Conjugate-Gradient Preconditioning Methods for Shift-Variant PET Image Reconstruction

Jeffrey A. Fessler, *Member, IEEE,* and Scott D. Booth

***Abstract*— Gradient-based iterative methods often converge slowly for tomographic image reconstruction and image restoration problems, but can be accelerated by suitable preconditioners. Diagonal preconditioners offer some improvement in convergence rate, but do not incorporate the structure of the Hessian matrices in imaging problems. Circulant preconditioners can provide remarkable acceleration for inverse problems that are approximately shift-invariant, i.e., for those with approximately block-Toeplitz or block-circulant Hessians. However, in applications with nonuniform noise variance, such as arises from Poisson statistics in emission tomography and in quantum-limited optical imaging, the Hessian of the weighted least-squares objective function is quite shift-variant, and circulant preconditioners perform poorly. Additional shift-variance is caused by edge-preserving regularization methods based on nonquadratic penalty functions. This paper describes new preconditioners that approximate more accurately the Hessian matrices of shift-variant imaging problems. Compared to diagonal or circulant preconditioning, the new preconditioners lead to significantly faster convergence rates for the unconstrained conjugate-gradient (CG) iteration. We also propose a new efficient method for the line-search step required by CG methods. Applications to positron emission tomography (PET) illustrate the method.**

***Index Terms*— Circulant matrix, edge-preserving, image restoration, PET, tomography.**

I. Introduction

TOMOGRAPHIC image reconstruction using statistical methods can provide more accurate system models, statistical models, and physical constraints than the conventional filtered backprojection (FBP) method. Most statistical methods for image reconstruction require minimizing an objective function related to the measurement statistics. For realistic image sizes, direct minimization methods are computationally intractable, so iterative methods are required. For objective functions that are quadratic, or at least convex and locally quadratic, conjugate-gradient (CG) algorithms are appealing for reasons of convergence rate, simplicity, and potential for parallelization [1]–[10]. This paper describes new preconditioning methods that accelerate the convergence of gradient-based iterative methods for penalized weighted least-squares tomographic image reconstruction. The proposed methods generalize those described in [8] and [11]. The methods also apply to other inverse problems such as image restoration.

Gradient-based iterations often converge slowly for poorly-conditioned problems and for large-scale problems where the Newton–Raphson method is impractical due to the size of the Hessian matrix. Broadly speaking, the goal when preconditioning is to induce a coordinate transformation that improves the condition number of a problem, which generally leads to faster convergence. Several "generic" preconditioners for CG methods are described in textbooks, e.g., [12]. The simplest and perhaps most prevalent preconditioners are diagonal scaling matrices [12], versions of which have been applied to image reconstruction[1] [4], [6]–[8], [14]. Although diagonal preconditioners can accelerate convergence in many optimization problems, these preconditioners cannot provide the fastest convergence rate for imaging problems since they ignore the off-diagonal structure of the Hessian of the objective function (the correlation between pixels and the $1/r$ response of tomographic systems).

Some simple (or perhaps over-simplified) imaging problems are nearly shift-invariant, i.e., the matrix[2] $\mathbf{G}'\mathbf{G}$, which is part of the Hessian of the objective function, is approximately block-Toeplitz, where $\mathbf{G}$ is a system matrix described in (1) below. For these problems the diagonal preconditioner is ineffective, but appropriate circulant preconditioners can provide very remarkable improvements in convergence rate. (See [15] for a recent thorough review of this subject.) Circulant preconditioners are particularly appealing since one can use the fast Fourier transform (FFT) for efficient implementation. Several optimal circulant preconditioners are available for Toeplitz problems [15]–[19]. Such circulant preconditioners, also called "Fourier" preconditioners, have been applied to both tomographic image reconstruction [20] and image restoration problems [21], [22]. Fig. 3, described in Section V, illustrates the well-known effectiveness of circulant preconditioners for shift-invariant problems. Circulant preconditioners have also been applied to total variation methods for nonlinear image restoration [23], [24].

Manuscript received February 7, 1997; revised June 19, 1997. This work was supported in part by the National Institutes of Health under Grants CA-60711 and CA-54362, and by the Whitaker Foundation. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. W. Clem Karl.

J. A. Fessler is with the Department of Electrical Engineering and Computer Science and the Department of Biomedical Engineering, University of Michigan, Ann Arbor, MI 48109-2122 USA (e-mail: fessler@umich.edu).

S. D. Booth is with the University of Virginia Medical Center, Charlottesville, VA 22906 USA.

Publisher Item Identifier S 1057-7149(99)03424-7.

[1] Curiously, the most prevalent preconditioner in the emission tomography literature is not the usual diagonal matrix (8) formed from the Hessian, but is rather the particular estimate-dependent diagonal matrix that is implicit in the popular expectation-maximization (EM) algorithm. (For emission tomography the EM algorithm is equivalent to a gradient-ascent iteration with a certain estimate-dependent diagonal preconditioner [13].) Kaufman [4] and Mumcuoglu *et al.* [6] incorporated this diagonal preconditioner into conjugate-gradient algorithms. The estimate-dependence of this preconditioner affects the conjugacy of the direction vectors, leading Lalush *et al.* to advocate an iteration-independent diagonal preconditioner [3], [14].

[2] We use "$'$" to denote matrix and vector Hermitian transpose.

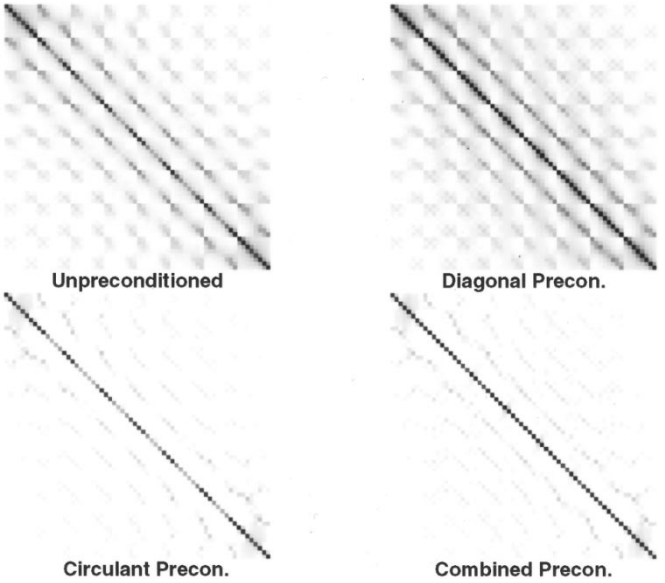


Fig. 1. Images of the preconditioned Hessian $\mathbf{MH}$ for an 8×8-pixel test image for a small tomographic imaging problem. Preconditioners: $\mathbf{I}$, $\mathbf{M}_D$, $\mathbf{M}_C$, $\mathbf{M}_{\mathrm{CDC}}$. The diagonal elements of $\mathbf{M}_C\mathbf{H}$ range from .55 to 1.48, whereas the diagonal elements of $\mathbf{M}_{\mathrm{CDC}}\mathbf{H}$ range from .77 to 1.04. The eigenvalues of $\mathbf{M}_{\mathrm{CDC}}\mathbf{H}$ are also more narrowly spread than for $\mathbf{M}_C\mathbf{H}$, which leads to faster convergence for the proposed preconditioner.

Unfortunately, many imaging problems are *shift variant*, for the following reasons. Firstly, many imaging systems produce heteroscedastic measurements, particularly in quantum-limited applications such as emission tomography and photon-limited optical imaging [25], [26]. In these applications, the measurement noise covariance is a diagonal matrix with very nonuniform entries, due to both nonuniform Poisson variance and to physical effects such as detector efficiency and attenuation [27]. Therefore, the Fisher information term $\mathbf{G}'\mathbf{W}\mathbf{G}$ within the Hessian (see (5) and Fig. 1) is shift-variant and thus poorly approximated by any circulant preconditioner. Secondly, incorporating a nonquadratic edge-preserving penalty function into the objective function to perform regularization introduces an additional form of shift variance into the Hessian matrix (5). Finally, for some imaging systems (e.g., SPECT, 3-D PET, and helical CT), even if we were to disregard the nonuniform noise variance, the matrix $\mathbf{G}'\mathbf{G}$ is still inherently shift-variant due to the system geometry and/or spatial variations in detector response. (See [28] for an image restoration method for shift-variant imaging systems.) Thus, neither diagonal nor circulant preconditioning is well-suited to shift-variant imaging problems. Since statistical methods for image reconstruction yield higher-quality images than FBP reconstruction but at a price of increased computation, it is important to develop methods for accelerating convergence of the iterative algorithms. This paper generalizes the quadratic method described in [8] by developing improved preconditioners that accommodate the shift-variance caused by nonuniform noise and nonquadratic penalties.

Section II reviews the image reconstruction problem and the preconditioned conjugate gradient iteration. Section III summarizes the preconditioners. Section IV describes a new method for the CG line-search step. Section V reports empirical results that demonstrate significantly improved convergence rates in PET reconstruction with real data.

II. Regularized Reconstruction Problem

Most tomographic image reconstruction and image restoration problems are specific cases of the following general inverse problem: find an estimate of object parameters $\underline{x} = [x_1, \ldots, x_p]'$ (e.g., pixel intensities) from a measurement vector $\underline{y}$ related to $\underline{x}$ by

$$\underline{y} = \mathbf{G}\underline{x}_{\mathrm{true}} + \text{noise}. \tag{1}$$

In the context of PET, x_j is the radioisotope concentration in the jth voxel, $\mathbf{G} = \{g_{ij}\}$ is the "system" matrix that describes the tomographic geometry, and $\underline{y}$ represents sinogram data that has been precorrected for the effects of random coincidences, attenuation, scatter, deadtime, etc. However, the proposed methods apply generally to problems of the form (1) for which $\mathbf{G}'\mathbf{G}$ is approximately block Toeplitz.

A. The Objective Function

One useful statistical criterion for estimating $\underline{x}$ from $\underline{y}$ is the following penalized weighted least-squares[3] objective function [5], [27]:

$$\Phi(\underline{x}) = \frac{1}{2}(\underline{y} - \mathbf{G}\underline{x})'\mathbf{W}(\underline{y} - \mathbf{G}\underline{x}) + \beta R(\underline{x}) \tag{2}$$

where $R(\underline{x})$ is a penalty function that encourages smooth or piecewise-smooth estimates, and β is a parameter that controls the tradeoff between spatial resolution and noise [30]. (Methods for choosing β to specify a desired resolution are described in [30].) Usually, $\mathbf{W}$ is the inverse of the covariance matrix of $\underline{y}$ (accounting for any measurement precorrections) or an estimate thereof [27]. We restrict the presentation to cases where $\mathbf{W}$ is a diagonal matrix, although generalizations are possible.[4] Our goal is to compute an estimate $\hat{\underline{x}}$ of $\underline{x}_{\mathrm{true}}$ from $\underline{y}$ by finding the minimizer of the objective function $\Phi(\underline{x})$.

