

Axial block coordinate descent (ABCD) algorithm for X-ray CT image reconstruction

Jeffrey A. Fessler and Donghwan Kim

Abstract—The primary drawback of statistical image reconstruction methods for X-ray CT is the computation time required by iterative algorithms. Iterative coordinate descent (ICD) algorithms converge in relatively few iterations but are challenging to parallelize due to their sequential updates. Conjugate gradient (CG) methods and ordered-subsets (OS) algorithms update all pixels simultaneously, which facilitates parallelization, but these algorithms require many more iterations to converge than ICD. This paper proposes a block coordinate descent algorithm for helical and axial cone-beam X-ray CT image reconstruction in which a group of voxels are updated simultaneously. We focus on updating all the voxels within one axial “column” of the 3D image simultaneously, so we refer to this approach as the axial block coordinate descent (ABCD) algorithm. Because this approach updates many voxels simultaneously (e.g., 64 in an axial scan and hundreds in a helical scan), it is reasonably well suited to parallel processing. At the same time, because the voxels within an axial column are relatively weakly coupled, which is why we selected axial blocks, the algorithm converges fairly quickly. In particular, the simultaneous update of one axial column requires inverting a banded matrix which can be done quickly (ABCD-BAND). An alternative version of the algorithm uses a simpler separable quadratic surrogate for the axial block (ABCD-SQS). Preliminary simulation results illustrate that the ABCD algorithms decrease a regularized weighted least-squares cost function much faster than a traditional separable quadratic surrogate (SQS) method that updates all pixels simultaneously. The proposed ABCD algorithms exhibit about the same decrease per iteration as the ICD method, while appearing much more amenable to parallelization.

I. INTRODUCTION

This paper focuses on statistical image reconstruction methods where one reconstructs the N voxels of the unknown 3D image $\mathbf{x} = (x_1, \dots, x_N)$ by minimizing a regularized weighted least-squares (WLS) cost function:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \Psi(\mathbf{x}), \quad \Psi(\mathbf{x}) = \sum_{i=1}^M \frac{w_i}{2} (y_i - [\mathbf{A}\mathbf{x}]_i)^2 + R(\mathbf{x}),$$

where \mathbf{y} denotes the X-ray CT projection data, w_i denotes the statistical weighting associated with the i th ray, for $i = 1, \dots, M$, M is the number of rays, \mathbf{A} is the $M \times N$ system matrix and $R(\mathbf{x})$ is an edge-preserving regularizer that controls noise while attempting to preserve spatial resolution. The forward projection operation is $[\mathbf{A}\mathbf{x}]_i = \sum_{j=1}^N a_{ij}x_j$. Although we focus here on a WLS data-fit term, the principles generalize readily to other statistical models like the transmission Poisson log-likelihood [1].

EECS Department, University of Michigan, Ann Arbor, MI, USA. Email: fessler@umich.edu, kimdongh@umich.edu. Supported in part by NIH grant R01-HL-098686.

For nonquadratic regularization methods finding the minimizer $\hat{\mathbf{x}}$ requires an iterative algorithm. The primary drawback of statistical image reconstruction methods for X-ray CT is the computation time of such algorithms, particularly the forward and back-projection operations. Iterative coordinate descent (ICD) algorithms [2], [3] converge in relatively few iterations but are challenging to parallelize due to their sequential updates. Conjugate gradient (CG) methods [4] and ordered-subsets (OS) algorithms based on separable quadratic surrogates (SQS) [5], [6] update all pixels simultaneously, which facilitates parallelization, but these algorithms require many more iterations to converge than ICD [7].

Considering that modern computing architectures are based on parallel processing, while also considering the convergence rate properties of ICD and CG/OS, it seems unlikely that the “optimal” practical algorithm will be at either the extreme of updating only one voxel at a time or at the other extreme of updating all voxels simultaneously. Instead, it is plausible that the best compromise will update a group of voxels (but not all voxels) simultaneously, and then cycle through all groups in some order. (It is likely that the choice of order will be important for achieving the fastest possible convergence [8] but we do not investigate that here.) Such “grouped” or “block” coordinate descent algorithms have been used in statistical estimation for over two decades [9], [10] and have also been applied to tomographic image reconstruction [11]–[14].

