Efficient Calculation of Resolution and Covariance for Penalized-Likelihood Reconstruction in Fully 3-D SPECT

J. Webster Stayman* and Jeffrey A. Fessler, *Senior Member, IEEE*

Abstract—Resolution and covariance predictors have been derived previously for penalized-likelihood estimators. These predictors can provide accurate approximations to the local resolution properties and covariance functions for tomographic systems given a good estimate of the mean measurements. Although these predictors may be evaluated iteratively, circulant approximations are often made for practical computation times. However, when numerous evaluations are made repeatedly (as in penalty design or calculation of variance images), these predictors still require large amounts of computing time. In Stayman and Fessler (2000), we discussed methods for precomputing a large portion of the predictor for shift-invariant system geometries. In this paper, we generalize the efficient procedure discussed in Stayman and Fessler (2000) to shift-variant single photon emission computed tomography (SPECT) systems. This generalization relies on a new attenuation approximation and several observations on the symmetries in SPECT systems. These new general procedures apply to both two-dimensional and fully three-dimensional (3-D) SPECT models, that may be either precomputed and stored, or written in procedural form. We demonstrate the high accuracy of the predictions based on these methods using a simulated anthropomorphic phantom and fully 3-D SPECT system. The evaluation of these predictors requires significantly less computation time than traditional prediction techniques, once the system geometry specific precomputations have been made.

Index Terms—Image quality, local impulse response, noise, tomography, variance.

I. INTRODUCTION

IN THE ANALYSIS of tomographic systems and their reconstruction algorithms, one would often like to be able to predict the noise and resolution properties of the reconstructed images. Much of the literature discussing noise and resolution properties investigates the image properties as a function of specific algorithm and iteration [2]–[4]. Here, we focus on the noise and resolution properties of estimators that maximize penalized-likelihood objective functions using algorithms that have been iterated to a nearly converged solution. Equations for predicting the mean and variance of such estimators have been derived in [5] and for resolution properties in [6]. These predictions have been applied to several applications including penalty design for uniform resolution [1], [6], [7], contrast optimization [8], and for computer observer models [9], [10]. While resolution and noise prediction has potential uses across a range of applications, calculation of the predictions is computationally expensive. This paper investigates a number of approximations that make these evaluations more practical when many evaluations need to be made for a particular SPECT geometry.

Approximations based on local space-invariance have been used [1], [9], [11] to provide good noise and resolution predictions with reasonable computation times. However, when very many estimates are required (e.g.: when resolution estimates are made for every pixel position, or noise estimates are made repeatedly for different reconstruction parameters or objects), the computational burden is still high. Generally the dominant computation is the calculation of repeated weighted backprojections of projection data. In some cases, as in an idealized positron emission tomography (PET) system where the system response is space-invariant, computation time can be reduced through an appropriate factorization and precomputation (see [1]). Qi [11] has used such a factorization and identified a number of system symmetries in three-dimensional (3-D) PET to greatly reduce computation and storage requirements for resolution and covariance prediction. (Similar techniques will be adopted here for use in the context of 3-D SPECT.) However, the previously investigated factorization approaches used in PET rely on a shift-invariant geometric system model and are not directly applicable to space-variant systems, such as for SPECT with a depth-dependent detector response. We have previously investigated fast methods for cases where the space-variant system may be modeled with a precomputed system matrix [7]. Such methods are generally impractical for three-dimensional system models, where the system model is too large to be precomputed and stored. Other attempts at reducing the calculation time have been made in [12] and [13] using the frequency-distance principle [14] and the approximation developed in [6, (30)–(32)].

In this paper, we discuss an alternative technique for making noise and resolution estimates for both two-dimensional (2-D) and fully 3-D SPECT systems. These methods are appropriate for systems where the system matrix must be computed "on-the-fly," and apply generally to systems that include noncircular orbits, nonuniform attenuation, and depth-dependent detector response. The approach presented here is an extension of the linear

operator approach discussed in [1] for space-invariant PET systems. Section II reviews the resolution and covariance predictors for penalized-likelihood SPECT reconstruction. Sections III and IV identify various approximations that allow us to precompute and store portions of the resolution and covariance predictors that are object-independent. We show how to use these precomputations to arrive at rapid predictions once the object-dependencies (i.e., the attenuation map and an estimate of the mean measurements) are known. This section also includes a discussion of the computation and storage requirements. In Section V, we validate our approximations using a simulated 3-D SPECT system and digital phantom. We show that our approximate resolution predictions are very close to iteratively calculated predictions. Similarly, we demonstrate that our covariance approximations agree well with sample covariances calculated from an ensemble of 3-D reconstructions.

II. BACKGROUND

Consider the following SPECT model. Let $\underline{Y} = (Y_1, \ldots, Y_N)$ denote the vector of N measurements. We parameterize the emission image using a pixel or voxel basis where $\underline{\lambda} = (\lambda_1, \ldots, \lambda_P)$ denotes a vector of nonnegative emission densities. Let x, y, and, z denote image coordinates and P_x, P_y, and P_z, denote the number of voxels along each direction of the discretized volume. The x and y terms denote the in-plane coordinates, the z term represents the axial coordinate, and $P = P_x P_y P_z$. The measurements means are related to the emission image through the following linear model:

$$\bar{Y}_i(\underline{\lambda}) = \sum_{j=1}^{P} h_{ij} \lambda_j + r_i$$
$$\bar{\underline{Y}}(\underline{\lambda}) = \boldsymbol{H}\underline{\lambda} + \underline{r} \qquad (1)$$

where $\boldsymbol{H}$ is the $N \times P$ system matrix that contains the $\{h_{ij}\}$ terms that model the projection operation. The vector $\underline{r} = (r_1, \ldots, r_N)$ is assumed to be a known quantity that represents the mean contribution of random events like background and scatter. The system matrix, $\boldsymbol{H}$, models all projection effects including detector responses, the detector orbit, and object-dependent effects like attenuation. This matrix can be precalculated, or defined implicitly as in the case of projectors and backprojectors that are implemented in a procedural form.

We reconstruct the emission densities from the measurements using a penalized-likelihood estimator, which is written as the implicit maximizer of an objective function (which we assume to be strictly convex to ensure a unique solution)

$$\hat{\underline{\lambda}} = \arg\max_{\underline{\lambda}} \Phi(\underline{\lambda}, \underline{Y})$$
$$\Phi(\underline{\lambda}, \underline{Y}) \triangleq L(\underline{\lambda}, \underline{Y}) - R(\underline{\lambda}) \qquad (2)$$

where the objective function, $\Phi(\underline{\lambda}, \underline{Y})$, is the difference of a log-likelihood term, $L(\underline{\lambda}, \underline{Y})$, and a roughness penalty term, $R(\underline{\lambda})$. We consider log-likelihoods of the following form:

$$L(\underline{\lambda}, \underline{Y}) = \sum_i L_i\left(Y_i, \bar{Y}_i(\underline{\lambda})\right) \qquad (3)$$

where $L_i(u, v)$ is a function that represents the log-likelihood for the ith measurement, u, and its mean, v. The general form of (3) accommodates a wide range of noise models with independent measurements.

A. Resolution Properties

Resolution properties for converged solutions of implicitly defined estimators have been discussed in [6], [7]. One can quantify local resolution properties by finding the local impulse response. The local impulse response was derived in [6] and can be written as follows:

$$\underline{l}^j = \left[-\nabla^{20}\Phi\left(\breve{\underline{\lambda}}, \bar{\underline{Y}}(\underline{\lambda})\right)\right]^{-1} \nabla^{11}\Phi\left(\breve{\underline{\lambda}}, \bar{\underline{Y}}(\underline{\lambda})\right) \frac{\partial}{\partial \lambda_j} \bar{\underline{Y}}(\underline{\lambda})$$
$$(4)$$

where ∇^{20} is an operator that yields a matrix whose (j, k)th element is $(\partial^2/\partial\lambda_j\partial\lambda_k)$, ∇^{11} is an operator that yields a matrix whose (j, i)th element is $(\partial^2/\partial\lambda_j\partial Y_i)$, and $\breve{\underline{\lambda}} \triangleq \hat{\underline{\lambda}}(\bar{\underline{Y}}(\underline{\lambda}))$, the estimate of $\underline{\lambda}$ using the mean data. Here, $\underline{l}^j$ is a vector that represents the 2-D or 3-D local impulse response function for a perturbation of the jth voxel.

Plugging (1) and (3) into (4), we find that the formulation for the local impulse response can be written as

$$\underline{l}^j = \left[\boldsymbol{H}'\boldsymbol{D}_1\boldsymbol{H} + \boldsymbol{R}(\breve{\underline{\lambda}})\right]^{-1} \boldsymbol{H}'\boldsymbol{D}_2\boldsymbol{H}\underline{e}^j \qquad (5)$$

where $'$ denotes the transpose operation, $\underline{e}^j$ denotes the jth unit vector, $\boldsymbol{R}(\underline{\lambda})$ is the Hessian of the penalty, and $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ are diagonal matrices whose elements are defined as follows:

$$[\boldsymbol{D}_1]_{ii} = -L_i^{02}\left(\bar{Y}_i(\underline{\lambda}), \bar{Y}_i(\breve{\underline{\lambda}})\right) \qquad (6)$$
$$[\boldsymbol{D}_2]_{ii} = L_i^{11}\left(\bar{Y}_i(\underline{\lambda}), \bar{Y}_i(\breve{\underline{\lambda}})\right) \qquad (7)$$

where $L_i^{02}(u, v) = (\partial^2/\partial v^2)L_i(u, v)$ and $L_i^{11}(u, v) = (\partial^2/\partial u \partial v)L_i(u, v)$. We will assume throughout the paper that (5) is a well-behaved function. This is true for sufficiently regularized image reconstruction problems; however, for unregularized maximum-likelihood estimation (5) is often not well-behaved and is not a good way to predict resolution properties.

In cases where the mean measurements are unavailable, it is common to approximate responses by simply plugging the noisy measurements, $\underline{Y}$ into $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$. This does not change the basic form for the local impulse response and typically yields good approximations. Nonquadratic penalties will require some kind of estimate of $\underline{\lambda}$ to evaluate $\boldsymbol{R}(\breve{\underline{\lambda}})$, if $\underline{\lambda}$ is unknown. However, the Hessian of quadratic penalties can be represented by a constant matrix, $\boldsymbol{R}$, which is object-independent.