The unregularized problem (with $\beta = 0$) is poorly conditioned or even underdetermined, so some regularization is required to ensure a stable solution. Gradient-based iterative methods generally converge only to local minima for nonconvex regularizing functions, so we focus here on convex penalty functions [31]. The following general form [32] expresses most of the convex penalty functions that have been proposed for regularization of imaging problems:

$$R(\underline{x}) = \sum_{k=1}^{K} \psi_k([\mathbf{C}\underline{x} - \underline{c}]_k), \tag{3}$$

where $\mathbf{C}$ is a $K \times p$ matrix and $\underline{c} \in R^K$, for some user-defined number K of soft "constraints" of the form $[\mathbf{C}\underline{x}]_k \approx c_k$ The standard roughness penalty simply penalizes differences between neighboring pixel values. This penalty is the special case of (3) where K is the number of pairs of neighboring

[3] One could generalize the approach to produce penalized-likelihood estimates by using iteratively reweighted least-squares [29].

[4] It suffices to have $\mathbf{G}'\mathbf{W}\mathbf{G} \approx \mathbf{D}\mathbf{M}\mathbf{D}$ where $\mathbf{D}$ is diagonal and $\mathbf{M}$ is approximately block-circulant.

pixels[5], the vector $\underline{c} = \underline{0}$, and each row of $\mathbf{C}$ contains one "$+1$" and one "-1" entry so that $[\mathbf{C}\underline{x}]_k$ corresponds to the difference between two neighboring pixel values [33].

In this paper, we first consider general convex nonquadratic functions ψ_k that are symmetric, twice-differentiable, and that have bounded, nonzero second derivatives.[6] We then treat the case of quadratic penalty functions,[7] where $\psi_k(t) = \omega_k t^2/2$ for a positive value ω_k, as a simpler special case. We also assume that the penalty matrix $\mathbf{C}$ has been chosen such that the matrix $\mathbf{G}'\mathbf{W}\mathbf{G} + \mathbf{C}'\mathcal{D}[\omega_k]\mathbf{C}$ is positive definite for any set of positive values $\{\omega_k\}_{k=1}^K$, where $\mathcal{D}[\omega_k]$ denotes the $K \times K$ diagonal matrix with entries $\omega_1, \cdots, \omega_K$ along its diagonal. In particular this assumption implies that $\mathbf{G}$ and $\mathbf{C}$ have disjoint null spaces. Regularization methods are generally designed to ensure such positive definiteness.

Since this paper focuses on comparing various preconditioners, for simplicity we ignore any nonnegativity constraint for $\underline{x}$. One could extend the methods to include a nonnegativity barrier/penalty function [6], [35], or active-set or gradient projection method [36], [37]. Improvements in convergence rate due to improved preconditioners should extend to methods that incorporate nonnegativity, as shown in [38].

B. The Gradients

Under the above assumptions, one can determine the unique minimizer $\hat{\underline{x}}$ of the objective function Φ by finding the zero of its gradient. One can express the column gradient of Φ as follows:

$$-\nabla'\Phi(\underline{x}) = \mathbf{G}'\mathbf{W}(\underline{y} - \mathbf{G}\underline{x}) - \beta\mathbf{C}'\underline{z}(\underline{x}), \tag{4}$$

where $\underline{z} : R^p \to R^K$ is defined by

$$z_k(\underline{x}) \triangleq \dot{\psi}_k([\mathbf{C}\underline{x} - \underline{c}]_k).$$

The Hessian of Φ (its matrix of second partial derivatives) is given by

$$\mathbf{H}(\underline{x}) = \mathbf{G}'\mathbf{W}\mathbf{G} + \beta\mathbf{C}'\mathbf{D}_{\ddot{\psi}}(\underline{x})\mathbf{C}, \tag{5}$$

where $\ddot{\psi}_k(t) = d^2/dt^2\,\psi_k(t)$ and

$$\mathbf{D}_{\ddot{\psi}}(\underline{x}) \triangleq \mathcal{D}[\ddot{\psi}_k([\mathbf{C}\underline{x} - \underline{c}]_k)].$$

In general there is no explicit analytical solution for the zero of the gradient (4). However, in the special case of quadratic penalty functions with $\psi_k(t) = \omega_k t^2/2$, the vector $\underline{z}(\underline{x})$ simplifies to $\underline{z}(\underline{x}) = \mathcal{D}[\omega_k](\mathbf{C}\underline{x} - \underline{c})$, and the gradient simplifies to

$$-\nabla'\Phi(\underline{x}) = \underline{b} - \mathbf{H}\underline{x}$$

where $\mathbf{H} = \mathbf{G}'\mathbf{G} + \beta\mathbf{C}'\mathcal{D}[\omega_k]\mathbf{C}$ and

$$\underline{b} \triangleq \mathbf{G}'\mathbf{W}\underline{y} + \mathbf{C}\mathcal{D}[\omega_k]\underline{c}.$$

[5] $K \approx 2p$ and $K \approx 4p$ for first and second-order neighborhoods, respectively.

[6] This assumption precludes the choice [31] $\psi_k(t) = |t|^p$ for $p < 2$ which has unbounded second derivative.

[7] For certain ROI quantification tasks the quadratic penalty is useful [30] and even outperforms nonquadratic penalties [34].

When $\underline{c} = \underline{0}$, which is the typical choice, $\underline{b} = \mathbf{G}'\mathbf{W}\underline{y}$ is essentially a weighted "backprojection" of $\underline{y}$. In the quadratic case, the zero of the gradient is "simply" the solution of the linear system of equations $\mathbf{H}\hat{\underline{x}} = \underline{b}$. Unfortunately, due to the size of $\mathbf{H}$ for realistic image sizes, even in the quadratic case one cannot compute directly the analytical solution $\hat{\underline{x}} = \mathbf{H}^{-1}\underline{b}$. Furthermore, for nonquadratic penalties there is no explicit analytical solution for $\hat{\underline{x}}$. Thus, one must use iterative methods to compute $\hat{\underline{x}}$.

The preconditioned conjugate-gradient (PCG) iteration is well-suited to solving $\mathbf{H}\hat{\underline{x}} = \underline{b}$, and is also useful for minimizing nonquadratic objective functions. The remainder of this section reviews the PCG algorithm.

C. Minimization Using Conjugate Gradients

Gradient-based minimization methods use the gradient of the objective function $\nabla'\Phi(\underline{x}^n)$ to determine a series of direction vectors $\{\underline{d}^n\}$ along which Φ is minimized via one-dimensional (1-D) line search [39]. Conjugate-gradient methods modify the search directions to ensure that they are mutually conjugate (or approximately so for nonquadratic problems). We use the following preconditioned form of the Polak–Ribiere CG method [6], [39]:

$$\begin{aligned}
\underline{g}^n &= -\nabla'\Phi(\underline{x}^n) \quad \text{(-gradient: see (4))} \\
\underline{p}^n &= \mathbf{M}\underline{g}^n \quad \text{(precondition)} \qquad (6) \\
\gamma_n &= \begin{cases} 0, & n = 0 \\ \dfrac{\langle \underline{g}^n - \underline{g}^{n-1}, \underline{p}^n \rangle}{\langle \underline{g}^{n-1}, \underline{p}^{n-1} \rangle}, & n > 0 \end{cases} \\
\underline{d}^n &= \underline{p}^n + \gamma_n \underline{d}^{n-1} \quad \text{(search direction)} \\
\alpha_n &= \arg\min_{\alpha} \Phi(\underline{x}^n + \alpha\underline{d}^n) \quad \text{(step size)} \qquad (7) \\
\underline{x}^{n+1} &= \underline{x}^n + \alpha_n \underline{d}^n \quad \text{(update)}.
\end{aligned}$$

For quadratic objectives the step size α_n is given explicitly by

$$\alpha_n = \langle \underline{d}^n, \underline{g}^n \rangle / \langle \underline{d}^n, \mathbf{H}\underline{d}^n \rangle,$$

and for $n > 0$ one can use the alternate form

$$\gamma_n = \langle \underline{g}^n, \underline{p}^n \rangle / \langle \underline{g}^{n-1}, \underline{p}^{n-1} \rangle.$$

For nonquadratic objectives Φ, one must find α_n using a line-search. We used the new efficient line-search method described in Section IV to generate the empirical results in Section V.

D. Computation Requirements

When $\mathbf{G}$ is a sparse $n \times p$ matrix, which is the case in tomography, the unpreconditioned CG algorithm for a first-order quadratic penalty function requires about $4\rho np + 8n + 17p$ floating point operations per iteration, where ρ is the fraction of nonzero elements of $\mathbf{G}$. In tomographic image reconstruction problems, typically $n \approx p$. For a tomographic system matrix $\mathbf{G}$ based on a "strip-integral" discretization of the Radon transform, the sparsity factor ρ is about $\rho = 2/\sqrt{n} \approx 2/\sqrt{p}$. Thus, the per-iteration computation of unpreconditioned CG is approximately $8p^{3/2}$. (For a 128^2 image, this approximation yields 16.8 megaflops, which agrees reasonably well with the first line of Table I.) For power-of-two image sizes,

TABLE I
CPU SECONDS AND MEGAFLOPS PER PCG ITERATION FOR THE THREE OBJECTIVE FUNCTIONS, FOR THE VARIOUS PRECONDITIONERS. EACH PCG ITERATION WITH THE PROPOSED PRECONDITIONER REQUIRES 10% MORE FLOPS AND ABOUT 13% MORE CPU TIME THAN WITH THE CONVENTIONAL CIRCULANT PRECONDITIONER

M	CPU Seconds | Mflops Per Iteration					
	QPULS		QPWLS		NPWLS	
I	0.63	18.5	0.64	18.5	0.89	25.3
M_{D}	0.63	18.5	0.64	18.5	0.92	25.3
M_{C}	0.72	21.2	0.73	21.2	0.99	28.1
M_{CDC}	–	–	0.73	21.2	–	–
M_9	–	–	–	–	1.12	30.8

computing an FFT requires about $6p\log_2 p$ flops. Typically, $\log_2 p \ll p^{1/2}$, so preconditioners that are based on FFT's require minimal additional computation for tomographic image reconstruction problems, as evidenced by Table I.

III. PRECONDITIONERS

The matrix $\mathbf{M}$ in (6) above is the preconditioner; choosing this matrix is part of the algorithm design. To guarantee convergence, $\mathbf{M}$ must be symmetric positive definite. For quadratic objectives, preconditioning the CG algorithm is equivalent to solving the transformed linear system of equations $\mathbf{M}^{1/2}\mathbf{H}\mathbf{M}^{1/2}\underline{\tilde{x}} = \mathbf{M}^{1/2}\underline{b}$, where $\underline{\tilde{x}} = \mathbf{M}^{-1/2}\underline{\hat{x}}$, where $\mathbf{H}$ is the Hessian of the objective function as defined in (5). The convergence rate of the CG algorithm generally improves as the condition number of $\mathbf{MH}$ decreases toward unity. For quadratic objective functions, the ideal preconditioner would be $\mathbf{M}_0 = \mathbf{H}^{-1}$ so that $\mathbf{M}_0\mathbf{H} = \mathbf{I}$, because the $p \times p$ identity matrix $\mathbf{I}$ has the minimal condition number (unity), and the preconditioned CG algorithm would converge in one step. For nonquadratic Φ, the inverse-Hessian preconditioner

$$\mathbf{M}_0(\underline{x}) = \mathbf{H}^{-1}(\underline{x}) = [\mathbf{G}'\mathbf{W}\mathbf{G} + \beta\mathbf{C}'\mathbf{D}_{\ddot{\psi}}(\underline{x})\mathbf{C}]^{-1}$$

yields superlinear convergence rates, as does the Newton–Raphson method [40]. Since we cannot compute $\mathbf{H}^{-1}$ for large p, we must develop preconditioners that *approximate* $\mathbf{H}^{-1}$.