In [13], a group of pixels (in 2D) was selected (based on checker-board patterns) for simultaneous update. Then optimization transfer principles were applied to develop a separable surrogate function for that group of pixels, and then all pixels in that group were updated simultaneously by minimizing the surrogate function. The pixels within a transaxial slice are relatively strongly coupled, so the surrogate functions have undesirably high curvatures, so the accelerations provided in 2D by the methods in [13] were somewhat modest. The pixels within a group were selected to be distant from each other, rather than adjacent neighbors, to try to minimize the coupling-related curvatures, but still there was coupling that prevented dramatic acceleration. Benson *et al.* [14] took a different approach that did not use optimization transfer, thereby avoiding these high curvatures. They used $k \times k$ blocks of neighboring pixels (in the x – y plane that are highly coupled) and updated all k^2 pixels within that block simultaneously by inverting a $k^2 \times k^2$ matrix. They then sequentially stepped along the axial direction, updating the same $k \times k$ block for each slice. This approach reduced the number of iterations but may increase the work per iteration substantially because the $k^2 \times k^2$ matrices are dense.

In this paper we propose a block coordinate descent algorithm for 3D helical and axial cone-beam X-ray CT image reconstruction that is adapted specifically to the geometry of standard CT scanners (both multi-slice and flat panel) in which one 2D detector axis is aligned with the rotation axis. (The proposed approach may not be ideal for C-arm systems with arbitrary source trajectories.) We focus on updating all the voxels within one axial “column” of the image volume simultaneously, so we refer to this approach as the axial block coordinate descent (ABCD) algorithm. Because this approach simultaneously updates many voxels (*e.g.*, 64 in an axial scan and hundreds in a helical scan), it is reasonably well suited to parallel processing. At the same time, because the voxels within an axial column are relatively weakly coupled, which is why we selected axial blocks, the algorithm converges fairly quickly. In particular, the simultaneous update of one axial column requires inverting a banded matrix, which can be done quickly using a well-known simple algorithm that is linear in the number of unknowns [15]. Unlike previous block-based algorithms for CT reconstruction that are applicable to an arbitrary system matrix \mathbf{A} , the ABCD method is designed to work hand-in-hand with the separable footprint (SF) projector [16]. Although ABCD may be adapted to other forward projection methods, the axial/transaxial separability of the SF projector facilitates the implementation of ABCD in a way that is very efficient and amenable to parallel processing. Preliminary simulation results illustrate that the ABCD algorithm converges at a rate per iteration comparable to ICD, and both converge much faster than conventional optimization transfer methods based on separable quadratic surrogates [6], [17].

II. THEORY

We first review the general framework for block coordinate descent approaches, of which ICD and ABCD are special cases. Partition the parameter vector \mathbf{x} into K sets: $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_K)$, where $1 \leq K \leq N$. (In general the sets do not have to be disjoint but it simplifies explanation and implementation to focus on disjoint sets.) The idea is to update each block of voxels \mathbf{x}_k in turn, *using the most recent estimates of all other blocks*. Specifically, within the n th iteration, we update the k th block by minimizing the cost function $\Psi(\mathbf{x})$ with respect to \mathbf{x}_k as follows:

$$\mathbf{x}_k^{(n+1)} = \arg \min_{\mathbf{x}_k} \Psi_k^{(n)}(\mathbf{x}_k)$$

$$\Psi_k^{(n)}(\mathbf{x}_k) \triangleq \Psi(\mathbf{x}_1^{(n+1)}, \dots, \mathbf{x}_{k-1}^{(n+1)}, \mathbf{x}_k, \mathbf{x}_{k+1}^{(n)}, \dots, \mathbf{x}_K^{(n)}).$$

If each block consists of a single pixel, then this approach reduces to the standard ICD algorithm [2], [3].

When nonquadratic regularization is used, there is no closed-form solution for the minimizer $\mathbf{x}_k^{(n+1)}$, so the block minimization step (arg min above) would itself require an inner iteration. This may be undesirable, so an alternative is to apply optimization transfer to derive a surrogate function $\phi_k^{(n)}$ for $\Psi_k^{(n)}$ and minimize the surrogate instead:

$$\mathbf{x}_k^{(n+1)} = \arg \min_{\mathbf{x}_k} \phi_k^{(n)}(\mathbf{x}_k).$$

Provided the surrogate function satisfies the usual majorization conditions of optimization transfer [1], this type of algorithm is guaranteed to decrease the original cost function $\Psi(\mathbf{x})$ monotonically every update, which in turn ensures convergence in the usual case that mild regularity conditions hold [18].

