Parallelizable algorithms for X-ray CT image reconstruction with spatially non-uniform updates

Donghwan Kim and Jeffrey A. Fessler

***Abstract*—Statistical image reconstruction methods for X-ray CT provide good images even for reduced dose levels but require substantial compute time. Iterative algorithms that converge in fewer iterations are preferable. Spatially non-homogeneous iterative coordinate descent (NH-ICD) accelerates convergence by updating more frequently the voxels that are predicted to change the most between the current image and the final image. However, the sequential update of NH-ICD reduces parallelism opportunities.**

This paper focuses on iterative algorithms that are more amenable to parallelization, namely the axial block coordinate descent (ABCD) algorithm and an ordered subsets algorithm based on separable quadratic surrogates (OS-SQS), because these have the potential to be faster than ICD in multiprocessor implementations. We first adapt the "non-homogeneous" approach to ABCD, which simply requires updating more frequently the axial blocks that are predicted to change the most during convergence. More interestingly, we derive a new version of the OS-SQS algorithm that leads to *spatially non-uniform updates* with larger step sizes for the voxels that are predicted to change the most between the current image and the final image. The single subset version of this algorithm is still guaranteed to converge monotonically.

We use a 3D patient CT scan to demonstrate that the proposed algorithms with spatially non-uniform updates converge faster than the ordinary algorithms. In particular, the NU approach accelerated the OS-SQS algorithm by a factor of three.

I. Introduction

Statistical image reconstruction for X-ray CT can provide good images even with reduced dose levels [1]. However, the substantial compute time required for the iterative algorithms is a drawback. This paper describes acceleration methods for parallelizable algorithms for penalized weighted least-squares (PWLS) image reconstruction.

Iterative coordinate descent (ICD) is a convergent method that can converge to a reconstructed image that is close to the minimizer of the PWLS cost function in a small number of iterations when initialized appropriately [1]. However, ICD updates each voxel sequentially so it is relatively difficult to parallelize. To accelerate ICD, one can try to predict which voxels will change the most between the current image and the final image, and then update those voxels more frequently. This non-homogeneous (NH) approach to ICD, called NH-ICD [2], can reduce the number of iterations needed but does not affect the parallelizability.

Considering the modern parallel computing architecture, we focus on two parallelizable algorithms: axial block coordinate descent (ABCD) [3] and an ordered subsets (OS) algorithm based on separable quadratic surrogates (SQS) called OS-SQS [4]. When appropriately parallelized, these algorithms should require less time per iteration, but need more iterations to converge than NH-ICD. (ABCD needed a similar number of iterations as ICD in one preliminary simulation [3], whereas OS-SQS needed far more iterations than ICD [5].) Inspired by the success of NH-ICD, in this paper, we develop similar acceleration methods for ABCD and OS-SQS. Applying the NH idea to ABCD is straightforward; we simply update more frequently the axial blocks that we predict will change the most during convergence. However, the original NH idea is not directly applicable to OS-SQS because it updates all voxels simultaneously. In this paper we derived a new version of the OS-SQS algorithm that leads to spatially non-uniform (NU) updates. Specifically, we design the surrogate functions so that the resulting iterations take larger step sizes for voxels that are predicted to change the most during convergence. Importantly, the theoretical derivation ensures that the new SQS algorithm (the one-subset version of the OS-SQS algorithm) is still guaranteed to converge monotonically. The derivation uses a modification of De Pierro's approach [6]. The resulting algorithm still updates all voxels simultaneously and thus is amenable to parallelization.

NH-ABCD and NU-OS-SQS are designed to work efficiently with the separable footprint (SF) projector [7]. The axial/transaxial separability of the SF projector facilitated the proposed algorithm to be highly efficient and parallelizable. We examined the performance of the proposed algorithms using a 3D patient CT scan. The results show that the proposed spatially non-uniform algorithms converge much faster than the ordinary algorithms. The proposed NU approach accelerated the OS-SQS algorithm by about a factor of three.