Direct evaluation of (5) is often difficult due to the size of the matrices and the presence of the matrix inverse. While one may approximate the local impulse response using iterative techniques,[1] for many applications this is too computationally expensive. However, since $\boldsymbol{H}'\boldsymbol{D}_1\boldsymbol{H}\underline{e}^j$ and $\boldsymbol{H}'\boldsymbol{D}_2\boldsymbol{H}\underline{e}^j$ are approximately locally space-invariant for varying j, one can use a cir-

---

[1]Recall that (5) represents the solution to a linear system of equations $([\boldsymbol{H}'\boldsymbol{D}_1\boldsymbol{H} + \boldsymbol{R}(\underline{\lambda})]\underline{x} = \boldsymbol{H}'\boldsymbol{D}_2\boldsymbol{H}\underline{e}^j)$ which may be solved iteratively using a number of different approaches such as the conjugate gradient algorithm.

culant approximation to (5) [1], [8], [11]. Specifically, we may approximate[2] (5) as

$$\underline{l}^j_{\mathrm{circ}} = \mathcal{F}^{-1} \left\{ \frac{\mathcal{F}\{\underline{e}^j\} \odot \mathcal{F}\{\boldsymbol{H}'\boldsymbol{D}_2\boldsymbol{H}\underline{e}^j\}}{\mathcal{F}\left\{\boldsymbol{H}'\boldsymbol{D}_1\boldsymbol{H}\underline{e}^j + R(\underline{\breve{\lambda}})\underline{e}^j\right\}} \right\} \quad (8)$$

where $\mathcal{F}\{\cdot\}$ represents the 2-D or 3-D (as appropriate) Fourier transform of its vector argument, the division is an element-by-element division, and $\odot$ represents an element-by-element multiplication. This approximation relies on the fact that the 3-D (or 2-D) Fourier transform can be used to diagonalize a triply (doubly) block-circulant matrix and the eigenvalues of that matrix may be formed by Fourier transforming a row of that circulant matrix. Also note the inclusion of the $\mathcal{F}\{\underline{e}^j\}$ term that incorporates the appropriate shift[3] to ensure the response is "centered" at location j. Thus, (8) can be calculated relatively quickly using fast Fourier transforms. Generally the most computationally expensive part of calculating (8) lies in the calculation of the weighted projection/backprojections. For fully 3-D SPECT systems the computational burden of the projections and backprojections greatly outweighs the burden due to the Fourier transform operations.

B. Covariance Estimation

In [5], an approximation for the covariance of implicitly defined estimators was derived. We restate that approximation here

$$\mathrm{Cov}\{\underline{\hat{\lambda}}\} = \left[-\nabla^{20}\Phi\left(\underline{\breve{\lambda}}, \underline{\bar{Y}}(\underline{\lambda})\right) \right]^{-1} \nabla^{11}\Phi\left(\underline{\breve{\lambda}}, \underline{\bar{Y}}(\underline{\lambda})\right)$$
$$\cdot \mathrm{Cov}\{\underline{Y}\} \left[\nabla^{11}\Phi\left(\underline{\breve{\lambda}}, \underline{\bar{Y}}(\underline{\lambda})\right) \right]' \left[-\nabla^{20}\Phi\left(\underline{\breve{\lambda}}, \underline{\bar{Y}}(\underline{\lambda})\right) \right]^{-1} \quad (9)$$

Plugging in (1) and (3), yields

$$\mathrm{Cov}\{\underline{\hat{\lambda}}\} = \left[\boldsymbol{H}'\boldsymbol{D}_1\boldsymbol{H} + R(\underline{\breve{\lambda}}) \right]^{-1} \boldsymbol{H}'\boldsymbol{D}_2\mathrm{Cov}\{\underline{Y}\}$$
$$\times \boldsymbol{D}_2\boldsymbol{H} \left[\boldsymbol{H}'\boldsymbol{D}_1\boldsymbol{H} + R(\underline{\breve{\lambda}}) \right]^{-1} \quad (10)$$

with $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ as defined in (6) and (7). Under the assumption of independent measurements $\mathrm{Cov}\{\underline{Y}\}$ is a diagonal matrix, thus we may write

$$\mathrm{Cov}\{\underline{\hat{\lambda}}\} = \left[\boldsymbol{H}'\boldsymbol{D}_1\boldsymbol{H} + R(\underline{\breve{\lambda}}) \right]^{-1} \boldsymbol{H}'\boldsymbol{D}_3\boldsymbol{H}$$
$$\times \left[\boldsymbol{H}'\boldsymbol{D}_1\boldsymbol{H} + R(\underline{\breve{\lambda}}) \right]^{-1} \quad (11)$$

where $\boldsymbol{D}_3 = \boldsymbol{D}_2\mathrm{Cov}\{\underline{Y}\}\boldsymbol{D}_2$. As with the resolution predictor, we will assume that (11) is a well-behaved function.

As in the case of the local impulse response, computing (11) is usually impractical. Iterative techniques were described in [5] for evaluating columns of (11), but they still require considerable computation. One may use a circulant approximation [11]

---

to (11), so that the covariance for the jth pixel position is approximated as

$$\mathrm{Cov}^j_{\mathrm{circ}}\{\underline{\hat{\lambda}}\} = \mathcal{F}^{-1} \left\{ \frac{\mathcal{F}\{\underline{e}^j\} \odot \mathcal{F}\{\boldsymbol{H}'\boldsymbol{D}_3\boldsymbol{H}\underline{e}^j\}}{\left| \mathcal{F}\left\{\boldsymbol{H}'\boldsymbol{D}_1\boldsymbol{H}\underline{e}^j + R(\underline{\breve{\lambda}})\underline{e}^j\right\} \right|^2} \right\}. \quad (12)$$

Again, calculations are dominated by the evaluation of terms of the form $\boldsymbol{H}'\boldsymbol{D}\boldsymbol{H}\underline{e}^j$, where we have taken $\boldsymbol{D}$ to denote a generic diagonal weighting. (That is, $\boldsymbol{D}$ could be any one of $\{\boldsymbol{D}_1, \boldsymbol{D}_2, \boldsymbol{D}_3\}$.) This term is commonly referred to as the Fisher information matrix, since $\boldsymbol{D}$ often represents an inverse of the measurement covariance, and the same form arises from estimation bounds on variance. This form can be found in a number of applications including resolution and variance prediction. Therefore, it would be very helpful to find efficient ways of calculating $\boldsymbol{H}'\boldsymbol{D}\boldsymbol{H}\underline{e}^j$.

Fast calculation of $\boldsymbol{H}'\boldsymbol{D}\boldsymbol{H}\underline{e}^j$ has been previously explored for resolution [6] and variance [15] predictors for space-invariant systems, where $\boldsymbol{H}'\boldsymbol{D}\boldsymbol{H}\underline{e}^j \approx \boldsymbol{D}^*\boldsymbol{G}'\boldsymbol{G}\boldsymbol{D}^*\underline{e}^j$. In this case $\boldsymbol{G}'\boldsymbol{G}$ represents a space-invariant operator and is premultiplied and postmultiplied by a diagonal weighting matrix. While this method yields fairly good variance estimates, the approximation often poorly models anisotropy in the local impulse response [1] and the covariance function.

III. EFFICIENT CALCULATION OF $\boldsymbol{H}'\boldsymbol{D}\boldsymbol{H}\underline{e}^j$

We have previously investigated a better approximations for efficient for calculation of $\boldsymbol{H}'\boldsymbol{D}\boldsymbol{H}\underline{e}^j$, when $\boldsymbol{H}$ can be factored such that $\boldsymbol{H} \approx \boldsymbol{D}_{\mathrm{PET}}\boldsymbol{G}$, where $\boldsymbol{G}'\boldsymbol{G}$ is approximately circulant (i.e., a space-invariant operator) and $\boldsymbol{D}_{\mathrm{PET}}$ is a diagonal matrix [1]. In this case, all object-dependence enters through the diagonal term, and $\boldsymbol{G}$ represents the geometric system model. Such factorizations are possible for idealized PET systems, and the methods in [1] allow one to precalculate many of the operations necessary for the evaluation of $\boldsymbol{H}'\boldsymbol{D}\boldsymbol{H}\underline{e}^j$. Due to attenuation, this kind of factorization is inappropriate for SPECT. Nevertheless, we can generalize the ideas used in [1] and apply them to the SPECT model. This section describes a series of approximations that allow many operations to be precomputed for the evaluation of $\boldsymbol{H}'\boldsymbol{D}\boldsymbol{H}\underline{e}^j$, when the system matrix fits a SPECT model. Similarly, these methods can be used for resolution and covariance prediction in shift-variant PET systems. In the special case of a space-invariant system and a PET-style attenuation model the results here simplify to the methods presented in [1].

A. Linear Operators

One important property of $\boldsymbol{H}'\boldsymbol{D}\boldsymbol{H}\underline{e}^j$ used in [1] is that it is linear in terms of the diagonal elements of $\boldsymbol{D}$. That is, we may write

$$\boldsymbol{H}'\boldsymbol{D}\boldsymbol{H}\underline{e}^j = \sum_{i=1}^{N} \underline{m}^j_i [\boldsymbol{D}]_{ii} = \boldsymbol{M}^j\underline{d} \quad (13)$$

where $\underline{m}^j_i$ are position-dependent vectors that are related to $\boldsymbol{H}$. Similarly, we may write this linear combination in terms of a $P \times N$ matrix, $\boldsymbol{M}^j = [\underline{m}^j_1 \ldots \underline{m}^j_N]$, and a vector of the diagonal elements of $\boldsymbol{D}$, which are denoted as $\underline{d}$ with $[\underline{d}]_i = [\boldsymbol{D}]_{ii}$.

One could construct $\boldsymbol{M}^j$ using the superposition principle. Specifically, $\underline{m}_i^j$ may be found by applying diagonalized unit vectors for each measurement such that

$$\underline{m}_i^j = \boldsymbol{H}' \mathrm{diag}\{\underline{e}^i\} \boldsymbol{H} \underline{e}^j. \tag{14}$$

In principle, if $\{\boldsymbol{M}^j\}_{j=1}^P$ could be precalculated, then $\boldsymbol{H}'\boldsymbol{D}\boldsymbol{H}\underline{e}^j$ can be evaluated for different diagonal matrices using about one half the computation of an ordinary evaluation. (Later, we will identify a series of approximations that rely on this precomputed form to drastically reduce computation time.) Unfortunately, there are several problems with this kind of precomputation. Perhaps the most significant problem is that, even if one were to calculate all $\{\boldsymbol{M}^j\}_{j=1}^P$ operators, these linear operators are object-dependent because the SPECT system matrix depends on the attenuation properties of the object. Thus, any such "precalculation" would need to be performed for every object. While one might be able to use a generic attenuation model in cases like brain imaging where there is less variability, we would like to develop an efficient technique that applies to a wide range of attenuating objects.

Another problem is the sheer size of $\{\boldsymbol{M}^j\}_{j=1}^P$. One must be able to store these precomputed linear operators to exploit any computational speed-up. Recall that each operator $\boldsymbol{M}^j$ is $P \times N$ in size. Generally it would be unfeasible to store all P operators since they have a similar degree of sparsity as the system matrix, $\boldsymbol{H}$.

We address these issues in the following sections.

B. Attenuation Approximations

To use the linear operator technique effectively we must eliminate the object-dependence from the precomputed portions of $\boldsymbol{H}'\boldsymbol{D}\boldsymbol{H}\underline{e}^j$. In SPECT, attenuation is typically modeled as both a pixel-dependent and ray-dependent effect[4] such that

$$\bar{Y}_i(\underline{\lambda}) = \sum_{j=1}^P b_i g_{ij} a_{ij} \lambda_j + r_i \tag{15}$$

where the g_{ij} terms incorporate the geometric response including all detector effects like the depth-dependent response. The b_i terms denote ray-dependent factors like uniformity correction factors to compensate for differences in sensitivity across the detector face. The a_{ij} terms incorporate the attenuation effects due to the object and generally reflect a "survival" probability for a photon emitted at the jth location and detected at the ith detector. These terms are often formed by approximating the attenuation using a central ray approximation in combination with Beer's Law

$$a_{ij} = \exp\left(\int_{\mathcal{L}_{ij}(x,y,z)} -\mu(x,y,z)\, dl \right) \tag{16}$$

where $\mu(x,y,z)$ denotes the attenuation coefficient distribution for the object and $\mathcal{L}_{ij}(x,y,z)$ represents the line segment connecting the jth pixel with the ith detector.