In [13], *separable* quadratic surrogate (SQS) functions were used of the form

$$\begin{aligned} \phi_k^{(n)}(\mathbf{x}_k) &= \Psi_k^{(n)}(\mathbf{x}_k^{(n)}) + \nabla_{\mathbf{x}_k} \Psi_k^{(n)}(\mathbf{x}_k^{(n)}) (\mathbf{x}_k - \mathbf{x}_k^{(n)}) \\ &\quad + \frac{1}{2} (\mathbf{x}_k - \mathbf{x}_k^{(n)})' \mathbf{D}_k^{(n)} (\mathbf{x}_k - \mathbf{x}_k^{(n)}), \end{aligned}$$

where $\mathbf{D}_k^{(n)}$ is a diagonal “curvature” matrix. This SQS approach leads to the following simple update equation:

$$\mathbf{x}_k^{(n+1)} = \mathbf{x}_k^{(n)} - \left[\mathbf{D}_k^{(n)} \right]^{-1} \nabla_{\mathbf{x}_k} \Psi_k^{(n)}(\mathbf{x}_k^{(n)}), \quad (1)$$

where inversion of \mathbf{D} is trivial because it is diagonal due to the separability $\phi_k^{(n)}$. In fact, in the extreme case where all voxels are in a single group ($K = 1$), this iteration is simply the conventional SQS approach [6] that is known to converge very slowly. When “too many” strongly coupled voxels are in the same group, the curvatures (elements of \mathbf{D}) become large, leading to small values for its inverse and hence small step sizes and slow convergence.

The approach of Benson *et al.* [14] used the *quadratic* surrogate approach of Yu *et al.* [19], which has its roots in Huber’s algorithm [20, p. 184], but they did not use a *separable* surrogate function. This leads to an algorithm of the form (1) except \mathbf{D} is replaced by a *non*-diagonal Hessian matrix $\mathbf{H}_k^{(n)} \triangleq \nabla^2 \phi_k^{(n)}$. For the transaxial blocks chosen in [14], this Hessian matrix $\mathbf{H}_k^{(n)}$ is dense (all elements nonzero in general), thus requiring more effort to invert.

The key idea proposed in this paper is to use *axial* groups of voxels instead of the transaxial blocks that were considered in previous work [13], [14]. The rationale for this choice is that there is far less “coupling” between neighboring voxels in the axial direction than between neighboring voxels in the same slice in the “cylindrical” geometries of helical and axial CT scans. (We define the “coupling” between voxel j and voxel j' as $\sum_{i=1}^M w_i a_{ij} a_{ij'}$, which, for $j \neq j'$, essentially is one of the off-diagonal elements of the Hessian matrix $\mathbf{H}_k^{(n)}$.)

In particular, for a typical multi-slice CT geometry where the slice spacing is matched to the detector row spacing (scaled by the system magnification at isocenter), we have shown that, for the SF projector [16] and also for the distance-driven (DD) projector [21], the Hessian matrix is pentadiagonal for all axial blocks within a standard 70 cm field of view. With this choice of blocks, the resulting axial block coordinate descent (ABCD) algorithm has the form

$$\mathbf{x}_k^{(n+1)} = \mathbf{x}_k^{(n)} - \left[\mathbf{T}_k^{(n)} \right]^{-1} \nabla_{\mathbf{x}_k} \Psi_k^{(n)}(\mathbf{x}_k^{(n)}), \quad (2)$$

where $\mathbf{T}_k^{(n)}$ is the banded Hessian of the (nonseparable) quadratic surrogate function $\phi_k^{(n)}$. For a reconstruction problem with N_z slices, $\mathbf{T}_k^{(n)}$ is $N_z \times N_z$ matrix, and one can solve a system of equations $\mathbf{T}_k^{(n)} \mathbf{u} = \mathbf{v}$ in $O(N_z)$ operations easily [15]. We refer to (2) as the ABCD-BAND algorithm because it uses the “inverse” of a banded matrix. Fig. 1 shows part of

a typical Hessian for a helical CT problem with $N_z = 176$ slices, for the unweighted case $\mathbf{W} = \mathbf{I}$. In this case, there are 3 dominant bands and the next two bands have negligible values.