A. Diagonal Preconditioner

The classical diagonal preconditioner is simply the inverse of the diagonal elements of $\mathbf{H}$:

$$\mathbf{M}_D(\underline{x}) \triangleq \mathcal{D}\left[\frac{1}{\mathbf{H}_{jj}(\underline{x})}\right]. \tag{8}$$

These diagonals are positive since $\mathbf{H}$ is positive definite. Since $\mathbf{G}$ is sparse and $\mathbf{W}$ is diagonal, one can easily compute the diagonal elements $\mathbf{H}_{jj}(\underline{x})$ without computing all of $\mathbf{H}(\underline{x})$. Implementing this preconditioner within a CG algorithm does not require storing the entire diagonal matrix $\mathbf{M}_D$ for step (7); one simply multiplies each element of the gradient vector by the corresponding diagonal entry of $\mathbf{M}_D$, i.e., $p_j^n = g_j^n/\mathbf{H}_{jj}(\underline{x})$, $j = 1, \cdots, p$. The diagonal preconditioner $\mathbf{M}_D(\underline{x})$ rescales the problem so that $\mathbf{M}_D^{-1}\mathbf{H}$ typically has a smaller condition number than $\mathbf{H}$. Fig. 1 displays a diagonally-preconditioned Hessian $\mathbf{M}_D^{-1}\mathbf{H}$ for a small two-dimensional (2-D) image reconstruction problem; the product is far from the ideal identity matrix, so the convergence rate of PCG is still fairly slow.

B. Circulant Preconditioner

If the Hessian is approximately circulant, then circulant preconditioners are effective. In general. the Hessian (5) is far from circulant. However, for didactic purposes, consider the case of a shift-invariant quadratic penalty with $\psi(t) = t^2/2$, and assume that the weighting matrix $\mathbf{W}$ has an approximately constant diagonal, i.e., $\mathbf{W} \approx \alpha\mathbf{I}$. Then we can make the following approximation

$$\mathbf{H}(\underline{x}) = \mathbf{G}'\mathbf{W}\mathbf{G} + \beta\mathbf{C}'\mathbf{C} \approx \alpha\mathbf{K}(\beta/\alpha), \tag{9}$$

where

$$\mathbf{K}(\eta) \triangleq \mathbf{G}'\mathbf{G} + \eta\mathbf{C}'\mathbf{C}. \tag{10}$$

The utility of approximation (9) is that the matrix $\mathbf{K}$ is suitable for circulant approximations in most image recovery problems. Specifically, for any η there is a diagonal matrix[8] $\mathbf{\Omega}(\eta)$ such that for any power l:

$$\mathbf{K}^l(\eta) = [\mathbf{G}'\mathbf{G} + \eta\mathbf{C}'\mathbf{C}]^l \approx \mathbf{T}'\mathbf{Q}'\mathbf{\Omega}^l(\eta)\mathbf{Q}\mathbf{T}, \tag{11}$$

where $\mathbf{Q}$ is the orthonormal version of the 2-D DFT operator[9]. Such circulant approximations to Toeplitz matrices have been studied extensively, e.g., [15], [22], [41].

The matrix $\mathbf{T}$ is simply the identity matrix when the image size is a power of two and when all pixels within the image are to be estimated. However, often only those pixels within some known support within the image are estimated and the other pixel values are fixed to zero, e.g., Fig. 2. In such cases, the matrix $\mathbf{T}$ is the matrix of ones and zeros that "embeds" the p pixel parameters being estimated into an image "matrix" (or lexicographically ordered version thereof) with size that is a power of 2, which facilitates use of the FFT. In particular, $\mathbf{T}'\mathbf{T} = \mathbf{I}_p$. For simplicity the reader may want to think of $\mathbf{T}$ as the identity matrix, but the full generality is often needed in practice.

Combining (9) with the circulant approximation (11) for $l = -1$ leads to the following circulant preconditioner:

$$\mathbf{M}_C \triangleq \frac{1}{\alpha}\mathbf{T}'\mathbf{Q}'\mathbf{\Omega}^{-1}(\beta/\alpha)\mathbf{Q}\mathbf{T}. \tag{12}$$

Clinthorne *et al.* applied this type of preconditioner to shift-invariant image reconstruction [20]. Circulant preconditioners can be efficiently incorporated into the CG algorithm using 2-D FFT's.

For comparison purposes, we have investigated the effectiveness of the circulant preconditioner even for shift-variant problems where the circulant approximation is poor. The best choice of α is unclear in the shift-variant case. In Section V,

[8] See [33] for details about computing $\mathbf{\Omega}$, which consists of the 2-D discrete Fourier transform (DFT) coefficients of the column of $\mathbf{K}$ corresponding to the pixel at the center of the image.

[9] For any $p_1 \times p_2$ 2-D image lexicographically ordered as a vector $\underline{u}$, $\mathbf{Q}\underline{u}$ is the lexicographically ordered vector of the 2-D DFT coefficients of that image divided by $\sqrt{p_1 p_2}$. Similarly, $\mathbf{Q}'\underline{u}$ is the inverse 2-D DFT of $\underline{u}$, scaled by the same square root factor.

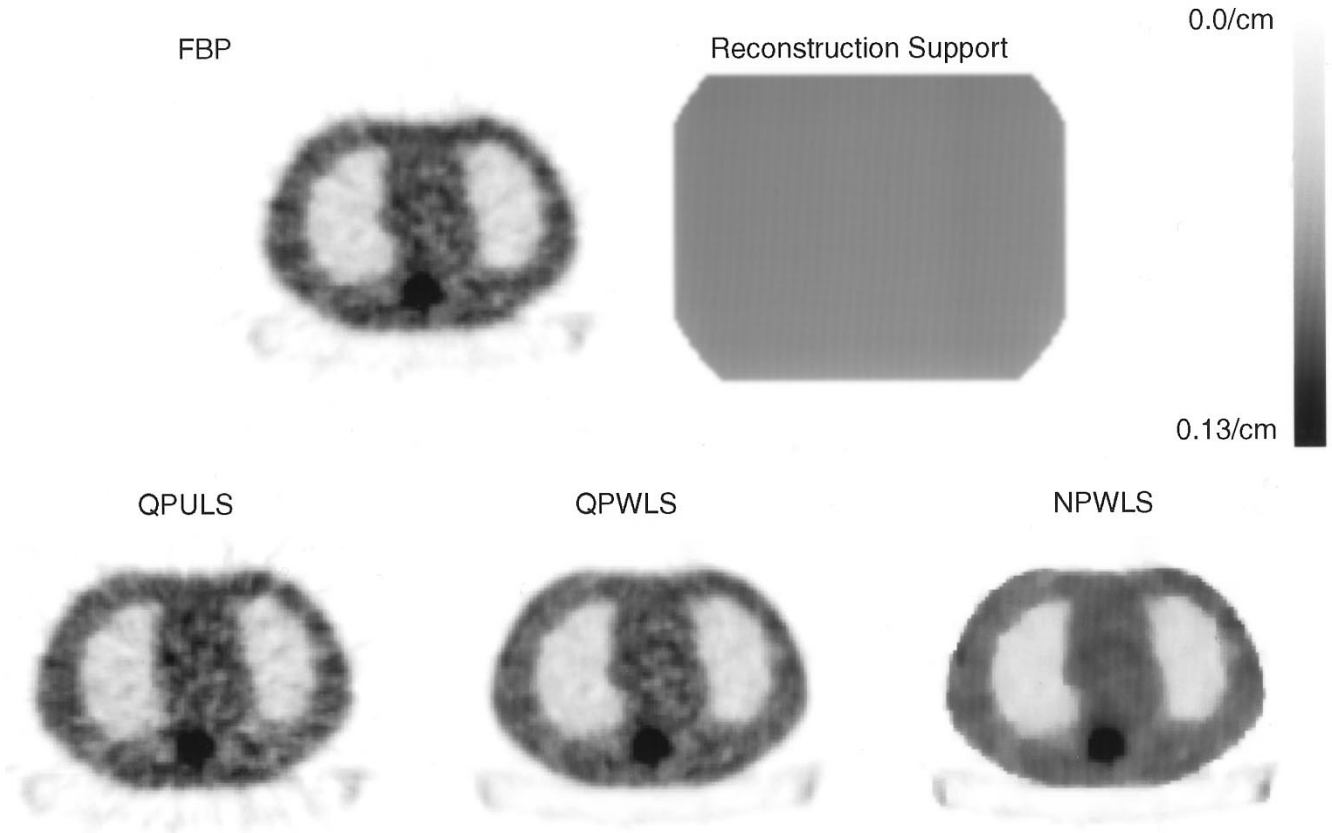


Fig. 2. Grayscale images of the reconstructed FBP image, the reconstruction support (the set of pixels estimated), and the images reconstructed using each of the three objective functions: QPULS, QPWLS, and NPWLS.

we apply the above preconditioner to shift-variant image reconstruction problems using the following choice:

$$\alpha \triangleq \frac{1}{p} \sum_{j=1}^{p} \frac{\sum_i g_{ij}^2 \mathbf{W}_{ii}}{\sum_i g_{ij}^2}.$$

This choice corresponds to the mean κ_j^2 value, where κ_j is defined in (14) below. If $\mathbf{W}$ is a *constant* diagonal matrix, i.e., $\mathbf{W} = \alpha \mathbf{I}$, then α equals that constant and (9) is exact rather than an approximation. The results in Section V demonstrate that $\mathbf{M}_C$ is ineffective for shift-variant problems.

C. Proposed Preconditioner

To understand the limitations of the circulant preconditioner for shift-variant problems, it is helpful to first consider qualitatively how it works in the shift-invariant 1-D case. If $\mathbf{H}$ is Toeplitz, then (ignoring end effects) each column of $\mathbf{H}$ corresponds to the same 1-D kernel but shifted to be centered around the pixel corresponding to that column. The circulant preconditioner $\mathbf{M}_C$ is essentially the inverse filter for that 1-D signal. Computing the product $\mathbf{M}_C \mathbf{H}$ is equivalent to applying that inverse filter to each column of $\mathbf{H}$. When the inverse filter is applied to the kernel of $\mathbf{H}$, the result is (approximately) a Kronecker delta function located at the center of the kernel. Thus $\mathbf{M}_C \mathbf{H} \approx \mathbf{I}$ since in the jth column the kernel is centered about the jth pixel in the 1-D case.

For shift-variant problems, the columns of $\mathbf{H}$ correspond to *different* signals, and no single inverse filter can reduce all columns to Kronecker delta functions simultaneously. Thus, essentially what is needed for shift-variant problems is a separate inverse filter for each column of the Hessian. However, such an approach appears to require p FFT's per iteration, which is impractical. As a practical compromise, our proposed preconditioning approach is the following. We precompute a small number of inverse filters (with a correspondingly moderate number of FFT's) and essentially interpolate between those inverse filters for each pixel (i.e., column of the Hessian). A subtlety is that a naive implementation of the above idea would lead to an asymmetric preconditioner that is not guaranteed to be nonnegative definite. Therefore, we apply the above idea to the matrix square root of the Hessian, and then "square" the resulting matrix to ensure a nonnegative definite preconditioner. From such a nonnegative definite preconditioner one can easily form a positive definite preconditioner by adding $\varepsilon \mathbf{I}$ to the preconditioner for $\varepsilon > 0$.