II. Problem

We reconstruct an image $x \in \mathbb{R}^N$ from a noisy CT measurement data $y \in \mathbb{R}^M$ by finding the minimizer $\hat{x}$ of the following PWLS cost function [1]:

$$\begin{aligned}\Psi(x) = Q(x) + \beta R(x) &= \frac{1}{2}||y - Ax||_W^2 + \beta R(x) \\ &= \sum_{i=1}^{M} q_i([Ax]_i) + \beta \sum_{r=1}^{K} \psi_r([Cx]_r), \end{aligned} \quad (1)$$

where A is a system matrix (projector), C is a finite differencing matrix, $W = \mathsf{diag}\{w_i\}$ is a statistical weighting for measurement data, $q_i(t) = \frac{1}{2} w_i (t - y_i)^2$, each $\psi_r(t)$ is a (edge preserving) potential function, and β is a regularization parameter. Our goal is to find the minimizer $\hat{x}$ more efficiently.

Dept. of Electrical Engin. and Computer Science, Univ. of Michigan, Ann Arbor, MI 48109 USA (e-mail: kimdongh@umich.edu, fessler@umich.edu).
Supported in part by NIH grant 1-R01-HL-098686.

III. Spatially non-homogeneous axial block coordinate descent (NH-ABCD)

A. Algorithm

ABCD sequentially updates each axial block of voxels [3]. The low coupling between voxels within an axial block simplifies the update [3]. Traditional ABCD updates the axial block sequentially, but the update order is flexible so we can easily adapt the NH idea of NH-ICD for the ABCD algorithm.

Let $x_k^{(n)}$ denote the vector of voxel values along the kth axial block at the nth iteration, and let $x_k^{(\infty)}$ denote the corresponding converged values, where k ranges from 1 to the number of voxels in one transaxial plane. One way to describe how much the voxels change between the nth iteration and the converged image is by this "*update-needed factor*" [2]:

$$\hat{u}_k^{(n)} = \left|\left| x_k^{(n)} - x_k^{(\infty)} \right|\right|_1.$$

Ideally NH-ICD would order the voxel updates based on $\hat{u}_k^{(n)}$, updating more frequently voxels within axial blocks having larger values of $\hat{u}_k^{(n)}$, accelerating convergence. However, $\hat{u}_k^{(n)}$ is unavailable at the nth iteration in practice, so NH-ICD uses the following factor instead:

$$u_k^{(n)} = \left|\left| x_k^{(n)} - x_k^{(n-1)} \right|\right|_1, \tag{2}$$

which is the difference between the current and previous kth axial block. (In addition $u_k^{(n)}$ is low-pass filtered to try to improve $u_k^{(n)}$.) In practice, the NH-ICD approach uses both homogeneous update orders and non-homogeneous update orders based on $u_k^{(n)}$ for fast convergence overall.

We adapted these NH ideas to the ABCD algorithm, yielding NH-ABCD, by non-uniformly updating axial blocks. We implemented a SQS version of NH-ABCD (NH-ABCD-SQS) that we expected to converge faster than ABCD-SQS.

IV. Spatially non-uniform separable quadratic surrogate (NU-SQS) approach

A. SQS Algorithm

SQS for PWLS has the benefit of low computation per iteration and high parallelizability [4]. However, it needs many iterations to converge. This section presents a new SQS algorithm that uses spatially non-uniform updates to accelerate convergence without reducing parallelizability.

In a simultaneous update algorithm like SQS, the idea of updating certain voxels more frequently is unnatural. Instead, we re-derive the algorithm to increase the *step size* of voxels that are predicted to need to change more during convergence. Simply weighting the step size arbitrarily would break the monotonicity of optimization, so instead we derive an appropriate weighting scheme that preserves the monotonicity (in the one subset version) by adapting De Pierro's approach [6].

For completeness, we repeat De Pierro's argument in [4]. We first rewrite forward projection $[Ax]_i$ as follows:

$$[Ax]_i = \sum_{j=1}^{N} a_{ij} x_j = \sum_{j=1}^{N} \pi_{ij}^{(n)} \left(\frac{a_{ij}}{\pi_{ij}^{(n)}} (x_j - x_j^{(n)}) + [Ax^{(n)}]_i \right),$$

where $\sum_{j=1}^{N} \pi_{ij}^{(n)} = 1$ and $\pi_{ij}^{(n)}$ is zero only if a_{ij} is zero. Using the convexity of $q_i(\cdot)$ and the convexity inequality:

$$q_i([Ax]_i) \le \sum_{j=1}^{N} \pi_{ij}^{(n)} \, q_i \left(\frac{a_{ij}}{\pi_{ij}^{(n)}} (x_j - x_j^{(n)}) + [Ax^{(n)}]_i \right).$$