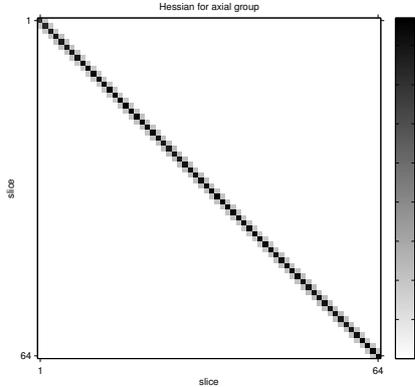


Fig. 1. Part of a banded Hessian matrix for N_z voxels grouped axially.

If one uses an unconventional relationship between the slice thickness and the detector row spacing, or if one uses a more complicated forward projector for which the axial footprint is larger than the models used in the SF and DD projectors, then it is possible that the Hessian matrix of the quadratic surrogate function could have more than 5 diagonals. Nevertheless, the Hessian for an axial group will always be *banded* rather than *dense* for any reasonable system model. In such cases there are several possible approaches that one can take. One option is to use a general approach to inverting banded matrices [15, Ch. 5]; these methods are also $O(N_z)$ but with a larger constant. We refer to these as ABCD-BAND algorithms. Another option is to derive a separable quadratic surrogate (SQS) for the axial group of voxels, leading to an update of the form (1). Because a typical axial group has lower coupling than a typical transaxial group, the elements of the diagonal Hessian $\mathbf{D}_k^{(n)}$ are smaller, leading to faster convergence. We refer to this as the ABCD-SQS algorithm. Yet another option is to derive a quadratic surrogate function with a tridiagonal Hessian for the general banded case. We postpone this more complicated possibility for future work.

The typical edge-preserving regularizers used for 3D CT [3] use the immediately neighboring voxels in all three directions (26 neighbors per voxel). Specifically, $\mathbf{R}(\mathbf{x}) = \sum_{j=1}^N \sum_{k \in \mathcal{N}_j} \psi(x_j - x_k)$ where \mathcal{N}_j denotes the 26 neighbors of the j th voxel and ψ is a potential function such as the Huber function. Because each voxel in an axial “column” is coupled via the regularizer to the voxel immediately above and immediately below, the Hessian matrix for the regularizer is exactly tridiagonal regardless of the voxel size or projection geometry. This is a further benefit of using axial groups and the ABCD-BAND approach.

So far we have focused on the issues associated with designing and inverting the Hessian of the quadratic surrogate function. Equally important to the overall efficiency of the algorithm is how one computes the gradient of the cost function

$\nabla_{\mathbf{x}_k} \Psi_k^{(n)}$ that is required by all of the above methods. Let \mathbf{A}_k denote the sub-matrix of \mathbf{A} with columns corresponding to the group \mathbf{x}_k . Then the data-fit term gradient is simply

$$\nabla_{\mathbf{x}_k} \Psi_k^{(n)} = \mathbf{A}_k' \mathbf{W} \mathbf{r} + \nabla_{\mathbf{x}_k} \mathbf{R}, \quad (3)$$

where $\mathbf{r} = \mathbf{A}\mathbf{x}^{\text{current}} - \mathbf{y}$ is the sinogram residual vector that is always kept as a state vector in ICD and block CD type algorithms [2]. Specifically, we start the algorithm by initializing the residual vector $\mathbf{r} = \mathbf{A}\mathbf{x}^{(0)} - \mathbf{y}$ and then after updating block \mathbf{x}_k using $\mathbf{x}_k^{(n+1)} = \mathbf{x}_k^{(n)} + \mathbf{d}_k^{(n)}$ we update the residual using

$$\mathbf{r} += \mathbf{A}_k \mathbf{d}_k^{(n)}. \quad (4)$$

So this update step requires multiplication by \mathbf{A}_k and the gradient step (3) requires multiplication by its transpose \mathbf{A}_k' . Space constraints prohibit a complete explanation, but the choice of an axial group is particularly well suited to the SF projector [16] because that method uses separable footprints in the transaxial and axial directions. For the back-projection step, inner-products between the transaxial footprint and the projection data can be computed with parallelism across views and across all detector rows. There is also opportunity for SIMD computations across rows. (In contrast ICD can exploit parallelism across views only, because a single voxel affects only a few rows of any given projection view.) After computing these inner products, the effect of the axial footprint is a simple 1D operation along z for each view, which we can also parallelize across views. There is similar opportunity for parallelism across both views and rows for the forward projection step (4).

