Optimized Momentum Steps for Accelerating X-ray CT Ordered Subsets Image Reconstruction

Donghwan Kim and Jeffrey A. Fessler

***Abstract*—Recently, we accelerated ordered subsets (OS) methods for low-dose X-ray CT image reconstruction using momentum techniques, particularly focusing on Nesterov's momentum method. This paper develops an "optimized" momentum method that is faster than Nesterov's method. Drori and Teboulle's original version requires substantial memory space and computation time per iteration. Therefore, we design an efficient implementation approach of the optimized momentum method that uses storage and computation comparable to Nesterov's method. We also propose to combine it with OS methods. We examine the acceleration of the proposed algorithm using 2D X-ray CT simulation data.**

I. Introduction

We consider low-dose X-ray CT image reconstruction solving the following optimization problem:

$$\hat{x} = \arg\min_x \Psi(x), \tag{1}$$

where a function $\Psi(x)$ belongs to a set $\mathcal{F}_L(\mathbb{R}^{N_p})$ of convex and continuously differentiable functions with L-Lipschitz continuous gradient. Specifically in X-ray CT reconstruction, we use a penalized weighted least squares (PWLS) cost function [1]:

$$\Psi(x) = \frac{1}{2}||y - Ax||_W^2 + R(x), \tag{2}$$

where $x \in \mathbb{R}^{N_p}$ is an unknown image, $y \in \mathbb{R}^{N_d}$ is a noisy measured sinogram data, $A \in \mathbb{R}^{N_d \times N_p}$ is a projection operator [2], a diagonal matrix $W \in \mathbb{R}^{N_d \times N_d}$ provides statistical weighting [3], and $R(x)$ is an edge-preserving regularization function.

In X-ray CT, iteratively minimizing the cost function $\Psi(x)$ requires long computation times due to the computationally expensive operators A and A'. Ordered subsets (OS) methods [4], [5], which use only submatrices of A and A' per iteration, have been used widely for computational efficiency. However, traditional OS methods require many iterations to be used practically, so we recently proposed to combine them with Nesterov's momentum method [6], yielding OS-momentum methods [7] that have faster initial convergence.

Nesterov's momentum method achieves the optimal convergence rate $O(1/n^2)$ where n counts the number of iterations [8]. But, the constant of the convergence rate can be large in Nesterov's method, motivating Drori and Teboulle (hereafter "DT")'s optimized momentum[1] approach [9]. That work constructs a momentum method that achieves the fastest possible convergence. However, each iteration of the optimized momentum method in [9] requires substantial memory space and computational cost for storing and (weighted-)summing all previous gradients. Here we propose a practical approach to circumvent this burden.

Section II and III review Nesterov's momentum method [6] and DT's optimized momentum method [9]. Section IV discusses the computational burden of the optimized momentum method and provides a much more practical approach. We combine this proposed computationally-efficient optimized momentum method with OS methods, and examine the acceleration using 2D CT simulation data, compared to OS methods with Nesterov's method.

II. Nesterov's momentum method

Table I summarizes Nesterov's momentum method [6], which reduces to a gradient descent (GD) method when $t^{(n)} = 1$ for all $n \geq 0$. The difference between $z^{(n+1)}$ and $z^{(n)}$ plays the role of momentum with carefully chosen coefficient $t^{(n)}$, where $(t^{(n)} - 1)/t^{(n+1)}$ increases from 0 to 1 as the algorithm iterates. This algorithm requires only one extra image storage and minimal additional computation in line 5 of Table I compared to GD, while significantly accelerating convergence.

```
1: Initialize $x^{(0)} = z^{(0)}$ and $t^{(0)} = 1$.
2: for $n = 0, 1, \cdots, N-1$
3:     $t^{(n+1)} = \frac{1}{2}\left(1 + \sqrt{1 + 4\left[t^{(n)}\right]^2}\right)$
4:     $z^{(n+1)} = x^{(n)} - \frac{1}{L}\nabla\Psi(x^{(n)})$
5:     $x^{(n+1)} = z^{(n+1)} + \frac{t^{(n)}-1}{t^{(n+1)}}\left(z^{(n+1)} - z^{(n)}\right)$
```

TABLE I
Nesterov's momentum method [6]

Nesterov's method in Table I satisfies the following convergence rate inequality[2] at any nth iteration [6]:

$$\Psi(z^{(n)}) - \Psi(\hat{x}) \leq \frac{2L||x^{(0)} - \hat{x}||^2}{(n+1)^2} \tag{3}$$

for all functions $\Psi(x)$ in $\mathcal{F}_L(\mathbb{R}^{N_p})$. The right term of (3) is the worst-case bound of Nesterov's momentum method [6]; Section III reviews the optimized momentum method that achieves the lowest worst-case bound.