Thus we have the following separable quadratic surrogate $\phi_Q^{(n)}(x)$ for the data-fit term $Q(x)$:

$$\begin{aligned} Q(x) &\le \phi_Q^{(n)}(x) \triangleq \sum_{j=1}^{N} \phi_{Q,j}^{(n)}(x_j) \\ &= \sum_{i=1}^{M} \sum_{j=1}^{N} \pi_{ij}^{(n)} \, q_i \left(\frac{a_{ij}}{\pi_{ij}^{(n)}} (x_j - x_j^{(n)}) + [Ax^{(n)}]_i \right). \end{aligned} \tag{3}$$

The second derivative of the surrogate $\phi_{Q,j}^{(n)}(x_j)$ is

$$d_j^{Q,(n)} \triangleq \frac{\partial^2}{\partial x_j^2} \phi_{Q,j}^{(n)}(x_j) = \sum_{i=1}^{M} w_i a_{ij}^2 / \pi_{ij}^{(n)}.$$

Then the step size $\Delta_j^{(n)}$ of SQS [4] has this relationship:

$$\Delta_j^{(n)} \triangleq x_j^{(n+1)} - x_j^{(n)} \propto \frac{1}{d_j^{Q,(n)}} \propto \pi_{ij}^{(n)}, \tag{4}$$

where small $d_j^{Q,(n)}$ and large $\pi_{ij}^{(n)}$ values lead to larger steps. Therefore we should encourage $\pi_{ij}^{(n)}$ to be large to accelerate the SQS algorithm, subject to the condition $\sum_{j=1}^{N} \pi_{ij}^{(n)} = 1$.

The standard choice [4], [8] is $\pi_{ij}^{(n)} = \frac{a_{ij}}{\sum_{l=1}^{N} a_{il}}$, leading to

$$d_j^{Q,(n)} = \sum_{i=1}^{M} w_i a_{ij} \left(\sum_{l=1}^{N} a_{il} \right). \tag{5}$$

This choice does not exploit the relationship (4). Thus, we propose to choose $\pi_{ij}^{(n)}$ to be larger if the jth voxel is predicted to need more update based on the following "update-needed factor" after the nth iteration:

$$u_j^{(n)} = \left| x_j^{(n)} - x_j^{(n-1)} \right|. \tag{6}$$

We select $\pi_{ij}^{(n)} = \frac{a_{ij} u_j^{(n)}}{\sum_{l=1}^{N} a_{il} u_l^{(n)}}$ which is proportional to $u_j^{(n)}$ and satisfies the conditions for $\pi_{ij}^{(n)}$. This choice for $\pi_{ij}^{(n)}$ leads to the following NU-based denominator:

$$\tilde{d}_j^{Q,(n)} = \frac{1}{u_j^{(n)}} \sum_{i=1}^{M} w_i a_{ij} \left(\sum_{l=1}^{N} a_{il} u_l^{(n)} \right), \tag{7}$$

which leads to spatially non-uniform updates $\Delta_j^{(n)} \propto u_j^{(n)}$. Computing (7) requires one forward and back projection which increases computation, but Sec. IV-B explains how to minimize this effect. The NU-based denominator (7) reduces to the standard denominator (5) when $u_j^{(n)}$ is uniform.

Recall that NH-ICD balanced between the uniform and non-uniform voxel ordering to provide fast convergence. Likewise, using values for $u_j^{(n)}$ with too large of dynamic range that would focus most of the updates on a few voxels would likely be undesirable. Therefore we modified the "update needed