---

[4]For comparison, in PET, it is usual for attenuation effects to be modeled as a purely ray-dependent effect.

Thus, we can write a factorization of the SPECT system matrix as

$$\boldsymbol{H} = \boldsymbol{B}(\boldsymbol{A} \odot \boldsymbol{G}) \tag{17}$$

where $\boldsymbol{B}$ is a diagonal matrix of b_i terms and $\boldsymbol{A}$ and $\boldsymbol{G}$, are collections of the attenuation terms, a_{ij}, and geometric terms, g_{ij}, respectively. (We do *not* require that geometric operator $\boldsymbol{G}'\boldsymbol{G}$ be a shift-invariant operator.) This factorization isolates all of the object-dependence in the $\boldsymbol{A}$ term. Since the same factorization applies to PET systems with $\boldsymbol{A} = \boldsymbol{1}$ and with PET attenuation factors in $\boldsymbol{B}$, the following prediction techniques may also be used for space-variant PET systems.

Let $\boldsymbol{F} = \boldsymbol{H}'\boldsymbol{D}\boldsymbol{H}$ denote the entire weighted projection-backprojection operator. Using the factorization in (17), we make the following sequence of observations regarding the (k,j)th element of $\boldsymbol{F}$

$$\begin{aligned}
[\boldsymbol{F}]_{kj} &= (\underline{e}^k)' \boldsymbol{F} \underline{e}^j \\
&= (\underline{e}^k)' \boldsymbol{H}' \boldsymbol{D} \boldsymbol{H} \underline{e}^j \\
&= (\underline{e}^k)' [\boldsymbol{B}(\boldsymbol{A} \odot \boldsymbol{G})]' \, \boldsymbol{D} \, [\boldsymbol{B}(\boldsymbol{A} \odot \boldsymbol{G})] \, \underline{e}^j \\
&= [(\boldsymbol{A}\underline{e}^k)' \odot (\boldsymbol{G}\underline{e}^k)'] \, \boldsymbol{BDB} \, [(\boldsymbol{A}\underline{e}^j) \odot (\boldsymbol{G}\underline{e}^j)] \\
&= (\boldsymbol{G}\underline{e}^k)' \mathrm{diag}\{\boldsymbol{A}\underline{e}^k\} \boldsymbol{BDB} \,\mathrm{diag}\{\boldsymbol{A}\underline{e}^j\} \boldsymbol{G}\underline{e}^j \\
&= (\underline{e}^k)' \boldsymbol{G}' \boldsymbol{D}^{jk} \boldsymbol{G} \underline{e}^j
\end{aligned} \tag{18}$$

where the diagonal matrix, $\boldsymbol{D}^{jk}$, has the following elements:

$$\begin{aligned}
[\boldsymbol{D}^{jk}]_{ii} &= [\boldsymbol{A}\underline{e}^k]_i [\boldsymbol{D}]_{ii} [\boldsymbol{B}]_{ii}^2 [\boldsymbol{A}\underline{e}^j]_i \\
&= b_i^2 a_{ij} a_{ik} [\boldsymbol{D}]_{ii}.
\end{aligned} \tag{19}$$

The equalities in (18) are the key to the method developed in this paper.

Because $\boldsymbol{A}\underline{e}^k$ generally varies relatively smoothly with changing k and $\boldsymbol{H}'\boldsymbol{D}\boldsymbol{H}\underline{e}^j$ is fairly concentrated about the pixel position j, we utilize (18) and (19) to make the following approximation:

$$\boldsymbol{F}\underline{e}^j = \boldsymbol{H}'\boldsymbol{D}\boldsymbol{H}\underline{e}^j \approx \boldsymbol{G}'\boldsymbol{D}^j \boldsymbol{G}\underline{e}^j \tag{20}$$

with elements of the diagonal matrix, $\boldsymbol{D}^j$, defined as

$$[\boldsymbol{D}^j]_{ii} = [\boldsymbol{D}^{jj}]_{ii} = b_i^2 a_{ij}^2 [\boldsymbol{D}]_{ii}. \tag{21}$$

Thus, we approximate $\boldsymbol{H}'\boldsymbol{D}\boldsymbol{H}\underline{e}^j$ using the geometric model $\boldsymbol{G}$ and a position-dependent diagonal weighting $\boldsymbol{D}^j$. This approximation is exact at location j and yields very good results for the neighborhood around j. Because the attenuation terms, $\{a_{ij}\}$, are formed from the integral in (16), there is an implicit smoothing in going from an attenuation map to $\{a_{ij}\}$. Thus, the above approximation tends to work well even for discontinuous attenuation maps.

The approximation for $\boldsymbol{H}'\boldsymbol{D}\boldsymbol{H}\underline{e}^j$ shown in (20) is important since all object-dependence enters through the "inner" diagonal term. In contrast, the approximation methods for 2-D and 3-D PET described in [1], [6], [11], and [15] move the object-dependence to the "outside" of the equation such that $\boldsymbol{H}'\boldsymbol{D}\boldsymbol{H}\underline{e}^j \approx \boldsymbol{D}^*\boldsymbol{G}'\boldsymbol{G}\boldsymbol{D}^*\underline{e}^j$, where $\boldsymbol{D}^*$ is a diagonal term that incorporates object-dependence. Xing [12], [13] has used a variation on these "outer" diagonal approximations to make weighted projection-backprojection estimates for shift-variant SPECT systems.

Mechanism	Approximation/Exact	Purpose
Linear Operators	Exact	Speedup via precalculation
SPECT Attenuation Approx.	Approximation	Ability to use precomputed operators in SPECT
Single Slice Sampling	Nearly Exact (most systems)	Storage: $1/P_z$ Computation: same
Partial Orbital Sampling	Exact (symmetric orbits)	Storage: $1/4$ Computation: slight increase
Small Support Volume	Approximation	Storage: η^3/P Computation: η^3/P
Spatial Subsampling	Approximation	Storage: $1/n_d^2$ Computation: $1/n_d^2$ + fast interp.
Projection Constant Weights	Approximation	Storage: n_a/N Computation: n_a/N + eq. (31) calcs.
Angular Subsampling	Approximation	Storage: $1/n_s$ Computation: $1/n_s$ + eq. (34) calcs.

*All stated storage and computational reductions are relative to the case of unapproximated precomputed linear operators in (13).

Since the righthand side of (20) is a linear function of the diagonal elements of $\boldsymbol{D}^j$, we may now calculate approximate precomputed operators, $\boldsymbol{M}^j$, whose columns are given by

$$\underline{m}_i^j = \boldsymbol{G}'\text{diag}\{\underline{e}^i\}\boldsymbol{G}\underline{e}^j. \qquad (22)$$

Since these $\underline{m}_i^j$ vectors depend only on the system geometry $\boldsymbol{G}$, but not on the object itself, we can precompute the object-independent portion of $\boldsymbol{H}'\boldsymbol{DH}\underline{e}^j$. These operators may be applied to form the approximation given in (20) as

$$\boldsymbol{H}'\boldsymbol{DH}\underline{e}^j \approx \boldsymbol{M}^j\underline{d}^j \qquad (23)$$

where $\underline{d}^j$ is a vector constructed from the elements of $\boldsymbol{D}^j$ in (21).

It is reasonable to include a_{ij} terms in the calculation of $\underline{d}^j$, since these factors generally must be computed for the reconstruction method that is chosen to estimate the SPECT image. In fact, while $\boldsymbol{G}$ is often too large to precompute and store for 3-D-SPECT, if $\boldsymbol{A}$ is modeled with the central ray approximation and the simple line integral model of (16), $\boldsymbol{A}$ is very sparse with only a single value per voxel per projection angle. Thus in some situations, it may be possible to compute and store $\boldsymbol{A}$ for a given object for both estimation of the SPECT image, and for evaluation of $\boldsymbol{H}'\boldsymbol{DH}\underline{e}^j$. (This storage is not required for our technique. The a_{ij} terms may instead be computed on-the-fly.)

Equation (23) represents an approximation that allows for precomputation of a portion of $\boldsymbol{H}'\boldsymbol{DH}\underline{e}^j$ using the linear operator technique of [1]. However, unlike the shift-invariant PET case described in [1], because the geometric response is space-variant for SPECT, it appears that one needs to calculate very many linear operators. Specifically, without further simplifications, one would need to compute, store, and use one $P \times N$ matrix for each voxel. In the following sections we demonstrate ways to reduce both storage requirements and computation time.

C. Image-Domain Simplifications

There are a number of observations and approximations that allow us to reduce the computation and storage requirements to practical levels. We break these simplifications into two groups: 1) Image-domain simplifications, that reduce either the number of operators that are stored, or the number of rows in each of the matrices. 2) Projection-domain simplifications, that reduce the number of columns required for each $\boldsymbol{M}^j$, and consequently the number of diagonal weighting elements (i.e., a smaller $\boldsymbol{D}$). We discuss the image-domain simplifications in this section, and discuss projection-domain simplifications in Section III-D.

For each approximation, we first describe the basic principle in words, and then give an explicit mathematical representation. Since matrices in the following sections represent operations on 3-D projections or images, care should be taken in interpreting the mathematical forms. A summary of simplifications and approximations can be found in Table I.

1) Single Slice Sampling: Because $\boldsymbol{H}$ is object-dependent due to attenuation, there are generally few symmetries that would allow one to reduce computation and storage requirements. However, because we are utilizing (22), which requires only the geometric model, $\boldsymbol{G}$, we can take advantage of symmetries in the SPECT geometry.

For many SPECT systems there are a number of symmetries in the imaging system that can simplify our goals. For most parallel and fan collimators, the detector response is essentially shift-invariant for axial shifts of the detector, excluding magnitude scaling factors [i.e., the b_i terms in (15)]. Thus, if one varies j only in the axial direction, $\boldsymbol{G}'\boldsymbol{G}\underline{e}^j$ only changes by a axial shift (with the exception of truncation effects at the edge of the field of view). Similarly, for the same j, the columns of our precalculated $\boldsymbol{M}^j$ in (22) would only differ by axial shifts.

Therefore it is not necessary to compute (22) for all j. A single slice is sufficient. Thus, we let

$$\underline{m}_i^{(x_j,y_j)} = \boldsymbol{G}'\text{diag}\{\underline{e}^i\}\boldsymbol{G}\underline{e}^{(x_j,y_j,z_0)} \qquad (24)$$

where x_j and y_j denote the x and y-coordinates of the jth voxel, and z_0 reflects the axial coordinate of the center slice. Consequently

$$\boldsymbol{H}'\boldsymbol{DH}\underline{e}^j \approx \boldsymbol{S}^{z_j}\boldsymbol{M}^{(x_j,y_j)}\left[\boldsymbol{S}_P^{z_j}\right]^{-1}\underline{d}^j \qquad (25)$$

where $\boldsymbol{S}^{z_j}$ shifts an image from the center slice to the z-coordinate of the j voxel, $\boldsymbol{M}^{(x_j,y_j)}$ is formed from columns of (24), and $\boldsymbol{S}_P^{z_j}$ is the projection domain analogue of $\boldsymbol{S}^{z_j}$, which shifts projection values along the axial direction. In terms of storage, we may now store $P_x \cdot P_y$ operators instead of $P = P_x \cdot P_y \cdot P_z$.