This section presents a sequence of approximations to the inverse of the Hessian matrix in (5). The final approximations yield a new preconditioner that is efficient computationally.[10] Note that our approximations have no effect on the final solution $\hat{\underline{x}}$, only on the *rate of convergence* to that solution. Thus, poor approximations do not cause reduced estimate accuracy, but merely suboptimal convergence rates. The results in Section V demonstrate that our proposed preconditioners lead to significantly accelerated convergence, despite possibly crude approximations in the development!

[10] I.e., it requires much less computation time than multiplying by $\mathbf{G}$ and $\mathbf{G}'$ in (4), which are the most time consuming steps in the CG iteration for tomographic image reconstruction.

The first key approximation is one that we have previously used for analyzing the spatial resolution properties of tomographic image reconstruction in [30]. Roughly speaking, this approximation brings the diagonal weighting matrix to the outside of the Fisher information matrix as follows:

$$\mathbf{G}'\mathbf{W}\mathbf{G} \approx \mathbf{D}_\kappa \mathbf{G}'\mathbf{G}\mathbf{D}_\kappa, \tag{13}$$

where $\mathbf{D}_\kappa = \mathcal{D}[\kappa_j]$ and

$$\kappa_j \triangleq \sqrt{\frac{\sum_i g_{ij}^2 \mathbf{W}_{ii}}{\sum_i g_{ij}^2}}. \tag{14}$$

This exchange is important in what follows because by assumption $\mathbf{G}'\mathbf{G}$ is approximately block-Toeplitz, i.e., shift-invariant, unlike $\mathbf{G}'\mathbf{W}\mathbf{G}$. The matrices on the two sides of approximation (13) are *exactly* equal along their diagonals, and would also be equal off-diagonal if the $\mathbf{W}_{ii}$'s were all equal. (The results in [30] demonstrate the accuracy and utility of this approximation.) Combining the exchange (13) with (5) leads to our first Hessian approximation:

$$\mathbf{H}(\underline{x}) \approx \mathbf{H}_1(\underline{x}) \triangleq \mathbf{D}_\kappa \mathbf{G}'\mathbf{G}\mathbf{D}_\kappa + \beta \mathbf{C}'\mathbf{D}_{\ddot{\psi}}(\underline{x})\mathbf{C}.$$

The inverse of this approximation is not a practical preconditioner, so we must further simplify. We apply an approximate exchange analogous to (13) to the regularization term:

$$\beta \mathbf{D}_\kappa^{-1}\mathbf{C}'\mathbf{D}_{\ddot{\psi}}(\underline{x})\mathbf{C}\mathbf{D}_\kappa^{-1} \approx \mathbf{D}_\eta(\underline{x})\mathbf{C}'\mathbf{C}\mathbf{D}_\eta(\underline{x}), \tag{15}$$

where $\mathbf{D}_\eta(\underline{x}) \triangleq \mathcal{D}[\sqrt{\eta_j(\underline{x})}]$ and

$$\eta_j(\underline{x}) \triangleq \frac{\beta}{\kappa_j^2} \frac{\sum_k c_{kj}^2 \ddot{\psi}_k([\mathbf{C}\underline{x} - \underline{c}]_k)}{\sum_k c_{kj}^2}. \tag{16}$$

As in (13), the two sides of (15) agree exactly along their diagonals. Furthermore, in the quadratic penalty case with $\psi_k(t) = t^2/2$, we have $\mathbf{D}_{\ddot{\psi}} = \mathbf{I}$ so (15) is then exact.

The factor $\eta_j(\underline{x})$ is an *effective regularization parameter* for the jth pixel [30]. To illustrate, note that if $\mathbf{W} = \mathbf{I}$ and $\psi_k(t) = t^2/2$, then $\kappa_j = 1$ and $\eta_j(\underline{x}) = \beta$ for all j and $\underline{x}$. Combining the approximation (15) with $\mathbf{H}_1(\underline{x})$ leads to our second inverse Hessian approximation:

$$\mathbf{H}^{-1}(\underline{x}) \approx \mathbf{H}_2^{-1}(\underline{x}) \triangleq \mathbf{D}_\kappa^{-1}\mathbf{B}^{-1}(\underline{x})\mathbf{D}_\kappa^{-1}, \tag{17}$$

where

$$\mathbf{B}(\underline{x}) \triangleq \mathbf{G}'\mathbf{G} + \mathbf{D}_\eta(\underline{x})\mathbf{C}'\mathbf{C}\mathbf{D}_\eta(\underline{x}). \tag{18}$$

It is easily shown that $\mathbf{B}$ is symmetric and positive definite under the (reasonable) sufficient conditions that $\ddot{\psi}_k(t) \neq 0\ \forall t$ and that the only vector in the nullspace of $\mathbf{C}$ is the vector of p ones, which must not be in the nullspace of $\mathbf{G}$ [27].

The approximations and preconditioners that we derive below all have the same form as (17), but with different approximations to $\mathbf{B}^{-1}(\underline{x})$. The requirement that the preconditioners be symmetric positive definite increases the challenge here.

To proceed, we exploit the heuristic that the effect of the penalty term is predominately *local*, i.e., the image that one would reconstruct from measurements corresponding to a point source at the jth pixel is determined primarily by the value of η_j. A convenient mathematical expression for this heuristic is as follows:

$$\mathbf{B}^{-1/2}(\underline{x})\underline{e}_j \approx \mathbf{K}^{-1/2}(\eta_j(\underline{x}))\underline{e}_j \triangleq \underline{v}_j(\underline{x}), \tag{19}$$

where $\mathbf{K}$ was defined in (10) and $\underline{e}_j$ is the jth unit vector (i.e., an impulse at the jth pixel). The vector $\underline{v}_j$ is an approximate "inverse filter" for the jth column (or row) of $\mathbf{B}^{1/2}$, in the sense that

$$\underline{e}_k' \mathbf{B}^{1/2}(\underline{x})\underline{v}_j(\underline{x}) \approx \begin{cases} 1, & k = j \\ 0, & k \neq j. \end{cases} \tag{20}$$

We synthesize a matrix approximation from (19) by using the following exact expansion of any $p \times p$ matrix $\mathbf{A}$:

$$\mathbf{A} = \mathbf{A}\mathbf{I} = \mathbf{A}\sum_{j=1}^{p} \underline{e}_j \underline{e}_j' = \sum_{j=1}^{p} (\mathbf{A}\underline{e}_j)\underline{e}_j'.$$

Applying this expansion to the matrix square root $\mathbf{A} = \mathbf{B}^{-1/2}$, which exists since $\mathbf{B}$ is positive definite, and combining with (19) leads to the approximation:

$$\mathbf{B}^{-1/2}(\underline{x}) \approx \sum_{j=1}^{p} \underline{v}_j(\underline{x})\underline{e}_j' \triangleq \mathbf{S}_3(\underline{x}), \tag{21}$$

or equivalently after "squaring":

$$\mathbf{B}^{-1}(\underline{x}) = \mathbf{B}^{-1/2}(\underline{x})\mathbf{B}^{-1/2}(\underline{x}) \approx \mathbf{S}_3'(\underline{x})\mathbf{S}_3(\underline{x}). \tag{22}$$

The matrix $\mathbf{S}_3$ is an approximation to $\mathbf{B}^{-1/2}$; the jth column of $\mathbf{S}_3$ is the jth column of $\mathbf{K}$ evaluated at the effective regularization parameter $\eta_j(\underline{x})$ of the jth pixel. Combining (22) and (21) with (17) leads to our third inverse Hessian approximation:

$$\mathbf{H}^{-1}(\underline{x}) \approx \mathbf{H}_3^{-1}(\underline{x}) \triangleq \mathbf{D}_\kappa^{-1}\mathbf{S}_3'(\underline{x})\mathbf{S}_3(\underline{x})\mathbf{D}_\kappa^{-1}. \tag{23}$$

We have used matrix square roots to ensure symmetry of the approximation. Thus an alternative approximation based on (21) and (22) is as follows[11]:

$$\mathbf{H}^{-1}(\underline{x}) \approx \mathbf{H}_4^{-1}(\underline{x}) \triangleq \mathbf{D}_\kappa^{-1}\mathbf{S}_3(\underline{x})\mathbf{S}_3'(\underline{x})\mathbf{D}_\kappa^{-1}. \tag{24}$$

Both $\mathbf{H}_3$ and $\mathbf{H}_4$ are plausible starting points for subsequent approximations.

The purpose of the above sequence of approximations was to derive a form such as (23) that depends on the approximately circulant matrix $\mathbf{K}$. Applying the circulant approximation (11) for $l = -1/2$ yields:

$$\mathbf{K}^{-1/2}(\eta)\underline{e}_j \approx \mathbf{T}'\mathbf{Q}'\mathbf{\Omega}^{-1/2}(\eta)\mathbf{Q}\mathbf{T}\underline{e}_j,$$

so combining with (19):

$$\underline{v}_j(\underline{x}) \approx \underline{w}_j(\underline{x}) \triangleq \mathbf{T}'\mathbf{Q}'\mathbf{\Omega}^{-1/2}(\eta_j(\underline{x}))\mathbf{Q}\mathbf{T}\underline{e}_j. \tag{25}$$

[11] One can easily show that $\mathbf{H}_4^{-1}$ can also be written in the following form: $\mathbf{H}_4^{-1}(\underline{x}) = \sum_{j=1}^{p} \underline{v}_j(\underline{x})\,\underline{v}_j'(\underline{x})$.

Essentially, $\underline{w}_j$ is an approximate inverse filter for the jth column of $\mathbf{B}^{1/2}$. Combining (25) with (23) and (21) leads to our fifth inverse Hessian approximation:

$$\mathbf{H}^{-1}(\underline{x}) \approx \mathbf{H}_5^{-1}(\underline{x}) \triangleq \mathbf{D}_\kappa^{-1}\mathbf{S}_5'(\underline{x})\mathbf{S}_5(\underline{x})\mathbf{D}_\kappa^{-1}, \quad (26)$$

$$\mathbf{S}_5(\underline{x}) \triangleq \sum_{j=1}^{p} \underline{w}_j(\underline{x})\underline{e}_j'. \quad (27)$$

Like $\mathbf{S}_3$, the matrix $\mathbf{S}_5$ is an approximation to $\mathbf{B}^{-1/2}$; the jth column of $\mathbf{S}_5$ is the jth column of the circulant approximation to $\mathbf{K}$. Again the matrix squaring suggests the alternate form:

$$\mathbf{H}^{-1}(\underline{x}) \approx \mathbf{H}_6^{-1}(\underline{x}) \triangleq \mathbf{D}_\kappa^{-1}\mathbf{S}_5(\underline{x})\mathbf{S}_5'(\underline{x})\mathbf{D}_\kappa^{-1}. \quad (28)$$

Implementing a preconditioner based on either $\mathbf{H}_5^{-1}$ or $\mathbf{H}_6^{-1}$ would appear to require $2(p+1)$ 2-D FFT's per iteration. Although this is less computation than required for $\mathbf{H}_3^{-1}$ or $\mathbf{H}_4^{-1}$, it remains impractical.

