Accelerated Line Search for Coordinate Descent Optimization

Zhou Yu, *Student Member, IEEE*, Jean-Baptiste Thibault, Ken Sauer *Member, IEEE*, Charles Bouman *Fellow, IEEE*, and Jiang Hsieh *Senior Member, IEEE*

***Abstract*— Iterative reconstruction (IR) methods show promise for image quality improvements in helical X-ray CT. Coordinate descent methods have good convergence properties for these problems, but include a one-dimensional, nonlinear optimization for each voxel's update. This paper presents two methods for speeding computation by replacing the 1D line search with one-step updates. Both methods greatly reduce this portion of IR computation, at no loss of convergence speed as measured by full iteration count. Experimental examples include both phantom and clinical CT scans.**

***Index Terms*— Computed tomography, iterative reconstruction, coordinate descent, line search.**

I. Introduction

Multi-slice CT scanning is particularly attractive for clinical applications due to short acquisition times, thin slices, and large organ coverage. As an attempt to provide more flexibility in the reconstruction choices, iterative reconstruction (IR) algorithms have been recently introduced for multi-slice helical CT images [1], [2]. Statistical reconstruction methods, such as the maximum *a posteriori* (MAP) estimate offer flexibility in dealing with noise and other inaccuracies in the data.

The MAP estimate can be computed using a variety of optimization methods. In particular, we have found that the iterative coordinate descent (ICD) method of optimization is flexible and has relatively rapid convergence [3]. The key step in the implementation of ICD is the update of a pixel x_j. Each pixel is updated to minimize the negative log *a posteriori* probability. This results in a basic update equation of the form

$$x_j^{(n+1)} = \arg\min_{x_j \geq 0} \left\{ \theta_1 x_j + \frac{\theta_2 x_j^2}{2} + \sum_{k \in \mathcal{N}_j} w_{jk} \rho(x_j - x_k^{(n)}) \right\} \quad (1)$$

where the parameters θ_1 and θ_2 are computed from the sinogram data, $\mathcal{N}_j$ is the set of indices for neighboring pixels, w_{jk} are fixed weights, and $\rho(\cdot)$ is the so-called potential function which penalizes large deviations in neighboring pixels.

Zhou Yu is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907. Email: yuz@purdue.edu

Jean-Baptiste Thibault is with the Applied Science Laboratory, GE Healthcare Technologies, 3000 N Grandview Bvd, W-1180, Waukesha, WI 53188. Telephone: (262) 312-7404. Email: jean-baptiste.thibault@med.ge.com

Ken Sauer is with the Department of Electrical Engineering, 275 Fitzpatrick, University of Notre Dame, Notre Dame, IN 46556-5637. Telephone: (574) 631-6999. Email: sauer@nd.edu

Charles Bouman is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907-0501. Telephone: (765) 494-0340. Email: bouman@ecn.purdue.edu

Jiang Hsieh is with the Applied Science Laboratory, GE Healthcare Technologies, 3000 N Grandview Bvd, W-1180, Waukesha, WI 53188. Telephone: (262) 312-7635. Email: jiang.hsieh@med.ge.com

The computation of (1) has two basic components. First, the parameters θ_1 and θ_2 must be computed from the sinogram data. Generally speaking, this is the most computationally expensive portion of the update. However, with modern parallel computing architectures, it is potentially possible to speed the evaluation of these two parameters. Once θ_1 and θ_2 are evaluated, the 1D minimization must be computed. Typically, this 1D minimization, or line-search, can be done using half-interval search which is simple and robust. However, half-interval search generally requires N steps to reach a precision of $2^{-N}\epsilon$ where ϵ is the initial uncertainty in the solution. If the desired precision of the line-search is high, then the number of steps, N, may be quite large. In this case, the cost of the half-interval search can substantially increase the total computational cost of an ICD iteration.

In this paper, we introduce two fast update methods for computing an approximate line-search in one step. Both update methods reduce the computation by approximately a factor of 5 as compared to half-interval search. Just as importantly, these update methods do not reduce the convergence speed in terms of the number of full iterations of ICD. The first update method is based on the use of sub-functions that guarantee monotone convergence of the ICD algorithm. The second update method uses linear interpolation of two samples of the first derivative of the function to estimate the 1D root.