2) Partial Orbital Sampling: We can also take advantage of symmetries in the SPECT detector orbit. Consider the 360° elliptical orbit SPECT system shown in Fig. 1. Suppose that we may only compute weighted projection-backprojections for points in quadrant IV. If one has obtained the projections for the black point in quadrant IV, one can obtain the projections for the gray point in quadrant I simply by reordering the projection images. Similarly, if one may only backproject projections obtained from points in quadrant IV, one can obtain pro-
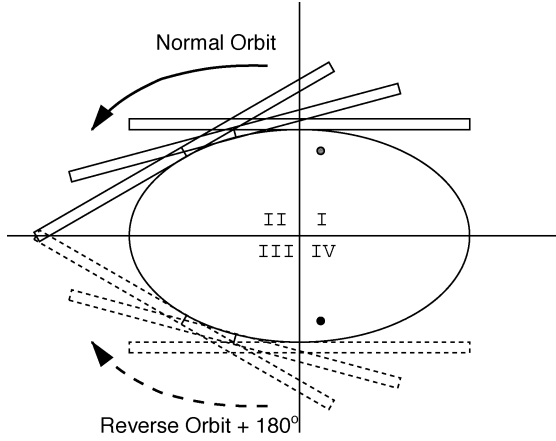
Fig. 1.   Symmetries in elliptical orbit SPECT.

jection-backprojections for points in the other quadrants using simple flips about the axes. In this way, we need to precompute only a single quadrant of linear operators. (For circular system geometries, only a single radial line of operators is required.)

While such symmetries can reduce storage requirements, position-dependent change of coordinates operations are required to calculate responses at all positions. (For example, there is a quadrant dependence for these operations for the elliptical orbit in Fig. 1.) For a system geometry with generic symmetries, we can represent the projection-domain reordering operation for the jth voxel by P^j, and the image-domain change of variables for the jth voxel by V^j. Incorporating this calculation technique into (25), we write the approximation of the weighted response as

$$H'DH\underline{e}^j \approx V^j S^{z_j} M^{(x_j,y_j)} \left[S_P^{z_j}\right]^{-1} P^j \underline{d}^j \qquad (26)$$

where the $M^{(x_j,y_j)}$ operators are still calculated via (24), but only over a subset of locations appropriate for the specific system symmetries.

3) Small Volume of Support: Because $H'DH\underline{e}^j$ is fairly concentrated about voxel j, many calculations involving $H'DH\underline{e}^j$ are also very concentrated about j. Specifically, the resolution and covariance functions generally go to zero far from j, and can be well-approximated using relatively small regions of support [7].

Thus, it is not necessary to store all the rows of M^j. Choosing a small $\eta \times \eta \times \eta$ volume[5] about voxel j, reduces the dimension of each M^j to $\eta^3 \times N$. For a typical SPECT system where $P = 128^2 \cdot 64$, a choice of $\eta = 30$ represents a decrease in storage by a factor of almost 40.

Thus, (24) and (26) become

$$\underline{m}_i^{(x_j,y_j)} = T^j G' \mathrm{diag}\{\underline{e}^i\} G\underline{e}^{(x_j,y_j,z_0)} \qquad (27)$$

and

$$T^j H'DH\underline{e}^j \approx V^j M^{(x_j,y_j)} \left[S_P^{z_j}\right]^{-1} P^j \underline{d}^j \qquad (28)$$

where T^j represents a position-dependent $\eta^3 \times P$ matrix that represents a truncation function that selects a small volume about pixel j. The image-domain shift operation in (25) is no

[5]There is no fundamental reason why the subvolumes must be cubes. We choose a cubical subvolume for simplicity.

longer necessary due to the truncation function T^j, since there is an implicit "centering" of the subvolume. Moreover, we note that V^j now operates on $\eta \times \eta \times \eta$ subvolumes instead of the entire image volume.

4) Spatial Subsampling: The weighted responses, $H'DH\underline{e}^j$, typically vary smoothly with position. Because this is the case, we have found that one can subsample the image domain over a grid of every n_dth voxel and evaluate $H'DH\underline{e}^j$ over a subset of positions and find the remaining positions using interpolation [7]. Using the single slice approximation, this reduces storage requirements by $1/n_d^2$.

D. Projection-Domain Simplifications

Just as one can approximate $H'DH\underline{e}^j$ using image-domain simplifications, one can make projection-domain approximations that reduce dimensionality, storage requirements, and computation times. Specifically, in the following subsections, we describe approximations that will reduce the number of columns required for the linear operators, M^j.

1) Projection-Constant Weightings: One approximation investigated in [1] relies on the observation that projections of a point are highly localized. That is, for individual projection angles, $H\underline{e}^j$ yields a relatively narrow response. Fig. 2(a) shows several projections of a point. The diagonal term, D, simply scales each element of the projection and is typically a smoothly-varying function over each projection. Recall from (6) and (7), elements of D are often defined as functions of the mean measurements, which are themselves relatively smooth due to the blur of the projection operator. Because these weightings are relatively smooth for each projection angle and the point projections are localized, we can approximate D with a new diagonal weighting which scales projections for individual angles by a single value.

Let $[H]'_{(i,1:P)}$ denote the ith row of H, and $\mathcal{P}_a$ denote the set of measurements in the projection at angle a. We make the approximation

$$H'DH\underline{e}^j \approx \sum_{a=1}^{n_a} \sum_{i \in \mathcal{P}_a} [H]'_{(i,1:P)} [\tilde{d}^j]_a [H]_{(i,1:P)} \underline{e}^j$$

$$= H' \mathrm{diag}\left\{C'_{\mathcal{P}} \underline{\tilde{d}}^j\right\} H\underline{e}^j$$

$$= H' \tilde{D}^j H\underline{e}^j \qquad (29)$$

where $[\tilde{d}^j]_a$ represents the position-dependent, projection-constant weighting for the ath angle, and the vector $\underline{\tilde{d}}^j$ denotes the collection of all projection-constant weightings over all n_a angles. The $n_a \times N$ matrix $C_{\mathcal{P}}$ combines measurements within a single projection angle into a single value, and is used to form the new diagonal matrix, $\tilde{D}^j$. This combination matrix can be written as

$$C_{\mathcal{P}} = \left[\underline{1}_{\mathcal{P}_1} \cdots \underline{1}_{\mathcal{P}_{n_a}}\right]' \qquad (30)$$

where $\underline{1}_{\mathcal{P}_a}$ is an indicator vector where the ith element of the vector is one if the element belongs in the projection at angle a, and is zero otherwise.

While there are many ways to calculate $\underline{\tilde{d}}^j$, elements of this vector can generally be approximated by some form of position-
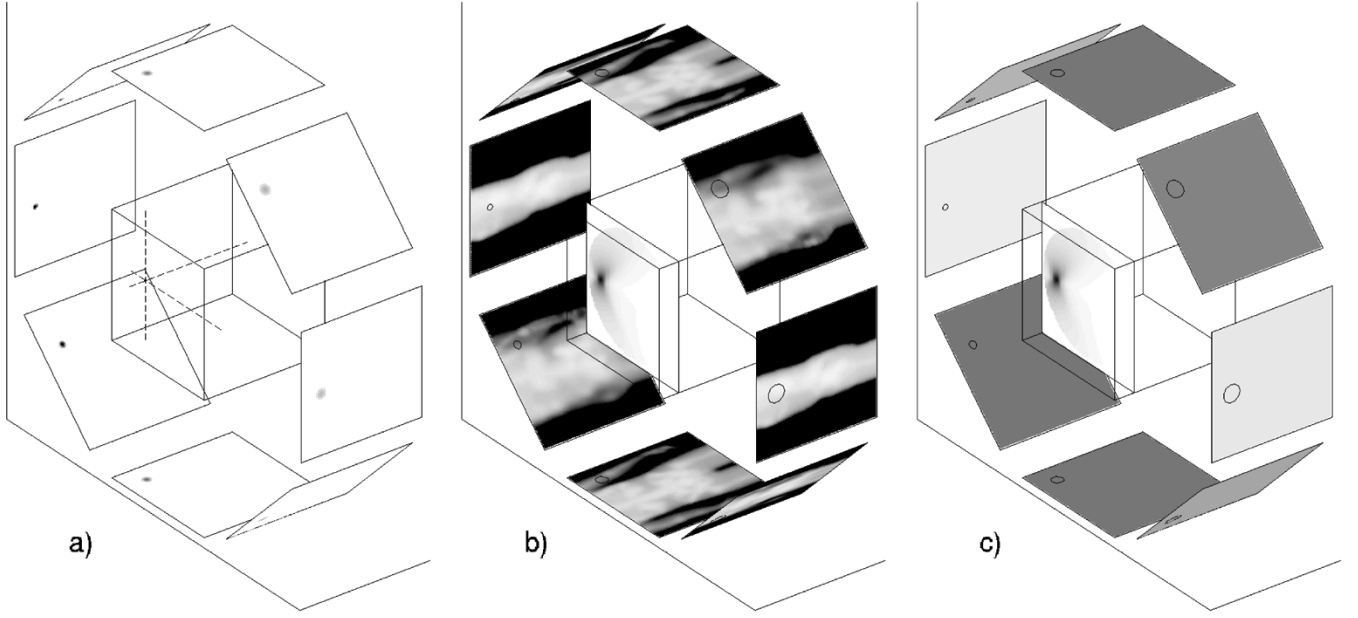
Fig. 2. Approximation of the weighted point projection-backprojection, $\boldsymbol{H}'\boldsymbol{D}\boldsymbol{H}\underline{e}^j$, using a projection-constant weighting. (a) Shows a particular point within the imaging volume and a few of its projections, $\boldsymbol{H}\underline{e}^j$. (b) Shows several projection weightings in the diagonal weighting $\boldsymbol{D}$, and a cross section of the associated weighted response, $\boldsymbol{H}'\boldsymbol{D}\boldsymbol{H}\underline{e}^j$ (shown at the center). We identify the approximate positions of the point projection using a small black circle in each projection. Because the point projections are highly localized, we may approximate the projection weighting using a position-dependent projection-constant weighting, $\bar{\boldsymbol{D}}^j$, shown in (c). The associated weighted response, $\boldsymbol{H}'\boldsymbol{C}'_{\mathcal{P}}\bar{\boldsymbol{D}}^j\boldsymbol{C}_{\mathcal{P}}\boldsymbol{H}\underline{e}^j$, is nearly identical to the unapproximated response, $\boldsymbol{H}'\boldsymbol{D}\boldsymbol{H}\underline{e}^j$.

dependent weighted average. For example, one simple technique that weighs elements of $\boldsymbol{D}$ by the intensity of a point projection is

$$[\tilde{\underline{d}}^j]_a = \frac{\underline{I}'_{\mathcal{P}_a}\boldsymbol{D}\hat{\boldsymbol{H}}\underline{e}^j}{\underline{I}'_{\mathcal{P}_a}\hat{\boldsymbol{H}}\underline{e}^j} \tag{31}$$

where $\hat{\boldsymbol{H}}$ is some form of the system matrix, $\boldsymbol{H}$. Because we have found that the approximation in (29) is relatively insensitive to the exact weightings, it is often sufficient to use an approximate $\hat{\boldsymbol{H}}$. In fact, we find using a simple line integral model without attenuation is often sufficient for $\hat{\boldsymbol{H}}$. Thus, it is straightforward to precompute and store the necessary weightings to compute $\tilde{\boldsymbol{D}}^j$.