III. SIMULATION RESULTS

We implemented the ABCD-BAND and ABCD-SQS algorithms in Matlab to compare their convergence rates. Because Matlab is a slow interpreted language, our comparisons were limited to small 3D image sizes. We also implemented the conventional SQS algorithm and the ICD algorithm for comparison. None of these implementations are optimized in terms of run time. Our focus here was on how fast the cost function $\Psi(\mathbf{x})$ decreases each iteration. It is well understood that the computation time per iteration will be quite different for the various algorithms due to their various levels of parallelism. We expect that the compute time per iteration for well-parallelized implementations will obey

$$\text{SQS} < \text{ABCD-SQS} < \text{ABCD-BAND} < \text{ICD} \quad (5)$$

On the other hand, rate of decrease of the cost Ψ per iteration will likely also roughly follow (5), although ABCD-BAND might converge faster per iteration than ICD.

Fig. 2 shows one representative slice of the small 3D images that were used in this preliminary study, along with the images reconstructed by FDK [22], [23] and by the four iterative algorithms listed in (5) after 15 iterations. The SQS algorithm converges very slowly, and has not come close to reaching the minimizer $\hat{\mathbf{x}}$ after 15 iterations. (If run more iterations it will eventually converge.) The ICD and ABCD algorithms reached nearly the same image within 15 iterations. Fig. 3 shows the cost function value $\Psi(\mathbf{x}^{(n)})$ as a function of iteration for the

four algorithms. The undesirably slow convergence of SQS is evident. In this small example, the other three algorithms appeared to reduce the cost function Ψ at nearly the same rate. We also observed similar trends for another case with a $64 \times 64 \times 16$ image volume. Hopefully this behavior will persist when we investigate realistic sized images with a more optimized implementation.

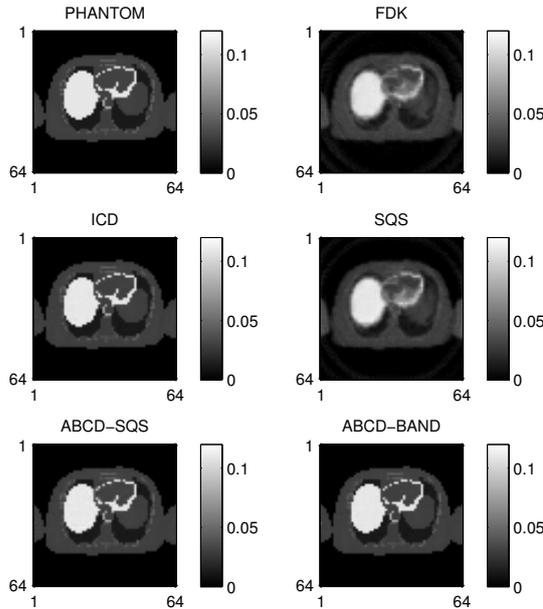


Fig. 2. Reconstructed images after 15 iterations for a small 3D problem.

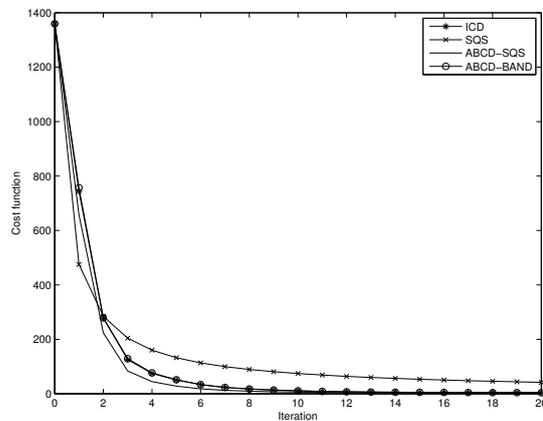


Fig. 3. Cost function $\Psi(\mathbf{x}^{(n)})$ versus iteration n for four algorithms.

