4, pp. 1090–8, Apr. 2016.

[9] A. S. Wang, J. W. Stayman, Y. Otake, G. Kleinszig, S. Vogt, and J. H. Siewerdsen, "Nesterov's method for accelerated penalized-likelihood statistical reconstruction for C-arm cone-beam CT," in *Proc. 3rd Intl. Mtg. on image formation in X-ray CT*, 2014, pp. 409–13.

[10] X. Xie, M. G. McGaffin, Y. Long, J. A. Fessler, M. Wen, and J. Lin, "Accelerating separable footprint (SF) forward and back projection on GPU," in *Proc. SPIE 10132 Medical Imaging: Phys. Med. Im.*, 2017, p. 101322S.

[11] W. P. Segars, M. Mahesh, T. J. Beck, E. C. Frey, and B. M. W. Tsui, "Realistic CT simulation using the 4D XCAT phantom," *Med. Phys.*, vol. 35, no. 8, pp. 3800–8, Aug. 2008.