In [7], we combined Nesterov's method in Table I with OS methods [4], [5] for X-ray CT reconstruction (2) by replacing

D. Kim and J. A. Fessler are with the Dept. of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mail:kimdongh@umich.edu, fessler@umich.edu).

Supported in part by NIH grant R01-HL-098686 and equipment donations from Intel Corporation.

[1] Momentum methods here refer to iterative algorithms that have access to only the first-order information of the cost function such as the value and the gradient of the objective as well as the Lipschitz constant L.

[2] DT [9] *numerically* showed that the sequence $\{x^{(n)}\}$ in Table I satisfies the inequality (3) of $\{z^{(n)}\}$ for many choices of n.

$\nabla\Psi(x)$ by the following approximation:

$$\nabla\Psi(x) \approx MA'_m W_m (A_m x - y_m) + \nabla R(x) \tag{4}$$

for $m = 1, \cdots, M$, where A_m, W_m and y_m are submatrices of A, W and y corresponding to mth subset of projection views out of total M subsets, yielding M-times reduced computational cost per update. So, we count one iteration after we visit M subsets considering the use of A and A' per update. Combining OS and Nesterov's momentum provided fast M^2-times initial acceleration [7], unlike M-times acceleration from conventional OS methods.

III. Optimized momentum methods

A. Achievable convergence rate of momentum methods

Nesterov's method [6] in Table I achieves the optimal rate $O(1/n^2)$, since Nesterov [8] found one function in $\mathcal{F}_L(\mathbb{R}^{N_p})$ that cannot be minimized faster than $O(1/n^2)$ by all momentum methods using only the gradient information and the Lipschitz constant L [8]. In particular, any momentum method generating $\{x^{(n)}\}$ satisfies the following lower bound:

$$\frac{3L||x^{(0)} - \hat{x}||^2}{32(n+1)^2} \le \Psi(x^{(n)}) - \Psi(\hat{x}) \tag{5}$$

for at least one function $\Psi(x)$ in $\mathcal{F}_L(\mathbb{R}^{N_p})$. The constant in (3) is $\frac{64}{3}$-times larger than that in (5), showing potential room for improving Nesterov's method in Table I.

B. Optimized momentum in N-iterations (OptMom-N)

DT [9] proposed an optimized momentum method that minimizes the upper-bound of $\Psi(x^{(N)}) - \Psi(\hat{x})$ for a given total number of iterations N among all possible momentum methods, achieving a lower bound with a constant smaller than 2 in (3) (but larger than $3/32$ in (5)). Our work was inspired by [9].

All momentum algorithms using a Lipschitz constant L can be written in the following general form [9]:

$$x^{(n+1)} = x^{(n)} - \frac{1}{L}\sum_{k=0}^{n} h_k^{(n)} \nabla\Psi(x^{(k)}) \tag{6}$$

for $n = 0, \cdots, N-1$, where each update is a weighted sum of all previous gradients with (precomputed) coefficients $\{h_k^{(n)}\}$. A constant-step GD has the form (6) with $h_k^{(n)} = 1$ for $k = n$, and 0 otherwise. Nesterov's method in Table I has this form (6) with the following coefficients [9]:

$$\bar{h}_k^{(n)} = \begin{cases} \frac{t^{(n)}-1}{t^{(n+1)}} \bar{h}_k^{(n-1)}, & 0 \le k \le n-2 \\ \frac{t^{(n)}-1}{t^{(n+1)}} (\bar{h}_{n-1}^{(n-1)} - 1), & k = n-1 \\ 1 + \frac{t^{(n)}-1}{t^{(n+1)}}, & k = n \end{cases} \tag{7}$$

for $n = 0, \cdots, N-1$ and $t^{(n)}$ in Table I. These coefficients $\{\bar{h}_k^{(n)}\}$ are independent of N, and Table II shows a few of them. The analysis using (6) and (7) means that both Table I and the algorithm (6) with $\{\bar{h}_k^{(n)}\}$ in (7) will generate the same sequence of images. However, using (7) in (6) would require storing all previous gradients and (weighted)-summing all of them at each update, whereas Table I uses a computationally efficient recursion.