To further reduce the number of FFT's, we propose to use interpolation. We choose a small number $m \ll p$ of values $\{\tilde{\eta}_k\}_{k=1}^m$ that cover (most of) the range of the values of the η_j's in (16), and precompute the 2-D DFT coefficients of $\mathbf{K}(\eta)$ in (10) for those values:

$$\mathbf{\Omega}_k \triangleq \mathbf{\Omega}(\tilde{\eta}_k), \quad k = 1, \cdots, m.$$

We then apply interpolation to approximate the 2-D DFT's corresponding to the required values η_j:

$$\mathbf{\Omega}^{-1/2}(\eta_j(\underline{x})) \approx \sum_{k=1}^{m} \lambda_k(\eta_j(\underline{x}))\mathbf{\Omega}_k^{-1/2}, \quad (29)$$

where $\lambda_k \in [0,1]$ are the interpolation factors with $\sum_{k=1}^m \lambda_k = 1$. Since the η's are positively valued, we currently determine the λ_k's by using linear interpolation with a logarithmic scale for the η's:

$$\lambda_k(\eta) \triangleq \begin{cases} \dfrac{\log\tilde{\eta}_{k+1} - \log\eta}{\log\tilde{\eta}_{k+1} - \log\tilde{\eta}_k}, & \tilde{\eta}_k \le \eta \le \tilde{\eta}_{k+1} \\ \dfrac{\log\eta - \log\tilde{\eta}_{k-1}}{\log\tilde{\eta}_k - \log\tilde{\eta}_{k-1}}, & \tilde{\eta}_{k-1} \le \eta \le \tilde{\eta}_k \\ 1, & k = 1, \quad \eta \le \tilde{\eta}_1 \\ 1, & k = m, \quad \eta \ge \tilde{\eta}_m \\ 0, & \text{otherwise.} \end{cases}$$

Incorporating the interpolation approximation (29) into (27) yields

$$\begin{aligned} \mathbf{S}_5(\underline{x}) &= \sum_{j=1}^{p} \mathbf{T}'\mathbf{Q}'\mathbf{\Omega}^{-1/2}(\eta_j)\mathbf{Q}\mathbf{T}\underline{e}_j\underline{e}_j' \\ &\approx \sum_{j=1}^{p} \mathbf{T}'\mathbf{Q}'\left(\sum_{k=1}^{m} \lambda_k(\eta_j(\underline{x}))\mathbf{\Omega}_k^{-1/2}\right)\mathbf{Q}\mathbf{T}\underline{e}_j\underline{e}_j' \\ &= \sum_{k=1}^{m} \mathbf{T}'\mathbf{Q}'\mathbf{\Omega}_k^{-1/2}\mathbf{Q}\mathbf{T}\left(\sum_{j=1}^{p} \lambda_k(\eta_j(\underline{x}))\underline{e}_j\underline{e}_j'\right) \\ &= \sum_{k=1}^{m} \mathbf{T}'\mathbf{Q}'\mathbf{\Omega}_k^{-1/2}\mathbf{Q}\mathbf{T}\mathbf{D}_{\lambda_k}(\underline{x}) \triangleq \mathbf{S}_7(\underline{x}), \end{aligned}$$

where $\mathbf{D}_{\lambda_k}(\underline{x}) \triangleq \mathcal{D}[\lambda_k(\eta_j(\underline{x}))]$. This approximation suggests the following preconditioners:

$$\mathbf{M}_7(\underline{x}) \triangleq \mathbf{D}_\kappa^{-1}\mathbf{S}_7'(\underline{x})\mathbf{S}_7(\underline{x})\mathbf{D}_\kappa^{-1}$$
$$\mathbf{M}_8(\underline{x}) \triangleq \mathbf{D}_\kappa^{-1}\mathbf{S}_7(\underline{x})\mathbf{S}_7'(\underline{x})\mathbf{D}_\kappa^{-1}.$$

Both of these preconditioners require $2(m+1)$ 2-D FFT's (or inverse FFT's) per iteration.

For further simplification, note that

$$\mathbf{S}_7'(\underline{x})\mathbf{S}_7(\underline{x}) = \left(\sum_{k=1}^{m} \mathbf{D}_{\lambda_k}(\underline{x})\mathbf{T}'\mathbf{Q}'\mathbf{\Omega}_k^{-1/2}\right)\mathbf{Q}\mathbf{T}\mathbf{T}'\mathbf{Q}' \times \left(\sum_{k=1}^{m} \mathbf{\Omega}_k^{-1/2}\mathbf{Q}\mathbf{T}\mathbf{D}_{\lambda_k}(\underline{x})\right).$$

Using the approximation $\mathbf{T}\mathbf{T}' \approx \mathbf{I}$ (which is exact when the image size is a power of 2 and when the entire image is estimated) and the fact that $\mathbf{Q}\mathbf{Q}' = \mathbf{I}$ (since $\mathbf{Q}$ is orthonormal), we have the following approximation:

$$\mathbf{S}_7'(\underline{x})\mathbf{S}_7(\underline{x}) \approx \mathbf{S}_9'(\underline{x})\mathbf{S}_9(\underline{x})$$
$$\mathbf{S}_9(\underline{x}) \triangleq \sum_{k=1}^{m} \mathbf{\Omega}_k^{-1/2}\mathbf{Q}\mathbf{T}\mathbf{D}_{\lambda_k}(\underline{x}).$$

This simplification leads to the following preconditioner:

$$\mathbf{M}_9(\underline{x}) \triangleq \mathbf{D}_\kappa^{-1}\mathbf{S}_9'(\underline{x})\mathbf{S}_9(\underline{x})\mathbf{D}_\kappa^{-1}, \quad (30)$$

which requires only $2m$ 2-D FFT's per iteration.

As a final simplification, we could consider dropping the cross terms in the product $\mathbf{S}_9'\mathbf{S}_9$, i.e.

$$\mathbf{S}_9'(\underline{x})\mathbf{S}_9(\underline{x}) \approx \sum_{k=1}^{m} \mathbf{D}_{\lambda_k}(\underline{x})\mathbf{T}'\mathbf{Q}'\mathbf{\Omega}_k^{-1}\mathbf{Q}\mathbf{T}\mathbf{D}_{\lambda_k}(\underline{x}).$$

This approximation leads to the following preconditioner:

$$\mathbf{M}_{10}(\underline{x}) \triangleq \mathbf{D}_\kappa^{-1}\left(\sum_{k=1}^{m} \mathbf{D}_{\lambda_k}(\underline{x})\mathbf{T}'\mathbf{Q}'\mathbf{\Omega}_k^{-1}\mathbf{Q}\mathbf{T}\mathbf{D}_{\lambda_k}(\underline{x})\right)\mathbf{D}_\kappa^{-1},$$

which also requires $2m$ FFT's per iteration. However, we have found empirically that $\mathbf{M}_9$ leads to faster convergence than $\mathbf{M}_{10}$ for nonquadratic penalties [33]. Since $\mathbf{M}_9$ and $\mathbf{M}_{10}$ have equivalent computation time, we consider $\mathbf{M}_9$ to be the preferred preconditioner for the nonquadratic case.

The preconditioners $\mathbf{M}_8$, $\mathbf{M}_9$, $\mathbf{M}_{10}$ are all practical in the sense that the computation of $2m$ 2-D FFT's is $O(mp\log_2 p)$, which for small m is moderate compared with the $O(\rho mp)$ computation required to compute the gradient. Note that while these preconditioners combine aspects of both the diagonal and circulant preconditioning methods, the particular diagonal matrices used here are very different from the conventional diagonal preconditioning matrix given in (8).

D. Circulant Preconditioner Revisited

As a sanity check, consider the shift-invariant case where all of the effective regularization parameters are identical: $\eta_j = \eta_0 \ \forall j$ and $\kappa_j = \kappa \ \forall j$. In this case, $m = 1$ suffices, and clearly we should choose $\tilde{\eta}_1 = \eta_0$. Then the preconditioners $\mathbf{M}_9$ and $\mathbf{M}_{10}$ simplify to the following:

$$\mathbf{M}_9 = \mathbf{M}_{10} = \frac{1}{\kappa^2}\mathbf{T}'\mathbf{Q}'\mathbf{\Omega}^{-1}(\eta_0)\mathbf{Q}\mathbf{T},$$

which is identical to the standard circulant preconditioner (12). Thus our approximations do not lead to any harm in the shift-invariant special case.

E. Implementation

Storing the $p \times p$ preconditioning matrices is fortunately unnecessary, since the CG algorithm requires only the ability to compute the product $\underline{p}^n = \mathbf{M}\underline{g}^n$. FFT's can compute these products efficiently using storage of only a few vectors of length p. For example, we compute $\mathbf{M}_9\underline{g}^n$ as follows:[12]

$$\begin{aligned}
\underline{t}_k^n &= \mathrm{fft}_2\left(\mathbf{T}\mathbf{D}_{\lambda_k}(\underline{x})\mathbf{D}_\kappa^{-1}\underline{g}^n\right) \\
\underline{t}^n &= \sum_{k=1}^{m} \mathbf{\Omega}_k^{-1/2}\underline{t}_k^n \\
\underline{p}_k^n &= \mathbf{T}'\mathrm{fft}_2^{-1}\left(\mathbf{\Omega}_k^{-1/2}\underline{t}^n\right) \\
\underline{p}^n &= \mathbf{D}_\kappa^{-1}\sum_{k=1}^{m}\underline{p}_k^n.
\end{aligned}$$

Even less computation and storage is actually required since one can compute a pair of real FFT's using just one complex FFT [39], and the summations can be done by in-place accumulation.

F. Combined Diagonal/Circulant Preconditioner for the Quadratic Case with Modified Penalty

For a quadratic penalty with $\psi_k(t) = t^2/2$ and $\underline{c} = \underline{0}$, the objective function simplifies to

$$\Phi(\underline{x}) = \frac{1}{2}(\underline{y} - \mathbf{G}\underline{x})'\mathbf{W}(\underline{y} - \mathbf{G}\underline{x}) + \frac{1}{2}\beta\underline{x}'\mathbf{R}\underline{x},$$

where the matrix $\mathbf{R} = \mathbf{C}'\mathbf{C}$ is the (symmetric) Hessian of the penalty function. The usual "uniform" first-order 2-D roughness penalty matrix $\mathbf{R}_0$ has 4's along its diagonal and -1's in the off-diagonal positions corresponding to each pixel's four neighbors. Surprisingly, this uniform penalty leads to *nonuniform spatial resolution* when $\mathbf{W} \neq \mathbf{I}$ [30]. Thus, for quadratic penalties we use the *modified penalty* of [30], for which

$$\mathbf{R} \approx \mathbf{D}_\kappa\mathbf{R}_0\mathbf{D}_\kappa. \tag{31}$$

This modified penalty leads to more uniform spatial resolution [30], which in turn leads to a Hessian that is particularly amenable to preconditioning, since uniform spatial resolution and shift-invariance are equivalent concepts. Under approximation (31), one can verify that $\eta_j = \beta \ \forall j$ for the quadratic penalty. Thus from (17) and (11), the following preconditioner is well-suited to the quadratic objective with the modified penalty:

$$\mathbf{M}_{\mathrm{CDC}} = \mathbf{D}_\kappa^{-1}\mathbf{T}'\mathbf{Q}'\mathbf{\Omega}^{-1}(\beta)\mathbf{Q}\mathbf{T}\mathbf{D}_\kappa^{-1}. \tag{32}$$

Based on [8], we refer to this approach as the "combined diagonal/circulant" preconditioner. The seemingly minor addition to (12) of the $\mathbf{D}_\kappa$ terms in (32) provides significant improvement in convergence rate as shown in Fig. 4 (see Section V). Fig. 1 also illustrates that $\mathbf{M}_{\mathrm{CDC}}\mathbf{H} \approx \mathbf{I}$ for the quadratic case with the modified penalty, and that that approximation is much better than with the circulant preconditioner (12).