Fig. 2 demonstrates the efficacy of this technique. Fig. 2(a) shows a few unweighted projections of a single point. Fig. 2(b) shows sample projection weights and a transaxial cross section of the associated weighted response. Approximate positions of the point projection are indicated with small black circles. We find an approximate projection-constant weighting based on (31), with $\hat{\boldsymbol{H}}$ equal to a simple line integral model with no attenuation. Thus (31) is simply a bilinear interpolation for each projection. (We suspect that even simple nearest-neighborhood interpolation would also be adequate.) Fig. 2(c) shows the projection-constant weights and a cross section of the weighted response. The two transaxial cross sections are nearly indistinguishable.

Before we discuss the resulting linear operator form of the approximation, we discuss one additional approximation that further reduces the the size of the diagonal weighting.

2) Angular Subsampling: Rather than computing the projections, $\boldsymbol{H}\underline{e}^j$, over all angles, we further approximate the projection (and backprojection) by reducing the number of projection angles involved. We will divide projection angles into K contiguous blocks, where a single block combines a neighborhood of n_s angles. Letting $\mathcal{S}_k$ denote the set of angles belonging to the kth block, we write

$$\boldsymbol{H}'\boldsymbol{D}\boldsymbol{H}\underline{e}^j \approx \sum_{k=1}^{K}\sum_{a\in\mathcal{S}_k}\sum_{i\in\mathcal{P}_a}[H]'_{(i,1:P)}[\check{\underline{d}}^j]_k[H]_{(i,1:P)}\underline{e}^j$$
$$= \boldsymbol{H}'\text{diag}\left\{\boldsymbol{C}'_{\mathcal{S}}\boldsymbol{C}'_{\mathcal{P}}\check{\underline{d}}^j\right\}\boldsymbol{H}\underline{e}^j$$
$$= \boldsymbol{H}'\text{diag}\left\{\boldsymbol{C}'_{\mathcal{PS}}\check{\underline{d}}^j\right\}\boldsymbol{H}\underline{e}^j$$
$$= \boldsymbol{H}\check{\boldsymbol{D}}^j\boldsymbol{H}\underline{e}^j, \tag{32}$$

where the combination matrix is defined as

$$\boldsymbol{C}_{\mathcal{S}} = \left[\underline{I}_{\mathcal{S}_1} \cdots \underline{I}_{\mathcal{S}_K}\right]'. \tag{33}$$

where the indicator vector, $\underline{I}_{\mathcal{S}_k}$, indicates membership of an angle in the set $\mathcal{S}_k$. We also define $\boldsymbol{C}_{\mathcal{PS}} \triangleq \boldsymbol{C}_{\mathcal{P}}\boldsymbol{C}_{\mathcal{S}}$.

We choose approximate position-dependent weights, $\check{\underline{d}}^j$, by simply averaging over angles in each set, $\mathcal{S}_k$. Specifically

$$[\check{\underline{d}}^j]_{kk} = \frac{1}{n_s}\sum_{a\in\mathcal{S}_k}\frac{\underline{I}'_{\mathcal{P}_a}\boldsymbol{D}\hat{\boldsymbol{H}}\underline{e}^j}{\underline{I}'_{\mathcal{P}_a}\hat{\boldsymbol{H}}\underline{e}^j}. \tag{34}$$

The diagonal matrix, $\check{\boldsymbol{D}}^j$, represents a significant decrease in the dimension from the original weighting, $\boldsymbol{D}$. Recall that $\boldsymbol{D}$ is $N \times N$, where N is the product of the number of measurements per projection (i.e., the number of pixels in each projection), and the number of projection angles, n_a. In comparison, $\check{\boldsymbol{D}}^j$ is

$K \times K$, where K is the number of projection angles, n_a, divided by the number of angles in each subset, n_s.

E. Simplified Linear Operators

We now combine the simplifications discussed in the previous sections to obtain a set of linear operators that is practical to implement and store.

Section III-D discussed two approximations that reduce the dimension of $\boldsymbol{D}$ from N to $K = n_a/n_s$. We may calculate the reduced dimension linear operator by applying the approximations in Section III-D to (27) to obtain

$$\underline{m}_k^{(x_j, y_j)} = \boldsymbol{T}^j \boldsymbol{G}' \mathrm{diag} \left\{ \boldsymbol{C}'_{\mathcal{PS}} \underline{e}^k \right\} \boldsymbol{G} \underline{e}^{(x_j, y_j, z_0)}. \quad (35)$$

For each position where (35) is evaluated a single point projection must be performed, followed by n_s (truncated) partial backprojections, which collectively have the same complexity as a single (truncated) backprojection.

The projection-constant weighting discussed in Section III-D-1 eliminates the need for the projection-domain shift operation introduced in (25). Thus, we may now write[6]

$$\boldsymbol{T}^j \boldsymbol{H}' \boldsymbol{D} \boldsymbol{H} \underline{e}^j \approx \boldsymbol{V}^j \boldsymbol{M}^{(x_j, y_j)} \boldsymbol{P}^j \underline{\check{d}}^j. \quad (36)$$

The vector $\underline{\check{d}}^j$ is formed from joining (34) with the attenuation approximation in (21). Specifically, the kth element of $\underline{\check{d}}^j$ is

$$[\underline{\check{d}}^j]_k = \frac{1}{n_s} \sum_{a \in \mathcal{S}_k} \frac{\underline{L}'_{\mathcal{P}_a} \left[\boldsymbol{B}^2 \mathrm{diag}\{\boldsymbol{A}\underline{e}^j\}^2 \boldsymbol{D} \right] \hat{\boldsymbol{H}} \underline{e}^j}{\underline{L}'_{\mathcal{P}_a} \hat{\boldsymbol{H}} \underline{e}^j}. \quad (37)$$

In terms of storage, we now have matrices, $\boldsymbol{M}^{(x_j, y_j)}$, that are $\eta^3 \times K$. From Section III-C, we need to store these matrices within only a single slice, or a single-quadrant of a single slice for orbits with two-fold symmetries. We may further subsample this quadrant to reduce computational costs. Thus for elliptical orbit SPECT, using all these simplifications in conjunction means we must store

$$\frac{1}{4} \frac{P_x P_y}{n_d^2} \eta^3 \frac{n_a}{n_s} \quad (38)$$

floating point numbers. Consider a sample SPECT system that incorporates a $128 \times 128 \times 64$ image volume and projections over 110 angles. For a sampling of every fourth image pixel in x and y, a subvolume of $30 \times 30 \times 30$, and blocks of 10 angles, we must store about 76 million floating point numbers. If stored as standard single precision floating point numbers, this represents about 290 Mb of storage space.

All of the computation and storage reductions we have discussed are summarized in Table I. This table specified whether or not a given technique is exact or an approximation and the level of reduction in computations and/or storage.

Equations (35), (36), and (37) represent a set of precomputations and the necessary operations for approximating $\boldsymbol{H}' \boldsymbol{D} \boldsymbol{H} \underline{e}^j$. While this weighted projection-backprojection may be of interest for some applications, additional simplifications can be made when resolution or covariance prediction is the goal. The following section discusses such simplifications.

---

[6]Note that the P^j operator has the same function as was described in Section III-C-2, but now operates on the smaller vector, $\underline{\check{d}}^j$, which contains projection weights for blocks of angles. Similarly, $\boldsymbol{V}^j$ now operates on $\eta \times \eta \times \eta$ subvolumes.

IV. ADDITIONAL SIMPLIFICATIONS FOR RESOLUTION AND COVARIANCE PREDICTION

To predict resolution or covariance, one can plug the approximation of the weighted projection-backprojection in (36) directly into the resolution or covariance predictors in (8) and (12). However, further investigation allows us to make additional simplifications that reduce storage and computation time.

Both (8) and (12) are based on using a circulant approximation to $\boldsymbol{H}' \boldsymbol{D} \boldsymbol{H}$. Because circulant matrices can be diagonalized using Fourier bases, we may find the eigenvalues of the circulant approximation using Fourier transforms, which allows one to avoid the full matrix inverse computations in (5) and (11). Because $\bar{\boldsymbol{D}}$ is a diagonal matrix composed of nonnegative elements, the eigenvalues of $\boldsymbol{H}' \boldsymbol{D} \boldsymbol{H}$ are necessarily real and nonnegative. It is common to enforce these constraints when Fourier transforming the weighted response $\boldsymbol{H}' \boldsymbol{D} \boldsymbol{H} \underline{e}^j$. The real constraint is typically enforced by ensuring point symmetry through the center of the response[7] (i.e., voxel j). An equivalent approach is to only use the real part of the Fourier transformed image. The nonnegativity constraint is often enforced simply by zeroing any negative components. The same constraints are applied to the penalty terms in (8) and (12). Thus, the resolution and covariance predictors may be written as follows:

$$\underline{l}_{\mathrm{circ}}^j = \mathcal{F}^{-1} \left\{ \frac{\mathcal{F}\{\underline{e}^j\} \odot \underline{\tilde{f}}_2^j}{\underline{\tilde{f}}_1^j + \underline{\tilde{r}}^j} \right\}$$

$$\mathrm{Cov}_{\mathrm{circ}}^j = \mathcal{F}^{-1} \left\{ \frac{\mathcal{F}\{\underline{e}^j\} \odot \underline{\tilde{f}}_3^j}{\left| \underline{\tilde{f}}_1^j + \underline{\tilde{r}}^j \right|^2} \right\} \quad (39)$$

where

$$\underline{\tilde{f}}_n^j \triangleq \max \left\{ \mathrm{re} \left\{ \frac{\mathcal{F}\{\boldsymbol{H}' \boldsymbol{D}_n \boldsymbol{H} \underline{e}^j\}}{\mathcal{F}\{\underline{e}^j\}} \right\}, 0 \right\} \quad (40)$$

$$\underline{\tilde{r}}^j \triangleq \max \left\{ \mathrm{re} \left\{ \frac{\mathcal{F}\{\boldsymbol{R}(\underline{\check{\lambda}})\underline{e}^j\}}{\mathcal{F}\{\underline{e}^j\}} \right\}, 0 \right\} \quad (41)$$

and the $\mathcal{F}\{\underline{e}^j\}$ are applied to shift the local impulse response or covariance measurement to the jth voxel. (Equivalently, this may be applied as an image-domain shifting operation.)