Coefficients $\{\bar{h}_k^{(n)}\}$ for Nesterov's momentum method [6]					
n \ k	0	1	2	3	4
0	1.0000				
1	0.0000	1.2818			
2	0.0000	0.1223	1.4340		
3	0.0000	0.0649	0.2305	1.5311	
4	0.0000	0.0389	0.1380	0.3180	1.5988

Coefficients $\{\hat{h}_k^{(n)}\}$ of DT's momentum method [9] for $N = 5$					
n \ k	0	1	2	3	4
0	1.6180				
1	0.1741	2.0194			
2	0.0756	0.4425	2.2317		
3	0.0401	0.2350	0.6541	2.3656	
4	0.0178	0.1040	0.2894	0.6043	2.0778

TABLE II
Coefficients of Nesterov's $\{\bar{h}_k^{(n)}\}$ (7) and DT's $\{\hat{h}_k^{(n)}\}$ (9) momentum methods.

DT [9] consider measuring the worst-case bound for a given number of iterations N, a given upper bound B of the distance between $x^{(0)}$ and $\hat{x}$, and a given candidate set of coefficients $\{h_k^{(n)}\}$:

$$\begin{aligned} &P_{N,B}(\{h_k^{(n)}\}) \triangleq \max_{\Psi(x)\in\mathcal{F}_L(\mathbb{R}^{N_p})} \left\{\Psi(x^{(N)}) - \Psi(\hat{x})\right\} \qquad (8) \\ &\text{s.t. } x^{(n+1)} = x^{(n)} - \frac{1}{L}\sum_{k=0}^{n} h_k^{(n)} \nabla\Psi(x^{(k)}), \quad n = 0, \cdots, N-1, \\ &\quad ||x^{(0)} - \hat{x}|| \le B. \end{aligned}$$

Since this problem (8) is intractable due to the functional constraint $\Psi(x) \in \mathcal{F}_L(\mathbb{R}^{N_p})$, DT relax (8) by replacing the functional constraint on $\Psi(x)$ by a basic property of the $\mathcal{F}_L(\mathbb{R}^{N_p})$ functions [8, Theorem 2.1.5]:

$$\frac{1}{2L}||\nabla\Psi(x) - \nabla\Psi(z)||^2 \le \Psi(x) - \Psi(z) - \nabla\Psi(z)'(x - z)$$

for all $x, z \in \mathbb{R}^{N_p}$. Even then, the problem needs several mathematical tricks to finally be transformed to a solvable semidefinite programming (SDP) problem.[3]

DT [9] use (8) to find the "optimized" coefficients $\{\hat{h}_k^{(n)}\}$ that minimize the worst-case bound for a given N as:

$$\{\hat{h}_k^{(n)}\} = \arg\min_{\{h_k^{(n)}\}} P_{N,B}(\{h_k^{(n)}\}), \tag{9}$$

and similarly, the problem (9) eventually becomes an SDP problem in [9]. Here, a solution $\{\hat{h}_k^{(n)}\}$ of (9) is independent of B [9]. An update (6) using the optimized coefficients $\{\hat{h}_k^{(n)}\}$ computed from (9) for a given N becomes an optimized momentum method in N-iterations (OptMom-N) [9].

For example, Table II shows the optimized coefficients $\{\hat{h}_k^{(n)}\}$ for $N = 5$ computed from (9), achieving the following inequality at the final $N = 5$th iteration:

$$\Psi(x^{(5)}) - \Psi(\hat{x}) \le 0.67\frac{L||x^{(0)} - \hat{x}||^2}{(5+1)^2}. \tag{10}$$

The constant here is less than half of that of Nesterov's method in (3) for $n = 5$. This (more than twice) acceleration has been confirmed for multiple choices of N in [9].

[3] We used CVX [10] to solve SDP programs in our experiments.

Similar to combining Nesterov's momentum with OS methods [7], here we consider combining DT's OptMom-N framework with OS methods to achieve faster convergence than OS methods with Nesterov's momentum. However, the substantial computational cost and storage requirements remain large in (6) in general. The next section describes a practical approach to reducing this burden while maintaining fast convergence rate.