G. Summary

We recommend the circulant preconditioner (12) for shift-invariant problems, the combined diagonal/circulant preconditioner (32) for quadratic penalized weighted least squares with the modified penalty of [30], and the new preconditioner (30) for the general shift-variant case.

IV. Line Search Method

For nonquadratic objectives, gradient-based methods require a "line search" in (7) to find the step size α that minimizes the objective function along the current search direction. General purpose line-search methods [39] are applicable but suboptimal, since they fail to exploit the specific form (particularly convexity) of our objective function (2). In this section, we present a new recursive line-search algorithm that is simple to implement and is *guaranteed* to monotonically minimize the objective function with respect to α. This algorithm is an adaptation of the iteration proposed by Huber [42] for robust M-estimation. (The essence of Huber's algorithm has also resurfaced in the imaging literature as the "half-quadratic" method [43]–[45].) The convergence proof in [42] requires the following assumptions: 1) ψ_k is convex, symmetric, and differentiable, and 2)

$$\omega_k(t) \triangleq \dot{\psi}_k(t)/t \tag{33}$$

is bounded and decreasing for $t > 0$. (The ω_k functions act as weighting functions that diminish the influence of neighboring pixels near edges between object regions for nonquadratic penalty functions [42], [46], [47].) These assumptions are reasonable for the convex penalty functions used in imaging.

The line-search algorithm can be stated as follows. Let $\underline{x}$ and $\underline{d}$ denote the current estimate and search direction, respectively. For (7) we would like to find the scalar value $\hat{\alpha}$ that minimizes

$$f(\alpha) = \Phi(\underline{x} + \alpha\underline{d}).$$

First define (and precompute) the following terms:

$$\begin{aligned}
\underline{r} &= \underline{y} - \mathbf{G}\underline{x}, \quad \underline{u} = \mathbf{C}\underline{x} - \underline{c}, \\
\underline{a} &= \mathbf{G}\underline{d}, \quad \underline{h} = \mathbf{C}\underline{d}, \\
f_1 &= \underline{r}'\mathbf{W}\underline{a}, \quad f_2 = \underline{a}'\mathbf{W}\underline{a},
\end{aligned}$$

[12] One can think of $\underline{g}^n$ as both a p-dimensional vector or as a $n_x \times n_y$ image, where n_x and n_y are the image column and row dimensions and $p = n_x n_y$.

where $\mathbf{G}$, $\mathbf{C}$, $\mathbf{W}$, and $\underline{y}$ were defined in Section II. Then with some manipulation of (2) and (3) one can show the following:

$$f(\alpha) = \frac{\underline{r}'\mathbf{W}\underline{r}}{2} - \alpha f_1 + \frac{\alpha^2}{2} f_2 + \beta \sum_k \psi_k(u_k + \alpha h_k)$$

$$\dot{f}(\alpha) = -f_1 + \alpha f_2 + \beta \sum_k h_k \dot{\psi}_k(u_k + \alpha h_k)$$

$$\ddot{f}(\alpha) = f_2 + \beta \sum_k h_k^2 \ddot{\psi}_k(u_k + \alpha h_k).$$

The Newton–Raphson iteration for α is not guaranteed to converge for this problem. However, the following iteration *is* guaranteed to converge to $\hat{\alpha}$:

$$\alpha^{i+1} = \alpha^i - \frac{\dot{f}(\alpha^i)}{f_2 + \beta \sum_k h_k^2 \omega_k \left(u_k + \alpha^i h_k\right)}, \qquad (34)$$

where ω_k was defined in (33). (We initialize with $\alpha^0 = 0$.) The iteration monotonically decreases $f(\alpha^i)$ by the proof of Lemma 8 in [42]. One can also show that the above iteration is a strict contraction: $|\alpha^i - \hat{\alpha}|$ monotonically decreases each iteration [47].

We use about five sub-iterations of the above recursion for each iteration of the CG algorithm. Since no forward or backward projections are required, these sub-iterations are a fairly small component of the computation time for each iteration. To summarize, the algorithm (34) provides an easily implemented yet rapidly converging method for finding the step size α for nonquadratic penalty functions.

V. Numerical Results

To compare the convergence rates of the CG algorithm using the preconditioners described above, we have applied the algorithms to both simulated and real PET transmission scans. Results for PET emission scans were reported in [8]. We have investigated the effects of the initial image, the type of penalty, the choice of weights $\mathbf{W}_{ii}$, and several measures of convergence rates. We synopsize representative results only; complete details are given in [33].

We acquired a 12-min PET transmission scan of a Data Spectrum thorax phantom on an CTI ECAT 921 EXACT PET scanner. This scan produced 920 653 prompt coincidences, which is quite noisy data. The sinogram size was 160 radial bins by 192 angles, with 0.3375 cm radial spacing, so $n = 30\,720$. Prompt and random coincidences were collected separately. The mean contributions of randoms were estimated by time scaling the delayed coincidences from a 15-h blank scan, as described in [48].

The reconstructed images were 128×128 pixels of width 0.42 cm. We reconstructed a FBP image by first smoothing the ratio of the transmission over the blank scan with the "constrained least squares" filter described in [30] and [49], computing the logarithm, and finally applying the ramp filter prior to pixel-driven backprojection. This approach closely matches the spatial resolution of the FBP image with that of the quadratically penalized statistical methods.

The log-likelihood for Poisson transmission data is nonquadratic [50], and therefore not directly of the form given in (2). A conventional approach is to make a ray-by-ray 2nd-order Taylor expansion of the log-likelihood [5], [51] to yield a quadratic functional of the form (2). Unfortunately, that approximation leads to systematic bias [51], [52]. Therefore, we used the FBP image to initialize the grouped-coordinate ascent algorithm of [50], which was run for two iterations to produce an intermediate estimate $\tilde{\underline{x}}$. We then computed a second-order Taylor expansion of the transmission log-likelihood about the reprojection of $\tilde{\underline{x}}$ to produce a quadratic data-fit term of the form given in (2). Regularized versions of this objective function were then minimized using the PCG algorithm. The exact form for y_i and $\mathbf{W}_{ii}$ are given in [33]. The system model $\mathbf{G}$ used for reconstruction assumed parallel strip integrals of width 0.3375 cm.

We investigated the convergence rates of the PCG algorithm with both the FBP image and a zero image as the initial image $\underline{x}^0$. The FBP initialization always led to faster convergence [33], so we report only those results here. A manually determined subset of the pixels (see Fig. 2) was reconstructed, which specifies the matrix $\mathbf{T}$ in (11).

To compare convergence rates, we examined the normalized l_2 distance between the nth iterate $\underline{x}^n$ and the limiting value $\underline{x}^\infty$: $\|\underline{x}^n - \underline{x}^\infty\|/\|\underline{x}^\infty\|$. (We also examined the l_1 and l_∞ norms, which led to comparable conclusions [33].) The PCG algorithms were implemented in ANSI C [53] using single floating-point precision on a DEC AlphaStation 600/5-333 workstation. However, to compute $\underline{x}^\infty$, we implemented a grouped-coordinate ascent algorithm similar to that described in [50] in Matlab using double precision. Several hundred iterations of this algorithm were run, until the change in the estimates reached the limiting precision. This provided a limiting value $\underline{x}^\infty$ that, while of course not "exact," is about 7 digits of precision more accurate than the single precision values computed in C.

We investigated three choices for the objective function. The simplest was quadratically-penalized *unweighted* least squares (QPULS), where $\mathbf{W} = \mathbf{I}$, giving equal weight to all ray measurements. In this case we used the standard first-order 2-D quadratic roughness penalty with $\psi_k(t) = t^2/2$. As illustrated by the image of $\underline{x}^\infty$ shown in Fig. 2, QPULS is a poor choice of an objective function since it ignores the statistics. However, this case has been extensively studied previously since the Hessian of Φ is approximately block-circulant. Therefore, we include it for comparison. Fig. 3 shows that the circulant preconditioner (12) provides significant acceleration for this nearly shift-invariant problem, as expected from previous reports.

The second objective function was quadratically penalized *weighted* least squares (QPWLS), where $\mathbf{W}$ is the curvature term in the second-order Taylor expansion of the transmission log-likelihood [33], [51]. In this case, we used the modified penalty that provides more uniform spatial resolution [30]. As illustrated in Fig. 2, the QPWLS objective function leads to a less noisy image, since ray measurements with greater uncertainty are given lower weight. However, this nonuniform weighting leads to significant shift-variance. As illustrated in Fig. 4, for QPWLS the standard circulant preconditioner performs very poorly. However, the proposed combined di-

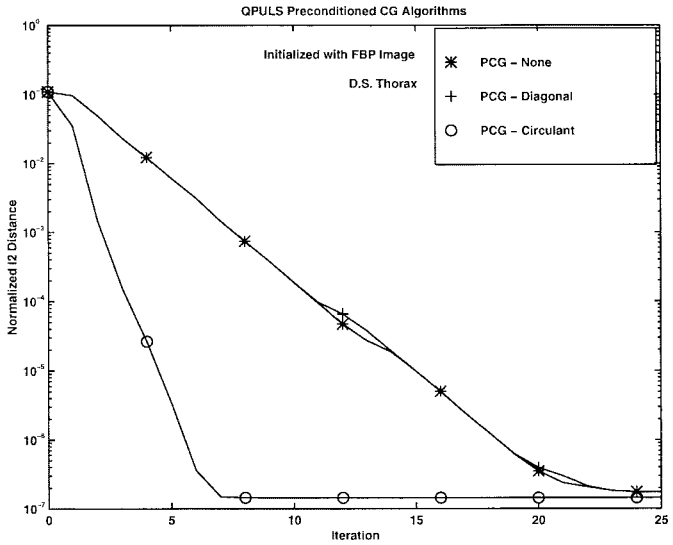


Fig. 3. Normalized l_2 distance to solution $\underline{x}^\infty$ versus iteration n for the quadratically-penalized *unweighted* least squares (QPULS) objective function. Shown is CG with no preconditioning, with the diagonal preconditioner (8), and with the circulant preconditioner (12). In the QPULS case, the Hessian is nearly block-circulant, so the circulant preconditioner provides remarkable acceleration.