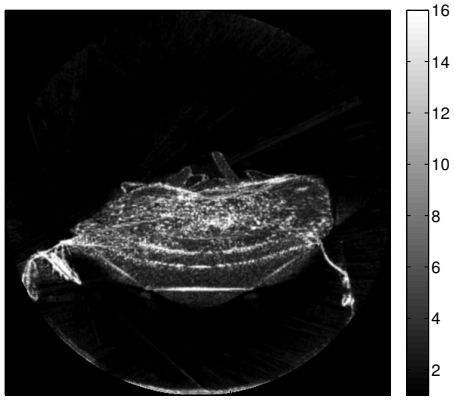

Fig. 1. Dynamic range compression (DRC) applied $u_j^{(4)}$ for NU-OS-SQS. In each case we map $u_j^{(n)}$ to 16 for the largest 5% voxels, to [8 4 2] for next [10% 20% 40%] voxels, and to 1 for the rest of the voxels, followed by low-pass filtering. NU-OS-SQS updates more the bright voxels, whereas ordinary OS-SQS updates all voxels equivalently.

factors" $u_j^{(n)}$ to have a reasonable dynamic range (see Fig. 1), which we call dynamic range compression (DRC).

Similar to the data-fit term, we derive the denominator of NU-SQS for the regularizer term to be:

$$\tilde{d}_j^{R,(n)} = \frac{1}{u_j^{(n)}} \sum_{r=1}^{K} \ddot{\psi}_r(0) |c_{rj}| \left(\sum_{l=1}^{N} |c_{rl}| u_l^{(n)} \right), \qquad (8)$$

by using the choice $\pi_{rj}^{(n)} = \frac{|c_{rj}| u_j^{(n)}}{\sum_{l=1}^{N} |c_{rl}| u_l^{(n)}}$ and using the maximum curvature $\ddot{\psi}_r(0) = \max_t \ddot{\psi}_r(t)$ for efficiency [4]. The computation of (8) is negligible compared to that of data-fit term.

Combining the above derivations leads to the following simple and parallelizable NU-SQS iteration:

$$x^{(n+1)} = x^{(n)} - \mathsf{diag}\left\{ \frac{1}{\tilde{d}_j^{Q,(n)} + \beta \tilde{d}_j^{R,(n)}} \right\} \nabla \Psi(x^{(n)}).$$

This algorithm monotonically decreases $\Psi(x)$ and is provably convergent [9]. We can further accelerate NU-SQS by using ordered subsets (OS) of projection views [4], [10] which we call NU-OS-SQS.

B. Implementation

The dependence of $\pi_{ij}^{(n)}$ on $u_j^{(n)}$ increases computation, but we found two practical way to reduce the burden. First, we found that it suffices to update $u_j^{(n)}$ every few iterations instead of every iteration. Second, in 3D CT we use forward and back-projectors that compute elements of the system matrix A on the fly, and as those elements are computed for gradient of $Q(x)$, which requires one forward and back projection, we simultaneously compute the forward and back-projection needed for the NU-based denominator (7). For the results shown below, we computed $u_j^{(n)}$ during one iteration and computed the NU-based denominator (7) during the next iteration, and then used it for several iterations. For the first iteration we form $u_j^{(0)}$ using a combination of edge and intensity detector. This is reasonable as the initial FBP is a good low-frequency estimate, so $\hat{u}_j^{(0)}$ will be bigger for voxels near edges.

C. Application of NU-SQS in ABCD algorithm

We also tried to further accelerate the ABCD algorithm by applying the NU-SQS principle to ABCD using the following NU-based denominator:

$$\tilde{d}_j^{Q,(n)} = \frac{1}{u_j^{(n)}} \sum_{i=1}^{M} w_i a_{ij} \left(\sum_{l \in \mathcal{B}_k} a_{il} u_l^{(n)} \right), \quad j \in \mathcal{B}_k, \qquad (9)$$

where $\mathcal{B}_k$ denotes the indices of the voxels in the kth axial block. For a typical multi-slice CT geometry, the set $\{l : a_{il} > 0,\ l \in \mathcal{B}_k\}$ contains at most three (adjacent) voxels with similar $u_l^{(n)}$ values, and the resulting acceleration was minimal. However, block coordinate descent (BCD) algorithms [11] that group voxels in transaxial plane could exploit non-uniformity.

V. Result

We implemented the proposed algorithms in C and applied them to a helical patient CT scan. We examined spatially nonuniform approaches for ABCD and OS-SQS algorithms in terms of convergence rate and compute time per iteration. Our implementations are not optimized in terms of run time, so we show the results of each method separately. Fig. 2 and Fig. 4 show the root mean squared (RMS) difference (in HU) from the converged image[1] versus normalized run time for NH-ABCD and NU-OS-SQS. The run time of the algorithms are normalized in time by one iteration of ABCD-SQS and OS-SQS respectively, and the plot markers show each iteration. Image reconstruction included the nonnegativity constraint.