IV. Proposed efficient implementation of optimized momentum methods in N-iterations

The general momentum methods in (6) require storing all previous gradients and (weighted-)summing them at each update. In contrast, Table I provides a clever method that uses minimal extra memory and is computationally efficient, implicitly using the coefficients in (7). In this paper, we propose an efficient version of DT's OptMom-N framework [9] in terms of memory and computation, instead of using the general recursion (6), by constraining the coefficients $\{h_k^{(n)}\}$ so that the implementation is efficient while preserving the fast convergence rate.

To transform the general momentum method (6) into a computationally efficient algorithm, we consider two modifications of (6). Firstly, we constrain the method to store at most $n_{\text{w}}+1$ linear combinations of gradient vectors in $\{G_0, \cdots, G_{n_{\text{w}}}\}$, so that the extra memory relative to GD is a fixed amount instead of growing with each iteration. This restriction is essential for a method to be practical in 3D CT. Secondly, we constrain the coefficients $\{h_k^{(n)}\}$ to satisfy the following condition:

$$h_{k-1}^{(n)} = \beta_k h_k^{(n)}, \tag{11}$$

for all $1 \leq k \leq n - n_{\text{w}}$ and $0 \leq n < N$, where $\{\beta_k\}$ is a set of multiplicative factors that we will optimize. The condition (11) enables the method to update recursively a weighted-sum of a part of previous gradients $\{\nabla\Psi(x^{(0)}), \cdots, \nabla\Psi(x^{(n-n_{\text{w}})})\}$ in one image memory space G_0 at the $n(\geq n_{\text{w}})$th iteration as:

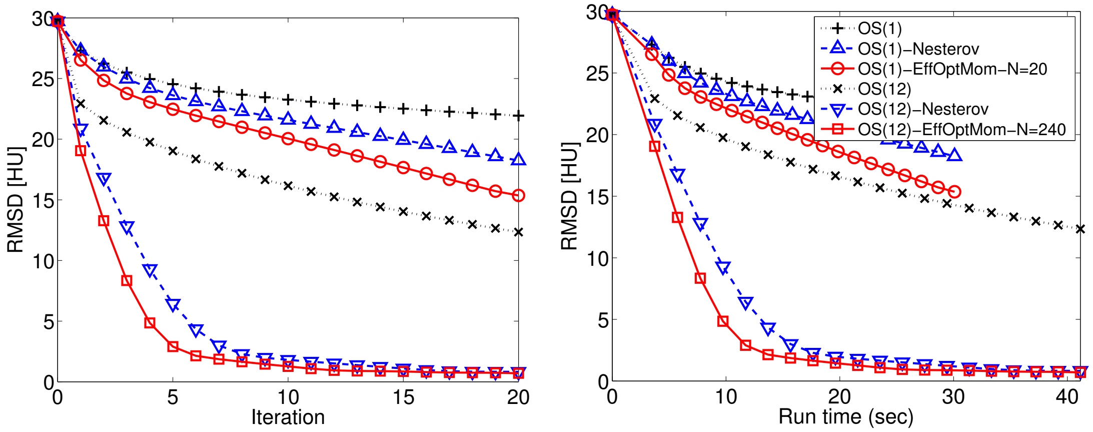
$$G_0^{(n)} \triangleq \sum_{k=0}^{n-n_{\text{w}}} \frac{h_k^{(n)}}{h_{n-n_{\text{w}}}^{(n)}} \nabla\Psi(x^{(k)}) = \sum_{k=0}^{n-n_{\text{w}}} \left(\prod_{l=k+1}^{n-n_{\text{w}}} \beta_l \right) \nabla\Psi(x^{(k)})$$
$$= \beta_{n-n_{\text{w}}} G_0^{(n-1)} + \nabla\Psi(x^{(n-n_{\text{w}})}). \tag{12}$$