II. Accelerated Line Search

Let the vector x be the three-dimensional image, and let y be the tomographic measurements. The objective function here, derived from Poisson statistics, is:

$$\hat{x} = \arg\min_{x \geq 0} \left\{ \frac{1}{2} (y - Ax)^T D (y - Ax) + U(x) \right\} \quad (2)$$

where D is a diagonal matrix that reflects inherent variations in the credibility of data, and the the regularizing term $U(x)$ represents the negative log of the *a priori* density. For this work, we will assume that $U(x)$ is formed by a sum of potential functions $\rho(x_j - x_k)$ where x_j and x_k are neighboring pixels. The potential function applied here has the form:

$$\rho(\Delta) = \frac{|\Delta|^p}{1 + |\Delta/c|^{p-q}} \quad (3)$$

with $p \geq q \geq 1$, and $\Delta = x_j - x_k$. The constants p and q determine the powers for low contrast (i.e. $|\Delta| << c$), and high contrast (i.e. $|\Delta| >> c$) regions respectively. The constant c determines the approximate threshold of transition between these low and high contrast regions. We restrict ourselves to the case $1 \leq q \leq p \leq 2$ since in this case $\rho(\Delta)$ can be proved to be convex. The non-quadratic nature of the objective that results appears important to adapt images to clinician's expectations, but makes optimization more difficult.

Application of the ICD optimization method to (2) results in the 1D minimization of (1). However, since ICD is an iterative algorithm, it is not necessary to find a precise 1D minimum at each pixel update. An approximate 1D minimum may be sufficient to insure global convergence of ICD without slowing convergence speed. In practice, our fast update method should meet two requirements: First, the update method should be non-iterative and be computationally efficient as compared to half-interval search. Second, the update method should not slow down the convergence of the ICD algorithm in terms of number of full iterations.

In section III and IV, we propose two alternatives for simplification of (1) through local quadratic approximations of the one-dimensional function. Section V shows the experimental results of both algorithms compared to half interval search.

III. Functional Substitution Method

The update method, here labeled functional substitution (FS), replaces the function $\rho(\Delta)$ in each term of (3) with a quadratic sub-function, which allows a closed form for the minimizer. Let $x_j^{(n)}$ denote the pixel at location j and iteration n, and $x_0 = x_j^{(n)} - x_k$. Let $[min, max]$ be the interval on which the solution is known to lie, and also assuming $x_j^(n) \in [\min, \max]$. Then the sub-function $f_{j,k}(x)$ for voxel pair (j, k) must have the properties that

$$f_{j,k}(x_0) = \rho(x_0) \quad (4)$$

$$f'_{j,k}(x_0) = \rho'(x_0) \quad (5)$$

$$f_{j,k}(x) \geq \rho(x), \forall x \in [\Delta min, \Delta max]. \quad (6)$$

where $\Delta max = max - x_k$ and $\Delta min = min - x_k$. It can be shown that a sub-function satisfying the above three conditions will result in a monotonically decreasing cost function, which guarantees the convergence of the ICD algorithm [4], [5].

A. Derivation of the sub-function

In the following, use the notation $f_s(x)$ for the sub-function (suppressing the dependency of f on i and j for simplicity). Let the quadratic sub-function $f_s(x) = ax^2 + bx + c$. Our objective is to select the parameters a, b and c so that (4), (5) and (6) hold. To do this, we will introduce a new constraint:

$$f_s(T) = \rho(T) \quad (7)$$

for a carefully selected value of T. Let T be given by:

$$T = \begin{cases} -x_0, & if\, |x_0| = \min\{|x_0|, |\Delta \min|, |\Delta \max|\} \\ \Delta \min, & if\, |\Delta \min| = \min\{|x_0|, |\Delta \min|, |\Delta \max|\} \\ \Delta \max, & if\, |\Delta \max| = \min\{|x_0|, |\Delta \min|, |\Delta \max|\} \end{cases} \quad (8)$$

So T selects the value among $\{-x_0, \Delta min, \Delta max\}$ with the minimum magnitude. The constants a, b and c can then