In Fig. 2, NH-ABCD converged 3× faster than ABCD, similar to the acceleration of NH-ICD in [2]. Using the NH idea in the ABCD algorithm increased compute time per iteration by only 3%.

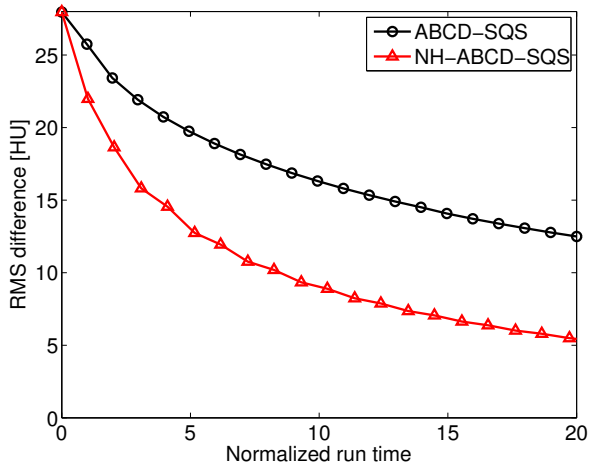

Fig. 2. RMS difference [HU] from converged image vs. normalized compute time for previous ABCD-SQS [3] and proposed NH-ABCD-SQS. Compute time is normalized by the elapsed time for one iteration of ABCD-SQS.

[1]We generated an (almost) converged image by running 100 iterations of NH-ABCD-SQS followed by 2000 iterations of SQS.

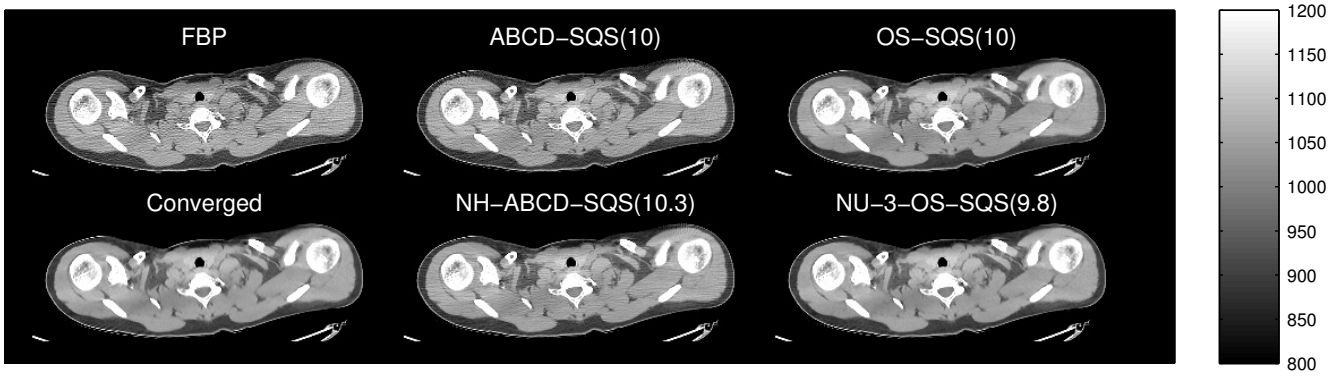

Fig. 3. FBP image $x^{(0)}$, converged image $\hat{x}$, and reconstructed images by ABCD and OS-SQS algorithms. Numbers in parentheses represent normalized compute time; ABCD and OS-SQS have different normalized compute time. The proposed NH and NU methods each accelerate convergence to $\hat{x}$.

In Fig. 4, the NU approach accelerated the OS algorithm by a factor of three. Incorporating the computation of the NU-based denominator (7) simultaneously with the gradient increased run time by 25%, but this increase was amortized by updating the NU-based denominator only every few iterations. Fig. 4 suggests that every 3-5 iterations is enough.

Compressing the dynamic range of $u_j^{(n)}$, as shown in Fig. 1, was essential to accelerate convergence compared with solely using (6). The DRC approach in Fig. 1 is just one of many possibilities, and we expect to find other candidates that will lead to even faster convergence.