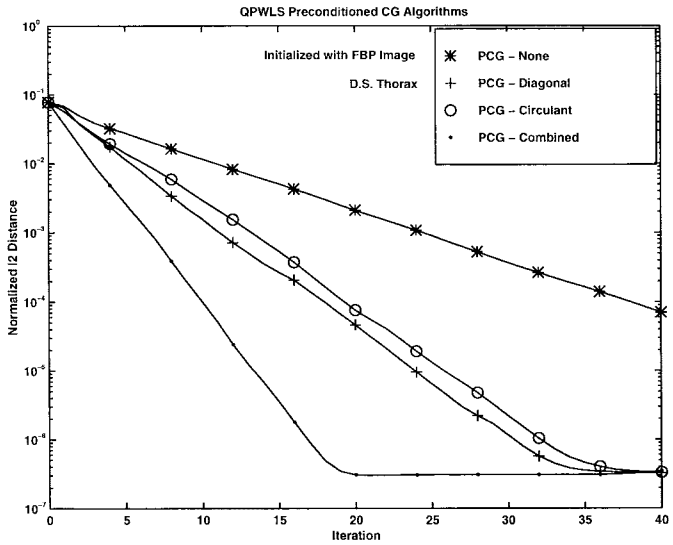


Fig. 4. Normalized l_2 distance to solution $\underline{x}^\infty$ versus iteration n for the quadratically-penalized *weighted* least squares (QPWLS) objective function. Shown is CG with no preconditioning, with the diagonal preconditioner (8), with the circulant preconditioner (12), and with the combined diagonal/circulant preconditioner (32). The new preconditioner $\mathbf{M}_{\mathrm{CDC}}$ leads to significantly faster convergence than the ordinary circulant preconditioner for this shift-variant problem.

agonal/circulant preconditioner (32) provides significant acceleration, with negligible increase in computation time per iteration over the circulant preconditioner.

The third objective function was *nonquadratically*-penalized weighted least squares (NPWLS). We used one of the penalties proposed in [54] for ψ_k:

$$\psi(t) = \delta^2[|t/\delta| - \log(1 + |t/\delta|)], \tag{35}$$

with $\delta = 0.004$ cm. This function is approximately quadratic for $|t| \ll \delta$, but is approximately linear for $|t| \gg \delta$, which provides a degree of edge preservation. We did not

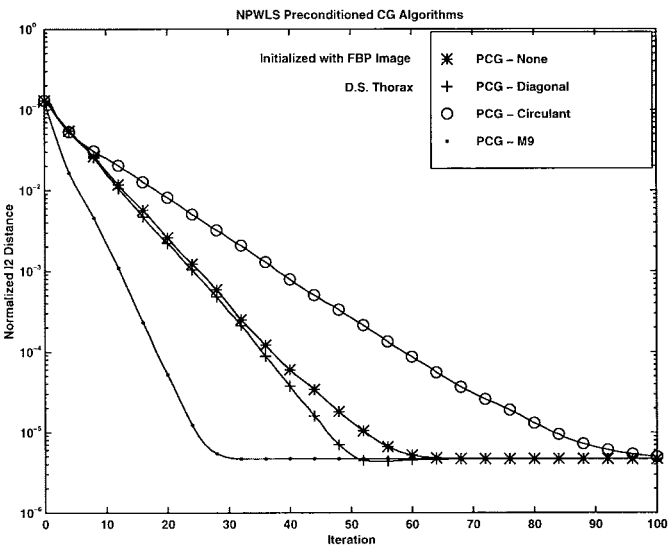


Fig. 5. Normalized l_2 distance to solution $\underline{x}^\infty$ versus iteration n for the *nonquadratically*-penalized weighted least squares (NPWLS) objective function. Shown is CG with no preconditioning, with the diagonal preconditioner (8), with the circulant preconditioner (12), and with the proposed $\mathbf{M}_9$preconditioner (30). The proposed preconditioner provides significant acceleration in convergence rate over conventional preconditioners.

TABLE II
NUMBER OF ITERATIONS AND TOTAL CPU SECONDS REQUIRED FOR THE PCG ITERATES TO "CONVERGE," AS DEFINED IN TEXT. ALTHOUGH THE PROPOSED PRECONDITIONER SLIGHTLY INCREASES COMPUTATION PER ITERATION, THE REDUCED NUMBER OF ITERATIONS LEADS TO SIGNIFICANTLY LESS OVERALL COMPUTATION

	Total Iterations		Total CPU Time			
M	QPULS		QPWLS		NPWLS	
$\boldsymbol{I}$	6	5.1	15	11.3	15	14.6
M_{D}	6	5.3	8	7.1	15	15.2
M_{C}	2	2.9	9	8.3	22	23.2
M_{CDC}	–	–	5	5.5	–	–
M_9	–	–	–	–	7	9.7

use the modified penalty weighting of [30] here, since its effect is currently unknown for nonquadratic penalties. The nonquadratic penalty provides shift-variant smoothing, which, as illustrated in Fig. 2, can lead to (somewhat) improved image quality (particularly for nearly piecewise-constant objects. The nonquadratic penalty leads to significant additional shift-variance, so the circulant preconditioner again performs very poorly, as shown in Fig. 5. However, the proposed preconditioner $\mathbf{M}_9$ leads to significant improvement in convergence rate, with only a modest increase in computation time per iteration (see Tables I and II).

To implement $\mathbf{M}_9$ we used $m = 4$ and $\tilde{\eta}_k = \{0.05, 0.2, 1., 2.\} \cdot \beta/\bar{\kappa}^2$, where $\bar{\kappa}^2 = \frac{1}{p}\sum_{j=1}^{p} \kappa_j^2$. This range reflects the fact that β/κ^2 would be the effective η in the shift-invariant case, and that $\ddot{\psi}_k(t) \leq \ddot{\psi}_k(0)$ so most of the required values of η_j in (16) are smaller than β/κ^2. Increasing m gave negligible improvements, which is expected from Table I of [33]; increasing m may make $\mathbf{M}_9$ a better approximation to $\mathbf{H}_5^{-1}$, but that can only accelerate convergence slightly since $\mathbf{H}_5$ is an imperfect approximation to $\mathbf{H}$.

Table I summarizes the CPU time per iteration of the PCG algorithm (as reported by the UNIX `clock()` function) for the various preconditioners and objective functions for

the data described above. The additional CPU time needed for the proposed preconditioners over conventional circulant preconditioning is modest (less than 15%). The resulting improvements in convergence rate surpass the small increase in computation per iteration. A more realistic system model would use a more dense matrix $\mathbf{G}$, which would further enhance the relative benefits of the proposed preconditioners. Table I also shows the required number of flops per iteration for this problem, as reported by Matlab's `flops` command. The preconditioners require only a modest increase in computation per iteration.

Table II summarizes how many iterations and what total CPU time is required for the PCG algorithm to "converge" for the various preconditioners and objective functions, where we define convergence here by the iterates having reached 99.9% of the asymptotic value of the objective function, i.e. $\Phi(\underline{x}^n) - \Phi(\underline{x}^0) \geq 0.999 \cdot [\Phi(\underline{x}^\infty) - \Phi(\underline{x}^0)]$. Table II clearly shows that even though the preconditioners require a slight increase in computation per iteration, the reduced number of iterations that result will more than compensate for this expense. Thus FFT-based preconditioners are very practical and effective for tomographic imaging problems.

We also compared the CG algorithm using the proposed preconditioners to the coordinate-descent algorithms of [5] and [27]. As in [5], we found that coordinate-descent often converged faster than *unpreconditioned* CG [33]. However, we consistently found that the CG algorithm *with the proposed preconditioners* converged significantly faster than coordinate descent [33]. How a nonnegativity constraint would affect the results requires further evaluation.

VI. Discussion

We have described new conjugate-gradient preconditioning methods for shift-variant imaging problems, and have presented representative results that demonstrate that the proposed preconditioners lead to significantly improved convergence rates over previous diagonal and circulant preconditioners for PET image reconstruction.

The development of the preconditioners used several heuristics about the properties of the Hessian for image reconstruction problems. Although the results show that the methods work, the derivation is somewhat less theoretically satisfying than the optimality results known for circulant preconditioners for Toeplitz problems [15]–[19]. Whether such results can be extended to the more complicated structure of shift-variant problems remains an inviting challenge.

Circulant-based methods are perhaps inherently poorly suited to shift-variant problems. Our preconditioner partially circumvents this drawback by essentially using separate Fourier transforms for each pixel (i.e., for each column of the Hessian), which is then simplified through the interpolation approximation (29). A more direct approach might begin with bases that better accommodate shift-variance, such as wavelets.

Although this paper focuses on image reconstruction, the preconditioning methods are also useful for calculating analytical metrics related to imaging system performance, such as bias, variance, and spatial resolution [30], [52], [55]. Many of these calculations require iterative methods for solving large-scale linear systems of equations involving the Hessian of the objective function.

Although the preconditioner development was presented for 2-D imaging, the entire argument also applies to higher dimensions, such as three-dimensional (3-D) PET imaging, or "2.5-D" imaging, in which each image in a stack is imaged separately but a 3-D penalty is used during reconstruction to provide regularization both within and between planes. We expect these preconditioners will be particularly useful for fully 3-D PET [10]. The inherent shift-variance of $\mathbf{G}'\mathbf{G}$ for such systems will add to the challenge. For 3-D SPECT reconstruction, one may be able to form suitable preconditioners by adapting linear methods such as in [56]. We plan to investigate such methods in the near future.

References

[1] E. Artzy, T. Elfving, and G. T. Herman, "Quadratic optimization for image reconstruction, II," *Comput. Graph. Image Processing*, vol. 11, pp. 242–261, 1979.

[2] S. Kawata and O. Nalcioglu, "Constrained iterative reconstruction by the conjugate gradient method," *IEEE Trans Med. Imag.*, vol. 4, pp. 65–71, June 1985.

[3] A. R. Formiconi, A. Pupi, and A. Passeri, "Compensation of spatial system response in SPECT with conjugate gradient reconstruction technique," *Phys. Med. Biol.*, vol. 34, pp. 69–84, 1989.

[4] L. Kaufman, "Maximum likelihood, least squares, and penalized least squares for PET," *IEEE Trans. Med. Imag.*, vol. 12, pp. 200–214, June 1993.

[5] K. Sauer and C. Bouman, "A local update strategy for iterative reconstruction from projections," *IEEE Trans. Signal Processing*, vol. 41, pp. 534–548, Feb. 1993.

[6] E. U. Mumcuoglu, R. Leahy, S. R. Cherry, and Z. Zhou, "Fast gradient-based methods for Bayesian reconstruction of transmission and emission PET images," *IEEE Trans. Med. Imag.*, vol. 13, pp. 687–701, Dec. 1994.

[7] D. S. Lalush and B. M. W. Tsui, "A fast and stable maximum a posteriori conjugate gradient reconstruction algorithm," *Med. Phys.*, vol. 22, pp. 1273–1284, Aug. 1995.

[8] S. D. Booth and J. A. Fessler, "Combined diagonal/Fourier preconditioning methods for image reconstruction in emission tomography," in *Proc. IEEE Int. Conf. Image Processing*, 1995, vol. 2, pp. 441–444.

[9] A. H. Delaney and Y. Bresler, "A fast and accurate Fourier algorithm for iterative parallel-beam tomography," *IEEE Trans. Image Processing*, vol. 5, pp. 740–753, May 1996.

[10] J. A. Fessler and E. P. Ficaro, "Fully 3D PET image reconstruction using a Fourier preconditioned conjugate-gradient algorithm," in *Proc. IEEE Nuclear Science Symp. Medical Imaging Conf.*, 1996, vol. 3, pp. 1599–1602.

[11] J. A. Fessler, "Preconditioning methods for shift-variant image reconstruction," in *Proc. IEEE Int. Conf. Image Processing*, 1997, vol. 1, pp. 185–188.