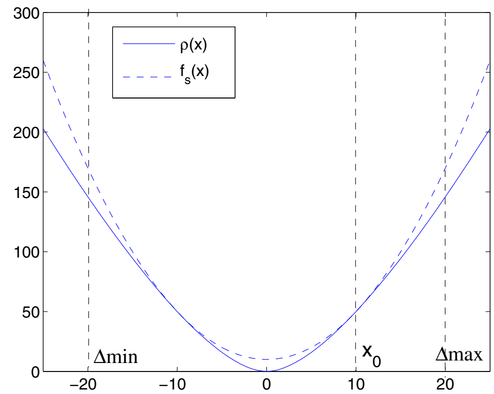

Fig. 1. Functional Substitute, case 1: $T = -x_0$

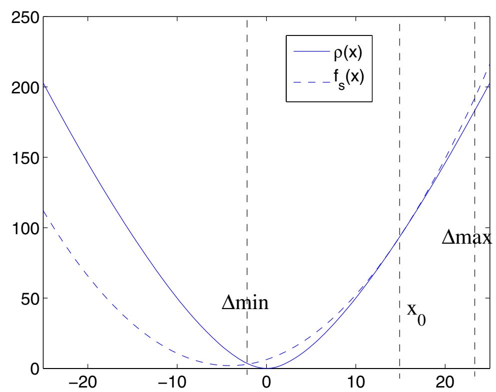

Fig. 2. Functional Substitute, case 2: $x_0 > 0$, $T = \Delta min$

be computed by using this value of T and applying equations (4), (5) and (7) to yield:

$$a = \frac{\rho(T) - \rho(x_0)}{(T - x_0)^2} - \frac{\rho'(x_0)}{T - x_0}$$

$$b = \rho'(x_0) - 2ax_0$$

$$c = \rho(x_0) - ax_0^2 - bx_0$$

The resulting sub-function will upper bound $\rho(x)$ on the interval $[\Delta min, \Delta max]$ as stated in the following theorem:

Theorem 1: Assume $\rho(x) \in C^1$ satisfy the following conditions:

1) $\rho(x)$ is an even function.
2) $\rho'(x)$ is strictly concave for $x > 0$ and strictly convex for $x \leq 0$

and $f_s(x_0)$ is a quadratic function which satisfies (4), (5) and (7), where T is given by (8), then $f_s(x) \geq \rho(x)$ on interval $[\Delta min, \Delta max]$. (see [7] for proof)

As stated in (8), there are three different cases for the choice of T. In the first case, $T = -x_0$, as illustrated in Fig. 1, $f_s(x) \geq \rho(x)$ for $\forall x \in \Re$. Furthermore, in this case, the expression for a and b can be simplified to:

$$a = \frac{\rho'(x_0)}{2x_0}$$

$$b = 0$$

Case 2 is illustrated in Fig. 2, where $T = \Delta min$. It can be shown that $f_s(x) \geq \rho(x)$ for $x \geq T$. Case 3, where $T = \Delta max$, is symmetric to case 2, and $f_s(x) \geq \rho(x)$ for $x \leq T$

B. Line search algorithm

The line search algorithm using the functional substitution method has two steps. The first step to compute sub-function for each term in (1), namely computing the coefficients a_{jk} and b_{jk}. This results in a substitute function $F_s(x)$ for the overal 1D cost function given by:

$$F_s(x_j) = \theta_1 x_j + \frac{\theta_2 x_j^2}{2} + \sum_{k \in \mathcal{N}_j} w_{jk} f_{j,k}(x_j - x_k)$$

The second step is to find the minimizer of the quadratic sub-function $F_s(x)$, i.e. instead of solving (1), we solve

$$x_j^{(n+1)} = \arg\min_{x_j \geq 0} F_s(x) \quad (9)$$

Since each term in equation (9) is quadratic, the minimizer is given by the closed form formula:

$$x_j^{(n+1)} = \frac{2 \sum\limits_{k \in N_j} a_{jk} w_{jk} x_k - \sum\limits_{k \in N_j} w_{jk} b_k + \theta_2 - \theta_1}{2 \sum\limits_{k \in N_j} w_{jk} a_{jk} + \theta_2} \quad (10)$$

Note that the above formula does not require the computation of c_{jk}. Only a_{jk} and b_{jk} need to be computed.

C. Over-relaxed Functional Substitute Method

Over-relaxed versions of coordinate descent methods have been successfully used in a number of numerical problems to improve the convergence speed. In the functional substitution method, the sub-function has to upper bound the original cost function to guarantee the convergence; however this makes the update in the FS method conservative, the over-relaxation method will encourage larger update steps while still guaranteeing that the cost function will monotonically decrease at each step.

Let x_j^* be the update value computed using the functional substitute method, the relaxation method updates the voxel using the following formula:

$$x_j^{(n+1)} = x_j^{(n)} + \alpha(x_j^* - x_j^{(n)})$$

where $0 < \alpha < 2$. If $\alpha \in (0, 1)$, it becomes under-relaxed method, and if $\alpha \in (1, 2)$, it is over-relaxed. If $x_j^{(n+1)}$ falls outside the interval $[\min \max]$, it is truncated to either $\min$ or $\max$.

Since the sub-function is quadratic, it can be shown that $F_s(x_j^{(n)} + 2(x_j^* - x_j^{(n)})) = F_s(x_j^{(n)})$. By convexity of the sub-function, choosing $\alpha \in (0, 2)$ will gaurantee that $F_s(x^{(n+1)}) < F_s(x^{(n)})$, thereby gauranteeing monotone reduction of the 1D cost function.