Since the approximation to $\boldsymbol{H}' \boldsymbol{D} \boldsymbol{H} \underline{e}^j$ discussed in Section III is eventually plugged into the above expressions, it would be advantageous to include as many of the operations in (40) in the precomputation step as possible. Because the Fourier transform is a linear operation, it is natural to incorporate these operations in $\boldsymbol{M}^j$ as well. Specifically, we may now redefine the operators specified in (35) as

$$\underline{m}_k^{(x_j, y_j)} = \mathrm{re} \left\{ \mathcal{F} \left\{ \boldsymbol{T}^j \boldsymbol{G}' \mathrm{diag} \left\{ \boldsymbol{C}'_{\mathcal{PS}} \underline{e}^k \right\} \boldsymbol{G} \underline{e}^{(x_j, y_j, z_0)} \right\} \right\}. \quad (42)$$

Noting that the change of coordinates represented by $\boldsymbol{V}^j$ is invertible, approximation (36) becomes

$$\mathrm{re} \left\{ \mathcal{F} \left\{ (\boldsymbol{V}^j)^{-1} \boldsymbol{T}^j \boldsymbol{H}' \boldsymbol{D} \boldsymbol{H} \underline{e}^j \right\} \right\} \approx \boldsymbol{M}^{(x_j, y_j)} \boldsymbol{P}^j \underline{\check{d}}^j. \quad (43)$$

---

[7]In general, SPECT responses can be asymmetric. Future work should include investigations of how best to handle cases such as 180° SPECT where responses can be highly asymmetric.

The transformation $(\boldsymbol{V}^j)^{-1}$ appears inside the Fourier transform, which appears to complicate our task. Fortunately, because the transformation $\boldsymbol{V}^j$ is only a renaming of image coordinates, we may apply the transformation in either image domain. That is, either before the $\mathcal{F}$ operation or after the $\mathcal{F}^{-1}$ operation. Therefore we may rewrite the circulant approximation to the predictors as

$$\underline{l}_{\mathrm{circ}}^j \approx \boldsymbol{V}^j \mathcal{F}^{-1} \left\{ \frac{\check{\underline{f}}_2^j}{\check{\underline{f}}_1^j + \check{r}^j} \right\} \tag{44}$$

$$\mathrm{Cov}_{\mathrm{circ}}^j \approx \boldsymbol{V}^j \mathcal{F}^{-1} \left\{ \frac{\check{\underline{f}}_3^j}{\left|\check{\underline{f}}_1^j + \check{r}^j\right|^2} \right\} \tag{45}$$

with

$$\check{\underline{f}}_n^j \triangleq \max\left\{ \boldsymbol{M}^{(x_j,y_j)} \boldsymbol{P}^j \underline{\check{d}}^j, 0 \right\} \tag{46}$$

$$\check{r}^j \triangleq \max\left\{ \mathrm{re}\left\{ \mathcal{F}\left\{ (\boldsymbol{V}^j)^{-1} \boldsymbol{T}^j \boldsymbol{R}(\check{\underline{\lambda}}) \underline{e}^j \right\} \right\}, 0 \right\}. \tag{47}$$

Because of the truncation operators, $\boldsymbol{T}^j$, in (42) and (47), there is an implicit "centering" about location j and the $\mathcal{F}\{\underline{e}^j\}$ terms of (39) are no longer needed. Consequently, the predicted local impulse response in (44), and the covariance prediction in (45) are evaluated over a smaller support defined by $\boldsymbol{T}^j$. Thus, in order to form even an approximate equality with (8) and (12), these small support approximations must be embedded into the larger image space. [We have ignored this embedding in (44) and (45).]

We have found that $\boldsymbol{M}^{(x_j,y_j)}$ generally contains negative values that are important for prediction. Thus we cannot apply the negative thresholding in the precomputation step. It must be applied after the operator is applied to $\boldsymbol{P}^j \underline{d}^j$, as shown in (46).

Equations (44)–(47) represent the final form of the approximate predictors developed in this paper. These predictors require storage of a set of matrices, $\{\boldsymbol{M}^{(x_j,y_j)}\}$, which consist of

$$\frac{1}{4} \frac{P_x P_y}{n_d^2} \left(\frac{\eta}{2} + 1 \right) \eta^2 \frac{n_a}{n_s} \tag{48}$$

floating point numbers. The storage requirements are roughly one-half of that which is stated in (38) since the Fourier transform of a real signal results in coefficients whose real part is symmetric.

Once the linear operators have been precomputed, the following set of calculations are required for resolution and covariance prediction: 1) The $\underline{\check{d}}^j$ term is calculated via (37). Using a simple line integral model requires approximately $25n_a$ floating point operations (flops). 2) Calculation of (46), which takes about $2\eta^3 n_a/n_s$ due to the application of the linear operator. [We concentrate on the case when $\check{\underline{f}}_1^j = \check{\underline{f}}_2^j = \check{\underline{f}}_3^j$, which is a realistic assumption[8] for most SPECT systems.] 3) Computation of the resolution or covariance prediction using (44) and (45), respectively. This entails a single inverse Fourier transform plus roughly $2\eta^3$ flops for a local resolution prediction and $3\eta^3$ flops for a local covariance estimate.

In many cases (47) can be computed once, such as elliptical orbit systems with penalties for which $\boldsymbol{T}^j \boldsymbol{R}(\check{\underline{\lambda}}) \underline{e}^j$ exhibits three-fold planar symmetry[9] across each coordinate axis. For example, such is the case if the penalty is isotropic. For anisotropic penalties, one can decompose the penalty into symmetric and asymmetric portions, which can be formed from a small set of bases precomputed from $\check{r}^j$ terms. Thus, (47) generally involves relatively little computation.

The remaining computation is in applying a linear operator and a single $\eta \times \eta \times \eta$ inverse Fourier transform for each position j of interest. In comparison, recall the original expressions for the predictors in Section II, which require multiple $P_x \times P_y \times P_z$ Fourier transform operations, a point projection, and a full backprojection for every position.

For some prediction tasks, even the single inverse Fourier transform may be eliminated. For example, for variance prediction one needs only to calculate the peak of the covariance function. Thus, one can eliminate both leading transform operations in (45), and simply sum over the η^3 Fourier coefficients and perform an appropriate normalization. Similar simplifications can be made to (44) for the contrast recovery coefficient studied in [8].

V. RESULTS

This section illustrates the predictors discussed in Section IV. We compare the performance of our resolution and covariance predictors versus more traditional predictors and estimators on a simulated fully 3-D SPECT system.

A. SPECT System and Object Model

The SPECT model includes 128 projection angles and 128 $\times$ 30 pixel projection views with 4.5-mm pixels. The image volume is discretized into $128 \times 128 \times 30$ voxels, where each voxel is a 4.5-mm cube.[10] The SPECT camera follows an elliptical orbit with a 283-mm radius on the x axis and a 220-mm radius on the y axis. The SPECT detector model includes a depth-dependent Gaussian response that is 1.75-mm full-width at half-maximum (FWHM) at the face of the collimator and increases linearly with a slope of 0.044 as the distance to the collimator is increased. When the camera aims along the x axis, this slope corresponds to a FWHM of about 14.2 mm at the center of the field of view.

We chose to simulate a $^{99\mathrm{m}}\mathrm{Tc}$ bone scan using the Zubal phantom [16], [17]. We modified this digital phantom to include an attenuating patient bed and resampled the data onto a 4.5-mm grid. Fig. 3 displays this phantom data. We assigned relative emission rates of 3.0 to the spine, rib cage, and kidneys, 1.5 to the long bones in the arms, 3.0 to the long bone marrow, and 0.5 to the remaining soft tissue background. The attenuation map used attenuation coefficients appropriate for 140-keV photons with 0.23 cm^{-1} for bone, 0.15 cm^{-1} for all soft tissues, and 0.18 cm^{-1} for the table.

---

[8]Moreover, we have concentrated on a Poisson noise model which is appropriate for SPECT. However, these fast methods may be applicable to other systems with similar forms of shift-variance under other noise models as well.

[9]It is important not to confuse the orbital symmetries associated with the $\boldsymbol{V}^j$ transformations, with the point symmetry through the origin, which is imposed by the real constraint on the Fourier coefficients.

[10]The fast methods presented here do not required matched sizes for image voxels and projection pixels.
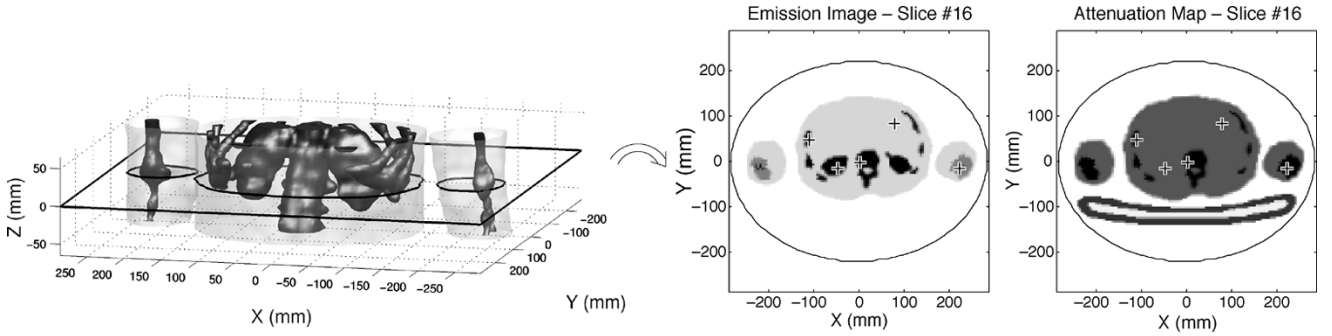
Fig. 3. The above figure shows the 3-D digital phantom used in the simulation studies. The anthropomorphic phantom simulates a ^{99m}Tc bone scan, with high activity in the bones and kidneys. A central slice of the emission image and the attenuation map is shown in the two images at the left. The orbit of the SPECT camera is indicated by the black ellipse. Additionally, five positions are indicated with $+$ marks for the investigation of resolution and covariance predictors.

We generated simulated SPECT measurements from the above phantom and system model. All studies used pseudo-random Poisson measurement data with a mean of $500\,000$ counts per slice, including a 20% known uniform background level [the r_i terms in (1)] to approximate the effects of scatter.

B. Reconstruction

We applied the penalized-likelihood estimator in (2) for reconstructing the emission images from the measurement data. The penalized-likelihood objective was maximized using an ordered-subsets paraboloidal surrogates iterative approach [18]–[20]. The algorithm was initialized with a filtered back-projection reconstruction. Following many iterations using 16 subsets, we applied convergent single subset iterations, to ensure a nearly converged solution For the penalty function we use a shift-invariant first-order quadratic penalty with the regularization parameter chosen to yield a spatial resolution of about 2 cm at the center of the field of view. For this penalty, the resolution at the edge of the object was about 4.5 mm. The reconstruction model matches the projection model exactly and used the true attenuation map.

C. Resolution Prediction

For the above SPECT system with Poisson measurements, the local impulse response of the penalized-likelihood estimator is given in (5) with diagonal components[11]

$$\boldsymbol{D}_1 = \boldsymbol{D}_2 = \text{diag}\left\{\frac{1}{\overline{Y}_i(\underline{\lambda})}\right\}. \tag{49}$$

The "traditional" slow approach to computing the local impulse response is to evaluate (5) iteratively. We initialized iterations with an impulse at the response position and used 500 conjugate gradient iterations to estimate the response. This yields a well-converged estimate. We compare this approach to the fast predictions described in Sections III and IV. For all fast predictions, we used the precomputed linear operators given in (42). The predictors were applied using the modified diagonal elements in (37).