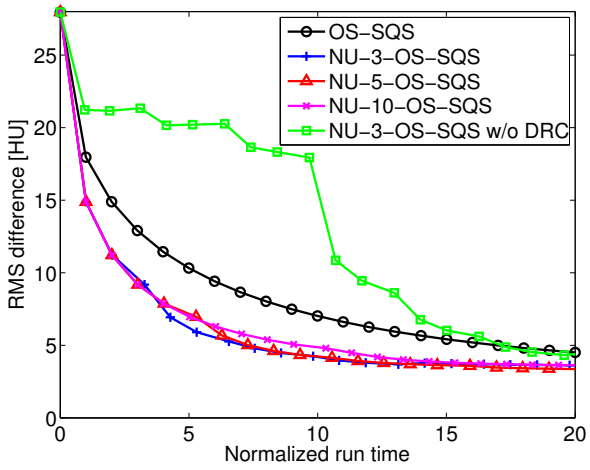

Fig. 4. RMS difference [HU] to converged image vs. normalized compute time for NU-OS-SQS with 246 subsets. Compute time is normalized by the elapsed time of one iteration of OS-SQS. Number in the legend indicates how often we update the NU-based denominator.

Fig. 3 shows the center slice of FBP, the converged image $\hat{x}$, and reconstructed images by ABCD and OS-SQS methods. The quality of $\hat{x}$ compared to FBP reaffirms the benefits of statistical image reconstruction. The reconstructed images with the proposed spatially non-uniform approaches are closer to the converged image $\hat{x}$ than the ordinary ABCD and OS-SQS reconstructed images.

VI. Discussion

We have used spatially non-uniform updates to accelerate parallelizable iterative algorithms ABCD and OS-SQS. In particular, we derived a new spatial non-uniformity approach for SQS, a simultaneous update algorithm, which improved the convergence rate by about a factor of three. The next step is to optimize the implementation in terms of compute time and parallelization, and compare the proposed algorithms with other algorithms such as NH-ICD.

References

[1] 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.

[2] Z. Yu, J-B. Thibault, C. A. Bouman, K. D. Sauer, and J. Hsieh, "Fast model-based X-ray CT reconstruction using spatially non-homogeneous ICD optimization," *IEEE Trans. Im. Proc.*, vol. 20, no. 1, pp. 161–75, Jan. 2011.

[3] J. A. Fessler and D. Kim, "Axial block coordinate descent (ABCD) algorithm for X-ray CT image reconstruction," in *Proc. Intl. Mtg. on Fully 3D Image Recon. in Rad. and Nuc. Med*, 2011, pp. 262–5.

[4] H. Erdoğan and J. A. Fessler, "Ordered subsets algorithms for transmission tomography," *Phys. Med. Biol.*, vol. 44, no. 11, pp. 2835–51, Nov. 1999.

[5] B. De Man, S. Basu, J-B. Thibault, J. Hsieh, J. A. Fessler, C. Bouman, and K. Sauer, "A study of different minimization approaches for iterative reconstruction in X-ray CT," in *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, 2005, vol. 5, pp. 2708–10.

[6] A. R. De Pierro, "A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography," *IEEE Trans. Med. Imag.*, vol. 14, no. 1, pp. 132–7, Mar. 1995.

[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. Erdoğan and J. A. Fessler, "Monotonic algorithms for transmission tomography," *IEEE Trans. Med. Imag.*, vol. 18, no. 9, pp. 801–14, Sept. 1999.

[9] M. W. Jacobson and J. A. Fessler, "An expanded theoretical treatment of iteration-dependent majorize-minimize algorithms," *IEEE Trans. Im. Proc.*, vol. 16, no. 10, pp. 2411–22, Oct. 2007.

[10] H. M. Hudson and R. S. Larkin, "Accelerated image reconstruction using ordered subsets of projection data," *IEEE Trans. Med. Imag.*, vol. 13, no. 4, pp. 601–9, Dec. 1994.

[11] J. A. Fessler, E. P. Ficaro, N. H. Clinthorne, and K. Lange, "Grouped-coordinate ascent algorithms for penalized-likelihood transmission image reconstruction," *IEEE Trans. Med. Imag.*, vol. 16, no. 2, pp. 166–75, Apr. 1997.