[12] O. Axelsson, *Iterative Solution Methods*, Cambridge, Cambridge, 1994.

[13] L. Kaufman, "Implementing and accelerating the EM algorithm for positron emission tomography," *IEEE Trans. Med. Imag.*, vol. 6, pp. 37–51, Mar. 1987.

[14] D. Lalush and B. M. W. Tsui, "The importance of preconditioners in fast Poisson based iterative reconstruction algorithms for SPECT," in *Proc. IEEE Nuclear Science Symp. Medical Imaging Conf.*, 1995, vol. 3, pp. 1326–1330.

[15] R. H. Chan and M. K. Ng, "Conjugate gradient methods for Toeplitz systems," *SIAM Rev.*, vol. 38, pp. 427–482, Sept. 1996.

[16] T. F. Chan, "An optimal circulant preconditioner for Toeplitz systems," *SIAM J. Sci. Stat. Comput.*, vol. 9, pp. 766–771, July 1988.

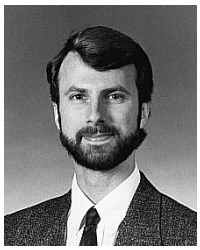
[17] R. H. Chan, J. G. Nagy, and R. J. Plemmons, "FFT-based preconditioners for Toeplitz-block least-squares problems," *SIAM J. Numer. Anal.*, vol. 30, pp. 1740–1768, Dec. 1993.

[18] R. H. Chan and C. K. Wong, "Best-conditioned circulant preconditioners," *Linear Algebra Applic.*, vol. 218, pp. 205–211, 1995.

[19] R. H. Chan, M. K. Ng, and C. K. Wong, "Sine transform based preconditioners for symmetric Toeplitz systems," *Linear Algebra Applicat.*, vol. 232, pp. 237–259, Jan. 1996.

[20] N. H. Clinthorne *et al.*, "Preconditioning methods for improved convergence rates in iterative reconstructions," *IEEE Trans. Med. Imag.*, vol.

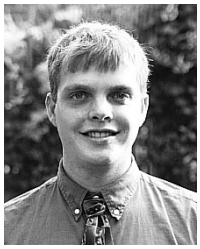
12, pp. 78–83, Mar. 1993.
[21] R. H. Chan, M. K. Ng, and R. J. Plemmons, "Generalization of Strang's preconditioner with applications to Toeplitz least squares problems," *Numer. Linear Algebra Applicat.*, vol. 3, pp. 45–64, 1996.
[22] J. G. Nagy, R. J. Plemmons, and T. C. Torgersen, "Iterative image restoration using approximate inverse preconditioning," *IEEE Trans. Image Processing*, vol. 5, pp. 1151–1162, July 1996.
[23] R. Chan, T. F. Chan, and C. Wong, "Cosine transform based preconditioners for total variation minimization problems in image processing," in *Iterative Methods in Linear Algebra, II, V3, IMACS Series in Computational and Applied Mathematics, Proc. 2nd IMACS Int. Symp. Iterative Methods in Linear Algebra*, S. Margenov and P. Vassilevski, Eds., IMACS, NJ, 1996, pp. 311–329.
[24] C. R. Vogel and M. E. Oman, "Fast numerical methods for total variation minimization in image reconstruction," in *Proc. SPIE 2563, Adv. Signal Process. Algorithms*, 1995, pp. 359–367.
[25] D. L. Snyder, A. M. Hammoud, and R. L. White, "Image recovery from data acquired with a charge-couple-device camera," *J. Opt. Soc. Amer. A*, vol. 10, pp. 1014–1023, May 1993.
[26] D. L. Snyder *et al.*, "Compensation for readout noise in CCD images," *J. Opt. Soc. Amer. A*, vol. 12, pp. 272–283, Feb. 1995.
[27] J. A. Fessler, "Penalized weighted least-squares image reconstruction for positron emission tomography," *IEEE Trans. Med. Imag.*, vol. 13, pp. 290–300, June 1994.
[28] J. G. Nagy and D. P. O'Leary, "Restoring images degraded by spatially-variant blur," *SIAM J. Sci. Comput.*, vol. 19, pp. 1063–1082, July 1998.
[29] P. J. Green, "Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives," *J. R. Stat. Soc. B*, vol. 46, pp. 149–192, 1984.
[30] J. A. Fessler and W. L. Rogers, "Spatial resolution properties of penalized-likelihood image reconstruction methods: Space-invariant tomographs," *IEEE Trans. Image Processing*, vol. 5, pp. 1346–1358, Sept. 1996.
[31] C. Bouman and K. Sauer, "A generalized Gaussian image model for edge-preserving MAP estimation," *IEEE Trans. Image Processing*, vol. 2, pp. 296–310, July 1993.
[32] A. R. De Pierro, "A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography," *IEEE Trans. Med. Imaging*, vol. 14, pp. 132–137, Mar. 1995.
[33] J. A. Fessler, "Conjugate-gradient preconditioning methods: Numerical results," Tech. Rep. 303, Commun. Signal Process. Lab., Dept. Elect. Eng., Comput. Sci., Univ. Michigan, Ann Arbor, MI, Jan. 1997, available from http://www.eecs.umich.edu/~fessler.
[34] E. U. Mumcuoglu, R. M. Leahy, Z. Zhou, and S. R. Cherry, "A phantom study of the quantitative behavior of Bayesian PET reconstruction methods," in *Proc. IEEE Nuclear Science Symp. Medical Imaging Conf.*, 1995, vol. 3, pp. 1703–1707.
[35] E. U. Mumcuoğlu, R. M. Leahy, and S. R. Cherry, "Bayesian reconstruction of PET images: Methodology and performance analysis," *Phys. Med. Biol.*, vol. 41, pp. 1777–1807, Sept. 1996.
[36] E. U. Mumcuoğlu and R. M. Leahy, "A gradient projection conjugate gradient algorithm for Bayesian PET reconstruction," in *Proc. IEEE Nuclear Science Symp. Medical Imaging Conf.*, 1994, vol. 3, pp. 1212–1216.
[37] J. J. Moré and G. Toraldo, "On the solution of large quadratic programming problems with bound constraints," *SIAM J. Optim.*, vol. 1, pp. 93–113, Feb. 1991.
[38] M. Bierlaire, P. L. Toint, and D. Tuyttens, "On iterative algorithms for linear least squares problems with bound constraints," *Linear Algebra Applicat.*, vol. 143, pp. 111–143, 1991.
[39] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C*. Cambridge, U.K.: Cambridge Univ. Press, 1988.
[40] M. Al-Baali and R. Fletcher, "On the order of convergence of preconditioned nonlinear conjugate gradient methods," *SIAM J. Sci. Comput.*, vol. 17, pp. 658–665, May 1996.
[41] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
[42] P. J. Huber, *Robust Statistics*. New York: Wiley, 1981.
[43] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud, "Two deterministic half-quadratic regularization algorithms for computed imaging," in *Proc. IEEE Int. Conf. Image Processing*, 1994, vol. 2, pp. 168–171.
[44] D. Geman and G. Reynolds, "Constrained restoration and the recovery of discontinuities," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 367–383, Mar. 1992.
[45] D. Geman and C. Yang, "Nonlinear image recovery with half-quadratic regularization," *IEEE Trans. Image Processing*, vol. 4, pp. 932–946, July 1995.
[46] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud, "Deterministic edge-preserving regularization in computed imaging," *IEEE Trans. Image Processing*, vol. 6, pp. 298–311, Feb. 1997.
[47] J. A. Fessler, "Grouped coordinate descent algorithms for robust edge-preserving image restoration," in *Proc. SPIE 3071, Image Reconst. Restor. II*, pp. 184–194, 1997.
[48] M. Yavuz and J. A. Fessler, "New statistical models for randoms-precorrected PET scans," in *Lecture Notes in Computer Science, Vol. 1230: Information Processing in Medical Imaging*, J. Duncan and G. Gindi, Eds. Berlin, Germany: Springer-Verlag, 1997, pp. 190–203.
[49] J. A. Fessler, "Resolution properties of regularized image reconstruction methods," Tech. Rep. 297, Commun. Signal Process. Lab., Dept. Elect. Eng. Comput. Sci., Univ. Michigan, Ann Arbor, MI, Aug. 1995.
[50] 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, pp. 166–175, Apr. 1997.
[51] J. A. Fessler, "Hybrid Poisson/polynomial objective functions for tomographic image reconstruction from transmission scans," *IEEE Trans. Image Processing*, vol. 4, pp. 1439–1450, Oct. 1995.
[52] ———, "Mean and variance of implicitly defined biased estimators (such as penalized maximum likelihood): Applications to tomography," *IEEE Trans. Image Processing*, vol. 5, pp. 493–506, Mar. 1996.
[53] ———, "ASPIRE 3.0 user's guide: A sparse iterative reconstruction library," Tech. Rep. 293, Commun. Signal Process. Lab., Dept. Elect. Eng. Comput. Sci., Univ. Michigan, Ann Arbor, July 1995.
[54] K. Lange, "Convergence of EM image reconstruction algorithms with Gibbs smoothing," *IEEE Trans. Med. Imag.*, vol. 9, pp. 439–446, Dec. 1990.
[55] A. O. Hero, M. Usman, A. C. Sauve, and J. A. Fessler, "Recursive algorithms for computing the Cramer-Rao bound," *IEEE Trans. Signal Processing*, vol. 45, pp. 803–807, Mar. 1997.
[56] W. Xia, R. M. Lewitt, and P. R. Edholm, "Fourier correction for spatially variant collimator blurring in SPECT," *IEEE Trans. Med. Imag.*, vol. 14, pp. 100–115, Mar. 1995.



Jeffrey A. Fessler (S'83–M'90) received the B.S.E.E. degree from Purdue University, West Lafayette, IN, in 1985, and the M.S.E.E. degree and M.S. degree in statistics, both from Stanford University, Stanford, CA, in 1986 and 1989, respectively. From 1985 to 1988, he was a National Science Foundation Graduate Fellow at Stanford University, where he received the Ph.D. degree in electrical engineering in 1990.

He has been with the University of Michigan, Ann Arbor, since 1990. From 1991 to 1992, he was a Department of Energy Alexander Hollaender Post-Doctoral Fellow in the Division of Nuclear Medicine. From 1993 to 1995, he was an Assistant Professor in Nuclear Medicine and the Bioengineering Program. Since 1995, he has been with the Department of Electrical Engineering and Computer Science, where he is an Associate Professor. His research interests are in statistical aspects of imaging problems.

Dr. Fessler received the Francois Erbsmann award for his IPMI'93 presentation. He serves as an associate editor for IEEE TRANSACTIONS ON MEDICAL IMAGING and IEEE TRANSACTIONS ON IMAGE PROCESSING, and was co-chair of the 1997 SPIE conference on Image Reconstruction and Restoration.



Scott D. Booth received the B.S. degree in biomedical and electrical engineering from Duke University, Durham, NC, in 1992, and the M.S. degree in bioengineering from the University of Michigan, Ann Arbor, in 1996.

In 1996 and 1997, he was with Cybernet Systems Corporation, Ann Arbor, where he designed computer hardware and software for embedded systems in medical instruments. He is currently with the University of Virginia Medical Center, Charlottesville, where he is working to automate the hospital's Clinical Laboratories. His research interests include iterative optimization methods, signal processing, statistical analysis, and simulation modeling.