Because the resolution properties of SPECT systems are space-variant, we investigated the resolution at several positions in the object. These positions are identified with $+$ marks in the left two central slice images in Fig. 3. From left to right, we label these positions: "Rib," "Left kidney," "Center," "Soft tissue," and "Right elbow."

For the first resolution investigation, we used precomputed operators with a $30 \times 30 \times 30$ subvolume [i.e., $\eta = 30$ in (48)] and 32 blocks of 4 angles. We stored operators within a single quadrant of the elliptical orbit and used a spatial subsampling with $n_d = 6$. Operators for unsampled positions are formed using bilinear interpolation. Thus, the precomputed and stored operators may be stored as single precision floating point numbers in approximately 125 Mb.

Fig. 4 compares the local impulse responses at four different locations. The left set of figures compares local impulse responses calculated at the "Rib" position. Transverse, sagittal, and coronal slices of the 3-D response are shown for the iteratively calculated response (top row) and for our fast prediction (middle row). The bottom row shows profiles through each axis of the iteratively calculated response (dashed line) and the fast prediction (solid line). The right set of figures shows axial profiles for three more points. (None of these locations coincide with operator sampling positions. Thus all fast predictions are based on interpolated operators.) The local impulse responses are space-variant and anisotropic with coarser resolution near the center of the field of view. Despite the multiple approximations and subsampling, our predictions are very close to the iteratively calculated responses. This is true even for the "Rib" position where the attenuation map changes rapidly near the evaluation position.

Assuming the $\boldsymbol{A}$ and $\hat{\boldsymbol{H}}$ matrices have been precomputed and loaded, the Matlab implementation of the resolution predictor used to compute the above predictions, takes roughly 1/15 of a second to compute a single local impulse response on an 800-MHz Pentium III computer. For comparison with the "traditional" slow iterative approach, we note that a single projection operation, $\boldsymbol{H}\underline{\lambda}$, implemented as an "on-the-fly" procedure in an efficient compiled C program takes more than a minute on the same computer.

The required size of the precomputed operators depends on a number of factors including the desired accuracy of the approximation, available storage, desired computation speed, the

---

[11]For the diagonal in (49), we have assumed that the blur due to the system model is much greater than the blur induced by regularization of the estimator. Thus, $\underline{\dot{Y}}(\underline{\lambda}) \approx \underline{\dot{Y}}(\underline{\check{\lambda}})$.

Moreover, when the true projections, $\bar{Y}_i$, are unknown, one can often obtain good estimates via a simple plug-in approach using the noisy data Y_i.
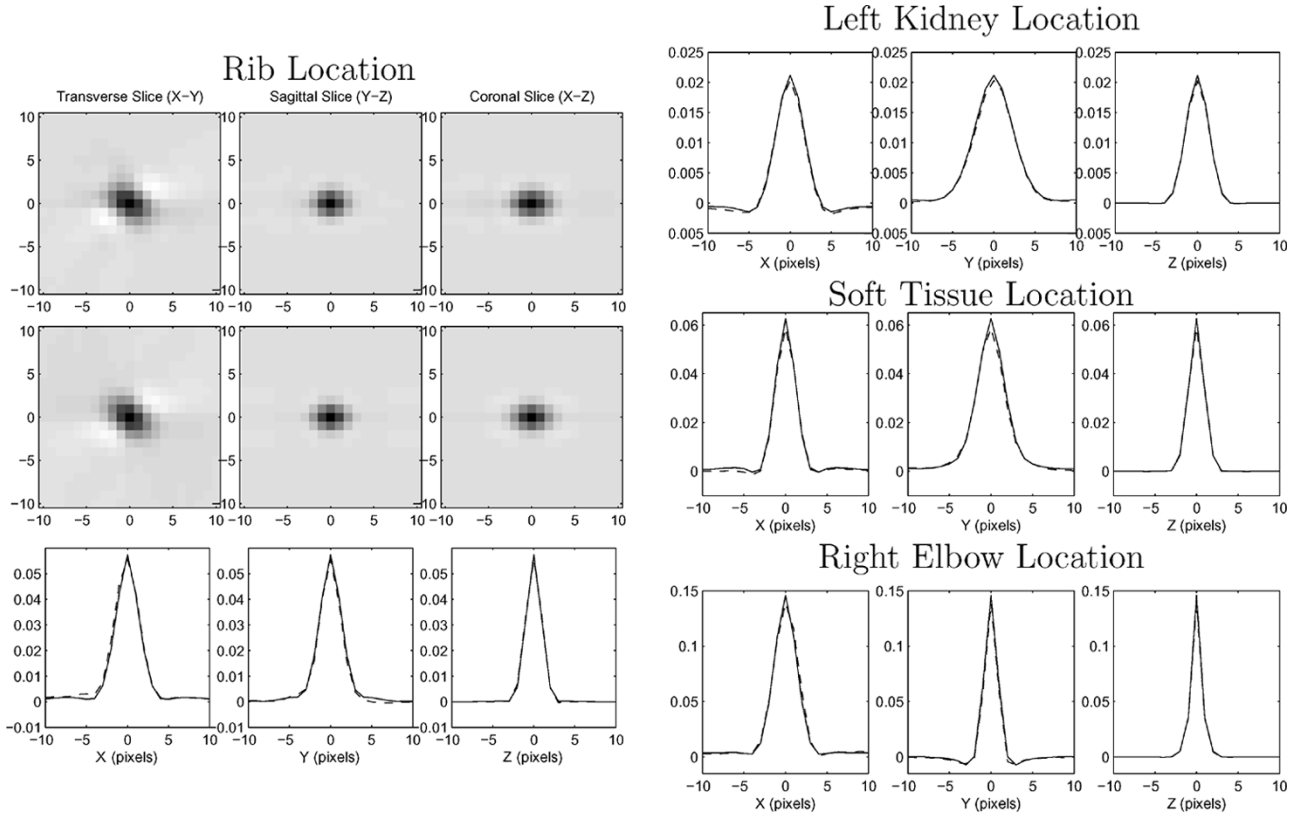
Fig. 4. A comparison of local impulse responses using the predictors of Section IV and an iterative calculation for four locations in the image volume. The left column compares transverse, sagittal, and coronal images of the 3-D local impulse response at the rib location using the iterative method (top row) versus the fast predictor (middle row). The bottom row shows profiles through each axis of the response for the iterative method (dashed line) and the fast predictor (solid line). The right column shows the axial profiles for three more image locations.

space-variance of the system, and the space-variance due to the object. We present two studies where the size of the operators are varied and briefly discuss the associated tradeoffs.

We first studied the local impulse responses at the five positions shown in Fig. 3 using operators computed with a range of support sizes. Specifically, cases where 60^3, 30^3, 20^3, and 14^3 voxels are stored. All angles are stored (i.e., 128 blocks) and the locations are sampled positions (therefore no interpolation of operators is performed). The results of this investigation are presented in Fig. 5.

Most support sizes give remarkably similar predictions across the supported pixels, even for the smaller support sizes where there is significant truncation of the local impulse response function. However, there are some noticeable differences for the smaller support sizes. Specifically, with additional truncation there are growing mismatches in the sidelobe behavior shown in the X and Y profiles for the center pixel's response. Similarly, for the smallest subvolume, a mismatch in the peak value of the local impulse response begins to be evident. We quantify this local impulse response mismatch for the five locations in the table in Fig. 5, where we have defined the normalized error as

$$\frac{\max_k \left| \hat{l}_k^j - l_k^j \right|}{\max_k \left| l_k^j \right|} \cdot 100\% \qquad (50)$$

where l_k^j denotes elements of the local impulse response at the jth location, as calculated by the "traditional" iterative

approach, and $\hat{l}_k^j$ denotes elements of the response as calculated by the fast approach. We also list the computation time for a single local impulse response evaluation for each support size. Since it appears that relatively good approximations can be made within the stored support, one may only need to store voxels over a region slightly larger than the desired portion of the response. This not only saves storage space for the precomputed operators, but also decreases prediction computation time by greatly reducing the dimension of the matrix multiplications.

We performed a second study, where the support size is held constant using 30^3 voxels and the angular subsampling is varied with 128 blocks, 16 blocks, 8 blocks, and a single block. Fig. 6 summarizes these results. For the coarser angular sampling, there are significant differences in the sidelobe behavior. These differences are most noticeable in the negative sidelobes in the X profile for the two coarsest samplings. These mismatches should be most pronounced in locations that differ from the geometric response in a very anisotropic fashion. The degree of mismatch will of course depend on the particular angular sampling and the properties of the object and system geometry. For this particular object and geometry, using only 8 blocks still yields approximations with less than 10% normalized error. We note that the "Rib" location generally has higher errors than the other locations. This is most likely due to the rapid local changes in attenuation, which are less likely to fit the approximation made in (20).

One other adjustable value is the coarseness of the operator position sampling represented by n_d. We have found that

SUMMARY OF NORMALIZED PREDICTION ERRORS

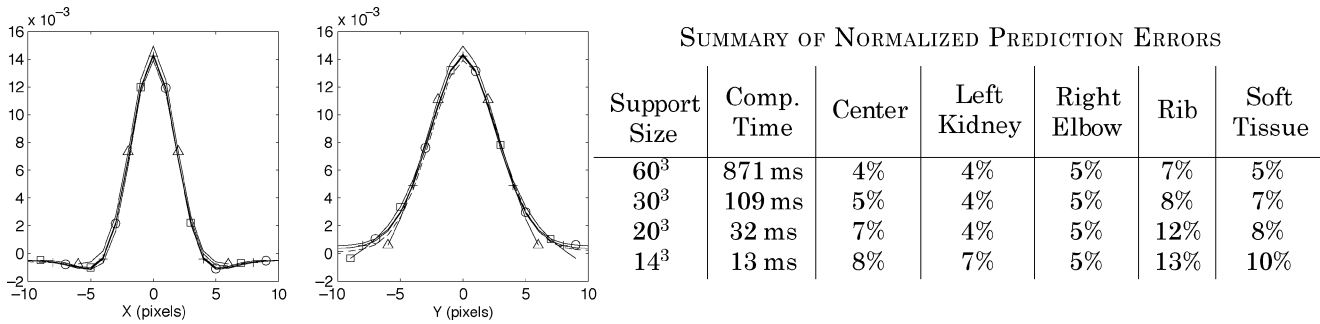| Support Size	Comp. Time	Center	Left Kidney	Right Elbow	Rib	Soft Tissue
60^3	871 ms	4%	4%	5%	7%	5%
30^3	109 ms	5%	4%	5%	8%	7%
20^3	32 ms	7%	4%	5%	12%	8%
14^3	13 ms	8%	7%	5%	13%	10%

Fig. 5. Resolution prediction with varying support size. The plots at left show profiles through the X and Y axes of the 3-D local impulse response at the center voxel with a support size of 60^3 voxels (+), 30^3 voxels (o), 20^3 voxels (□), and 14^3 voxels (△). The iteratively computed response is also shown (dashed line). The table on the right summarizes normalized error and computation time for resolution predictions at various locations and support sizes.