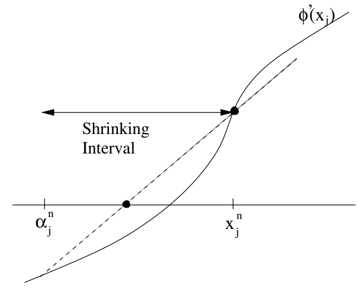

Fig. 3. Linear interpolation of first derivative used for one-step update.

D. Analysis of the functional substitution method

1) Computation Cost: The major computation in each step of FS comes from the parameter estimation of a_{jk}. To compute a_{jk} requires one evaluation of $\rho'(x)$, and at most two evaluations of $\rho(x)$. The complexity of evaluating $\rho'(x)$ is equivalent to two evaluations of $\rho(x)$. Half interval search requires one evaluation of $U'(x)$ for each step. Therefore the substitute function method has at most twice the computation of a single step of the half interval method, while half interval method requires on average around 10 update steps for each pixel. Thus, FS can reduce the computational cost of line search by over 80%.

2) Parallelization of FS algorithm: The computation of each a_{jk} is independent of others; therefore the computation can be distributed onto multiple processors of a parallel machine.

IV. Linear Interpolation of Derivative Method

A second approach to the problem is an attempt to improve the estimate of the second derivative in the region of the solution of the one-dimensional optimization at step n. We denote the total objective of (2) at the n-th step in terms of the single variable $x_j^{(n)}$ as $\phi(x_j)$. We use samples of the first derivative of $\phi(x_j)$ at two points, one being the current state, $x_j^{(n)}$. The sign of the derivative at $x_j^{(n)}$ determines the direction in which the solution must lie. A second point, $\alpha_j^{(n)}$, is then chosen which, it is hoped, forms an interval with $x_j^{(n)}$ which contains the root of the derivative. The substitute function in this method is a quadratic which matches the total first derivative at two points. This technique, which we will call linear interpolation of derivatives (LID), is illustrated in Figure 3.

Unlike the functional substitution method, LID makes a single approximation to $U(x)$ in (2) as a function of $x_j^{(n)}$, rather than considering each component $\rho(x_j^{(n)} - x_k)$ separately. If we use the notation $\tilde{\rho}(x_j)$ (suppressing the dependence of $\tilde{\rho}$ on n and j for simplicity in this one-dimensional exposition) for the summation term in (1), the quadratic substitute for the regularization component of the objective is $\frac{\tilde{a} x_j^2}{2} + \tilde{b} x_j$, with

$$\tilde{a} = \frac{\tilde{\rho}'(x_j^{(n)}) - \tilde{\rho}'(\alpha_j^{(n)})}{x_j^{(n)} - \alpha_j^{(n)}}, \quad (11)$$

$$\tilde{b} = \tilde{\rho}'(x_j^{(n)}) - \tilde{a} x_j^{(n)}. \quad (12)$$

The minimizer of the resulting 1D quadratic then takes the form

$$x_j^{(n+1)} = -\frac{\theta_1 + \tilde{b}}{\theta_2 + \tilde{a}} \quad (13)$$

The substitute function in LID is not a sub-functional, and convergence is not simply guaranteed. Particularly when the selected interval does not include the quadratic's solution, large updates may cause at least temporary increase in the cost functional. Checking and reselection of $\alpha_j^{(n)}$ may provide a remedy, but complicates a process whose simplicity is essential. As a heuristic solution, we augment LID with two features: (1) we constrain the updated value $x_j^{(n+1)}$ to lie between $\alpha_j^{(n)}$ and $x_j^{(n)}$ and (2) we form a schedule for the selection of α_j^n with

$$\begin{aligned} \alpha_j^{(n)} &= x_j^{(n)} - \Delta^{(n)},\ \phi'(x_j) > 0 \\ \alpha_j^{(n)} &= x_j^{(n)} + \Delta^{(n)},\ \phi'(x_j) < 0 \end{aligned}$$

and $\Delta^{(n)}$ decreasing with iteration number to capitalize on the decreasing size of pixel updates.

A. Computational Cost of LID

Similarly to FS, LID requires evaluation of components of the *a priori* portion of the cost at two points, but only a single computation of the log-likelihood parameters θ_1 and θ_2. Because the log-likelihood portion of the computation has by far the greater cost, ICD also reduces computation by a factor of 80-90 percent relative to the half-interval search. Neighboring pixels' contributions to derivatives can be computed in parallel also with LID.