We use the remaining memory space $\{G_1, \cdots, G_{n_{\text{w}}}\}$ for storing the n_{w} most recent gradients $\{\nabla\Psi(x^{(n-n_{\text{w}}+1)}), \cdots, \nabla\Psi(x^{(n)})\}$ separately. Table III describes the corresponding efficient implementation of (6) for coefficients $\{h_k^{(n)}\}$ that satisfy the constraint (11).

```
Initialize $x^{(0)}$, $N$, $n_{\text{w}}$, and $G_l = 0$ for $l = 0, \cdots, n_{\text{w}}$.
Choose $\{\beta_k\}_{k=1}^{N-n_{\text{w}}}$ and $\{\{h_l^{(n)}\}_{l=n-n_{\text{w}}}^{n}\}_{n=0}^{N-1}$.
for $n = 0, 1, \cdots, N-1$
    if $n \leq n_{\text{w}} - 1$
        $G_{n+1} \leftarrow \nabla\Psi(x^{(n)})$
    else
        $G_0 \leftarrow \beta_{n-n_{\text{w}}} G_0 + G_1$
        $G_l \leftarrow G_{l+1}$ for $l = 1, \cdots, n_{\text{w}} - 1$
        $G_{n_{\text{w}}} \leftarrow \nabla\Psi(x^{(n)})$
    endif
    $x^{(n+1)} = x^{(n)} - \frac{1}{L}\left( \sum_{l=1}^{n_{\text{w}}} h_{n-n_{\text{w}}+l}^{(n)} G_l + h_{n-n_{\text{w}}}^{(n)} G_0 \right)$
```

TABLE III
Proposed efficient implementation of optimized momentum methods in N-iterations.

To optimize the factors $\{\beta_k\}$ in (11) and Table III, we insert the condition (11) in (9) and solve a modified SDP problem. Alternatively, as a simpler approach, we can project the optimized coefficients computed from (9) onto the subspace of coefficients satisfying (11). Interestingly, we found empirically that the optimized coefficients $\{\hat{h}_k^{(n)}\}$ computed from (9) satisfy the condition (11) for any[4] $n_{\text{w}} \geq 1$. Thus, we chose the smallest $n_{\text{w}} = 1$, which requires same memory space and computational cost as Nesterov's method in Table I. Finally, the momentum method in Table III with $n_{\text{w}} = 1$, $\{\beta_k \triangleq \hat{h}_{k-1}^{(N-1)} / \hat{h}_k^{(N-1)}\}_{k=1}^{N-n_{\text{w}}}$ and $\{\{\hat{h}_l^{(n)}\}_{l=n-n_{\text{w}}}^{n}\}_{n=0}^{N-1}$ using $\{\hat{h}_k^{(n)}\}$ in (9) becomes our proposed efficient implementation of an optimized momentum in N-iterations (EffOptMom-N).

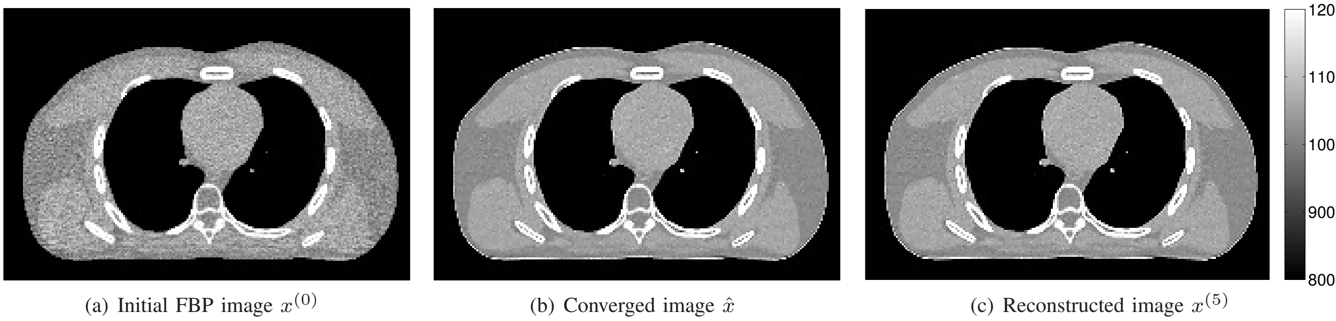


(a) vs. Iteration

(b) vs. Run time

Fig. 1. Plots of RMSD [HU] versus (a) iteration and (b) run time (sec) for OS methods using 1 and 12 subsets with and without momentum techniques. Each iteration of OS methods with 12 subsets performs 12 sub-iterations.

[4] We recently found an analytical solution for $\{\hat{h}_k^{(n)}\}$ of (9) that we will submit to arXiv in near future.