IV. SUMMARY

We have proposed ABCD algorithms based on using block coordinate descent with blocks formed from axial “columns” of voxels for helical and axial cone-beam CT. The methods currently are implemented in Matlab to prove the principle of the update. The next step is to implement the method in

C and exploit the abundant parallelism and then compare the convergence rate and compute efficiency with ICD [3] and with SQS-OS [6] algorithms using multi-core systems.

REFERENCES

- [1] J. A. Fessler. Statistical image reconstruction methods for transmission tomography. In M. Sonka and J. Michael Fitzpatrick, editors, *Handbook of Medical Imaging, Volume 2. Medical Image Processing and Analysis*, pages 1–70. SPIE, Bellingham, 2000.
- [2] K. Sauer and C. Bouman. A local update strategy for iterative reconstruction from projections. *IEEE Trans. Sig. Proc.*, 41(2):534–48, February 1993.
- [3] 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.*, 34(11):4526–44, November 2007.
- [4] J. A. Fessler and S. D. Booth. Conjugate-gradient preconditioning methods for shift-variant PET image reconstruction. *IEEE Trans. Im. Proc.*, 8(5):688–99, May 1999.
- [5] K. Kamphuis and F. J. Beekman. Accelerated iterative transmission CT reconstruction using an ordered subsets convex algorithm. *IEEE Trans. Med. Imag.*, 17(6):1001–5, December 1998.
- [6] H. Erdoğan and J. A. Fessler. Ordered subsets algorithms for transmission tomography. *Phys. Med. Biol.*, 44(11):2835–51, November 1999.
- [7] 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.*, volume 5, pages 2708–10, 2005.
- [8] 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.*, 20(1):161–75, January 2011.
- [9] R. J. Hathaway and J. C. Bezdek. Grouped coordinate minimization using Newton’s method for inexact minimization in one vector coordinate. *J. Optim. Theory Appl.*, 71(3):503–16, December 1991.
- [10] S. T. Jensen, S. Johansen, and S. L. Lauritzen. Globally convergent algorithms for maximizing a likelihood function. *Biometrika*, 78(4):867–77, December 1991.
- [11] K. D. Sauer, S. Borman, and C. A. Bouman. Parallel computation of sequential pixel updates in statistical tomographic reconstruction. In *Proc. IEEE Intl. Conf. on Image Processing*, volume 3, pages 93–6, 1995.
- [12] J. A. Fessler, E. P. Ficaro, N. H. Clinthorne, and K. Lange. Fast parallelizable algorithms for transmission image reconstruction. In *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, volume 3, pages 1346–50, 1995.
- [13] 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.*, 16(2):166–75, April 1997.
- [14] T. M. Benson, B. K. B. D. Man, L. Fu, and J.-B. Thibault. Block-based iterative coordinate descent. In *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, 2010.
- [15] G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins Univ. Press, 2 edition, 1989.
- [16] 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.*, 29(11):1839–50, November 2010.
- [17] H. Erdoğan and J. A. Fessler. Monotonic algorithms for transmission tomography. *IEEE Trans. Med. Imag.*, 18(9):801–14, September 1999.
- [18] M. W. Jacobson and J. A. Fessler. An expanded theoretical treatment of iteration-dependent majorize-minimize algorithms. *IEEE Trans. Im. Proc.*, 16(10):2411–22, October 2007.
- [19] Z. Yu, J.-B. Thibault, K. Sauer, C. Bouman, and J. Hsieh. Accelerated line search for coordinate descent optimization. In *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, volume 5, pages 2841–4, 2006.
- [20] P. J. Huber. *Robust statistics*. Wiley, New York, 1981.
- [21] B. De Man and S. Basu. Distance-driven projection and backprojection in three dimensions. *Phys. Med. Biol.*, 49(11):2463–75, June 2004.
- [22] J. A. Fessler. Matlab tomography toolbox, 2004. Available from <http://www.eecs.umich.edu/~fessler>.
- [23] L. A. Feldkamp, L. C. Davis, and J. W. Kress. Practical cone beam algorithm. *J. Opt. Soc. Am. A*, 1(6):612–9, June 1984.