V. RESULTS

We compare convergence behavior of the search algorithm with both one-step approximations in the plots below. Reconstructions was initialized with a filtered backprojection image, which furnishes relatively accurate low frequency information. Such a high quality initial estimate is essential for ICD's performance. FS updates were over-relaxed by factor 1.5, and the interval used for LID had length $\Delta^{(n)} = 300\exp(-0.2n)$ Hounsfield units, where n is iteration number. All data presented here was from helical scans at medium dosage.

Computation time consumed by the one-dimensional optimization falls by approximately 80% using either one-step update. A somewhat surprising result is that the rate of convergence of the objective function can actually be improved somewhat by the approximate methods. In Figure 4, LID is significantly faster than the search algorithm, and while FS converges less rapidly early on, by the 8th iteration it has exceeded the half-interval search and appears to continue faster asymptotic convergence. A second experiment, in Figure 5, shows results on an axial, clinical chest scan, with a form of LID which adapts its interval length according to update history. Here the three methods are very close, with adaptive LID initially faster, but slowing in the intermediate iterations.

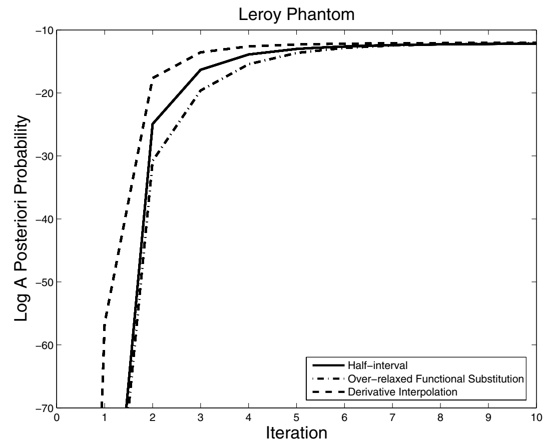

Fig. 4. Chest phantom: convergence of global log *a posteriori* probability density as a function of iteration number.

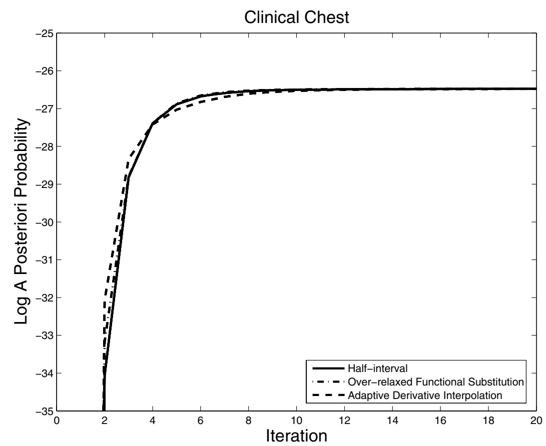

Fig. 5. Clinical chest scan: convergence of global log *a posteriori* probability density as a function of iteration number.

REFERENCES

[1] J.-B. Thibault, K. Sauer, C. Bouman, and J. Hsieh, "Three-dimensional statistical modeling for image quality improvements in mult-slice CT," in *Proc. of the 8th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, Salt Lake City, UT, July 6-9 2005, pp. 271–274.

[2] D. Politte, S. Yan, J. O'Sullivan, D. Snyder, and B. Whiting, "Implementation of alternating minimization algorithms for fully 3D CT imaging," in *Proceedings of the SPIE/IS&T Symposium on Electronic Imaging Science and Technology - Computational Imaging*, vol. 5674, San Jose, CA, January 17-18 2005, pp. 362–373.

[3] C. A. Bouman and K. Sauer, "A unified approach to statistical tomography using coordinate descent optimization," *IEEE Trans. on Image Processing*, vol. 5, no. 3, pp. 480–492, March 1996.

[4] J. Zheng, S. Saquib, K. Sauer, and C. A. Bouman. Parallelizable Bayesian tomography algorithms with rapid, guaranteed convergence. IEEE Trans. on Image Processing, 9(10):1745–1759, October 2000.

[5] J. A. Fessler and A. O. Hero, "Penalized maximumlikelihood image reconstruction using space-alternating generalized EM algorithms," IEEE Transactions on Image Processing, vol. 4, pp. 1417–1429, 1995.

[6] H. Erdogan and J. A. Fessler, "Monotonic Algorithms for Transmission Tomography," IEEE Trans. Med. Imaging, vol. 18, no. 9, pp. 801–814, September 1999.

[7] http://www.ece.purdue.edu/˜bouman/publications/pdf/mic06proof.pdf