(a) Initial FBP image $x^{(0)}$ (b) Converged image $\hat{x}$ (c) Reconstructed image $x^{(5)}$

Fig. 2. 2D XCAT simulation: (a) an initial FBP image $x^{(0)}$, (b) a converged image $\hat{x}$, and (c) a reconstructed image $x^{(5)}$ from 5 iterations of the proposed OS(12)-EffOptMom-$N = 240$ algorithm using 12 subsets.

For further acceleration, we combine the efficient version of the optimized momentum method in Table III with OS methods, by replacing $\nabla\Psi(x)$ with (4). We expect this OS-EffOptMom-N method to converge faster than OS methods with Nesterov's momentum method. We also replaced the $1/L$ factor in Table III with a diagonal matrix D^{-1} based on separable quadratic surrogates [5], [11]; this D is easier to compute than the (smallest) Lipschitz constant L.

V. Results

We simulated 2D fan-beam CT 492×444 noisy sinogram data from a 512×512 XCAT phantom image [12]. We reconstructed a 256×256 image from the sinogram using OS methods (1 and 12 subsets) with and without momentum techniques for 20 iterations.

Fig. 1 illustrates the root mean square difference (RMSD) between $x^{(n)}$ and the converged image $\hat{x}$ in Hounsfield Units (HU):

$$\text{RMSD}^{(n)} = \frac{||x^{(n)} - \hat{x}||}{\sqrt{N_p}} \text{ [HU]} \tag{13}$$

versus both iteration and run time, to evaluate the convergence rate. The results show that two momentum techniques provide acceleration. Particularly, the proposed EffOptMom-$N = 20$ algorithm reaches the converged image faster than Nesterov's method in both iteration and run time, as expected. Even though the (Eff)OptMom-N algorithm is known to achieve the fast convergence only at the final Nth iteration, the algorithm shows acceleration within all N iterations in this experiment.

In Fig. 1, using 12 subsets in OS methods accelerated all algorithms, even though it slightly increased the computation time per iteration for executing 12 sub-iterations per each iteration. The EffOptMom-N algorithm with OS(12) method for 20 iterations requires $N = 240$ sub-iterations, leading to solving a large SDP problem (9) with $N = 240$ to compute the optimized coefficients $\{\hat{h}_k^{(n)}\}$. However, these coefficients can be precomputed for a given N regardless of the data set, so we can neglect the computation of SDP problem in practice (and in Fig. 1). Considering a large $N = 240$, we note that an (inefficient) OptMom-$N = 240$ framework would require 240 image space, while our proposed efficient implementation uses only one extra image space for storing a linear combination of previous gradients.

Fig. 2 shows an initial filtered back-projection (FBP) image $x^{(0)}$, a converged image $\hat{x}$, and a reconstructed image from 5 iterations of the proposed EffOptMom-N algorithm with OS(12) method. The result indicates that we can reach nearby the converged image within very few iterations using the proposed algorithm.

VI. Conclusion

We proposed an efficient implementation of optimized momentum [9] in N-iterations for X-ray CT image reconstruction and showed that it converges faster in both N-iterations and run time than Nesterov's method. We combined it with OS methods for further acceleration, leading to faster convergence than our previous combination of OS methods and Nesterov's momentum method [7]. We will next investigate this acceleration in real 3D CT data.

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] 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.

[3] 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.

[4] 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.

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

[6] Y. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$," *Dokl. Akad. Nauk. USSR*, vol. 269, no. 3, pp. 543–7, 1983.

[7] D. Kim, S. Ramani, and J. A. Fessler, "Ordered subsets with momentum for accelerated X-ray CT image reconstruction," in *Proc. IEEE Conf. Acoust. Speech Sig. Proc.*, 2013, pp. 920–3.

[8] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*, Kluwer, 2004.

[9] Y. Drori and M. Teboulle, "Performance of first-order methods for smooth convex minimization: A novel approach," *Mathematical Programming*, 2013.

[10] M. Grant, S. Boyd, and Y. Ye, "Disciplined convex programming," 2006.

[11] D. Kim, D. Pal, J-B. Thibault, and J. A. Fessler, "Accelerating ordered subsets image reconstruction for X-ray CT using spatially non-uniform optimization transfer," *IEEE Trans. Med. Imag.*, vol. 32, no. 11, pp. 1965–78, Nov. 2013.

[12] 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.