SUMMARY OF NORMALIZED APPROXIMATION ERRORS

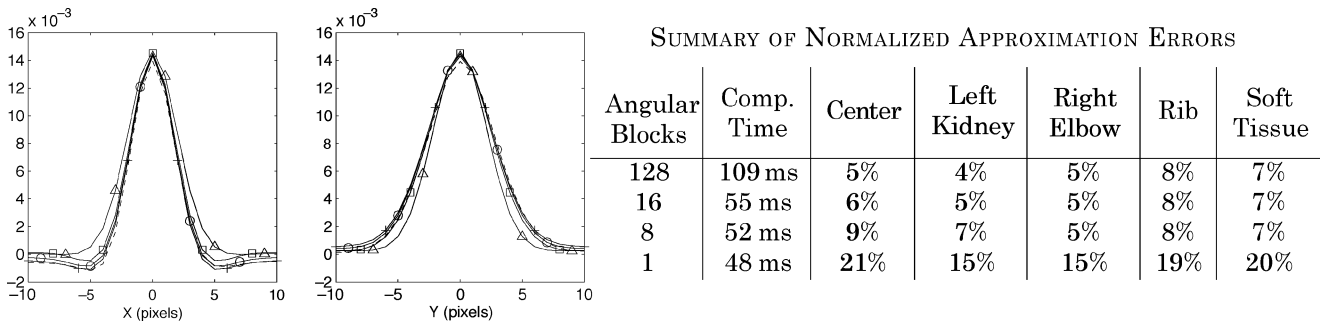| Angular Blocks	Comp. Time	Center	Left Kidney	Right Elbow	Rib	Soft Tissue
128	109 ms	5%	4%	5%	8%	7%
16	55 ms	6%	5%	5%	8%	7%
8	52 ms	9%	7%	5%	8%	7%
1	48 ms	21%	15%	15%	19%	20%

Fig. 6. Resolution prediction with varying angular sampling. The plots at left show profiles through the X and Y axes of the 3-D local impulse response at the center voxel with 128 blocks (+), 16 blocks (o), 8 blocks (□), and a single block (△). The iteratively computed response is also shown (dashed line). The table on the right summarizes normalized error and computation time for resolution predictions at various locations and angular samplings.

one can use a fairly coarse sampling $(n_d = 6)$, since the (unweighted) geometric response varies very smoothly. Finer sampling helps reduce interpolation computations. However, the required sampling is quite coarse, and ultimately depends on the particular system geometry.

D. Covariance Prediction

We also investigated local covariance predictions. We compared the fast predicted covariance functions versus empirical covariance functions estimated from 500 noisy reconstructions. As with the resolution predictors, we use the precomputed operators given in (42) in conjunction with the modified diagonal elements stated in (37). We use the covariance equation given in (45) and the diagonal weighting $D_3 = D_1$, as in (49). We used the same operator dimensions and subsamplings as in the initial resolution investigation.

Fig. 7 presents the empirical covariance functions and the predicted covariance functions for four positions in the digital phantom. The variation in the sample covariances is quite evident in the image slices and the profiles. Thus, we have included error bars on the sample covariance estimates (based on an assumption of a Gaussian distribution of the reconstructed image values). These error bars indicate plus and minus one standard deviation of the covariance estimate. The covariance predictions appear quite accurate over these four positions, lying within the error bars for most locations. It seems likely that these predictions would be sufficiently accurate for typical applications such as making variance images or evaluating computer observer performance.

We performed one final investigation of the accuracy and speed of the predictions. We calculated a variance image for the central slice of the 3-D phantom. We used precomputed operators with 12^3 voxels and 16 blocks of 8 angles. We stored operators with $n_d = 1$ over a single quadrant (within the elliptical orbit). This takes approximately 160 Mb of storage space.

We used the variance predictor discussed in Section IV, which eliminates the inverse Fourier transforms. We also applied the scaling technique developed by Qi in [11] to account for the effects of nonnegativity constraint on the reconstructed images. Fig. 8 shows the predicted and empirical standard deviation images. Sample standard deviations were calculated using the 500 noisy reconstructions (left image) and the fast predictors (center). We also show a central horizontal profile of the standard deviations, which have been normalized to be a percentage of the warm background in the phantom. Plus and minus single standard deviation error bars on the sample variance estimates are also shown. The predictions agree very well with the sample variance estimates laying within the error bars for almost all positions.

Given the precomputed matrices specified by (42) and the precomputed bilinear interpolator, $\hat{H}$ in (37), the entire (single slice) standard deviation image was computed in less than 20 s using a Matlab implementation on an 800–MHz Pentium III processor. Thus, the variance of the entire volume can be predicted in less than 10 min. We expect that efficient routines written in a compiled C program would be significantly faster.
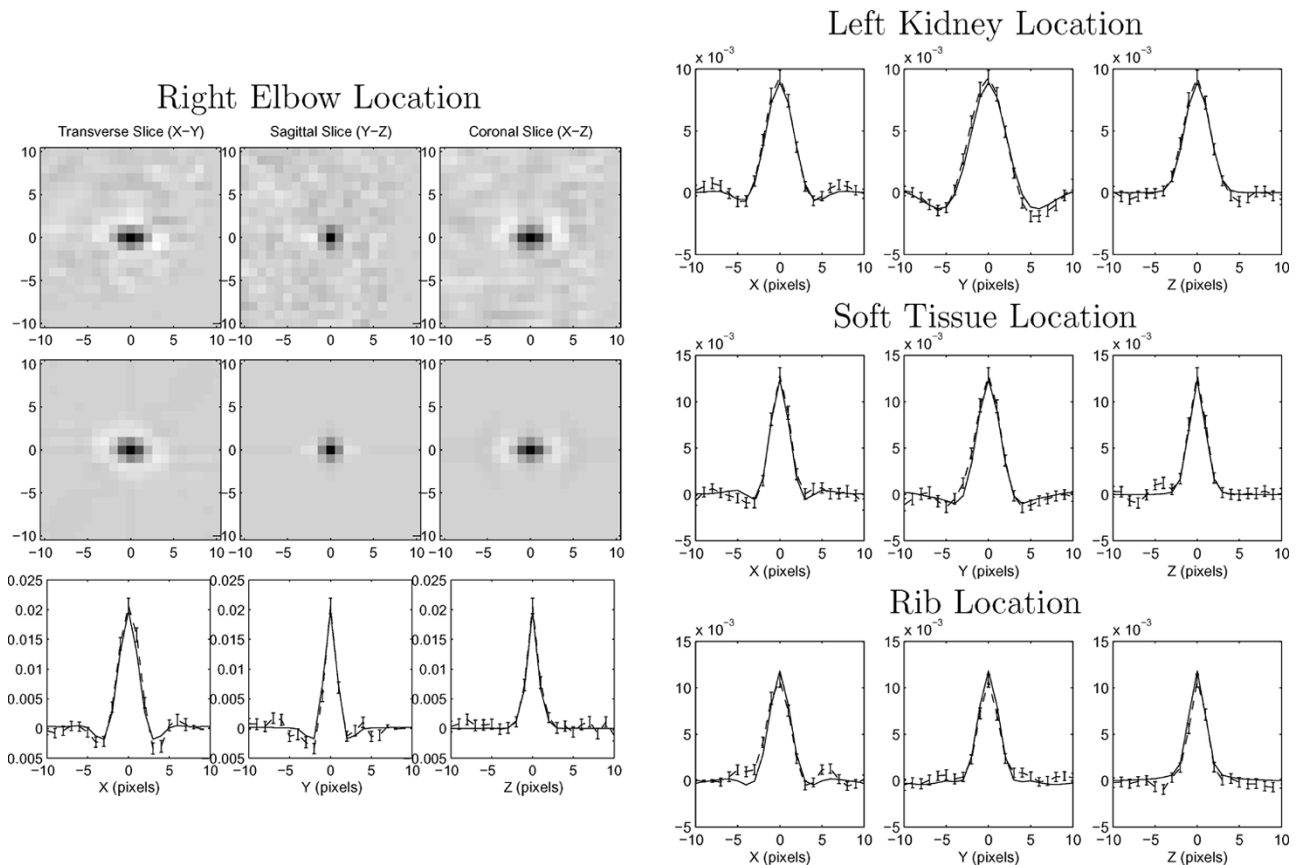
Fig. 7. A comparison of covariance functions calculated using the predictors of Section IV and estimates the sample covariance from 500 noisy reconstructions for four locations in the image volume. The left column compares transverse, sagittal, and coronal images of the 3-D covariance function at the elbow location calculated from the 500 reconstructions (top row) versus the fast predictor (middle row). The bottom row shows profiles through each axis of the response for sample covariance (dashed line) and the fast predictor (solid line). The right column shows the axial profiles for three more image locations.
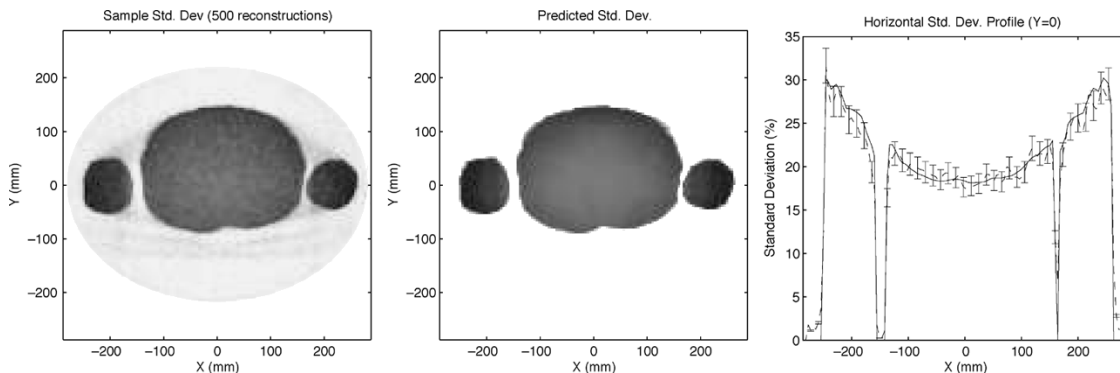

Fig. 8. A comparison of central slice standard deviation images created from calculating the sample standard deviation of 500 3-D reconstructions, and from the fast variance predictor discussed in Section IV. A horizontal profile is shown at left showing the sample standard deviation values (dashed line) and the predicted values (solid line).

VI. DISCUSSION

This paper has presented extensions of the techniques discussed in [1] to allow for rapid calculation of the local noise and resolution properties of penalized-likelihood image estimates. These techniques are appropriate for 2-D or 3-D SPECT systems with nonuniform attenuation and are based on precomputing portions of the predictors that are independent of the object being scanned. The projection and backprojection operators for SPECT system need not be available in precomputed form. Of course, the attenuation factors represented by a_{ij} must be

readily available to provide rapid predications. The predictions based on these methods are very accurate and can be evaluated with very practical computation times, once the precomputations have been performed for a given SPECT geometry.

Shift-variant PET systems can also use the methods presented here. Similarly, some of the general principles may apply to other shift-variant statistical reconstruction problems such as X-ray computed tomography and magnetic resonance imaging.

The prediction speed is a function of the size and sampling of the precomputed operators, M^j. Thus the speed is directly related to how many precomputations one is willing to store.

We have demonstrated that accurate predictions can be made for practical storage sizes (e.g., better than 10% error with 125 Mb of storage for the sample SPECT system we have investigated), but the exact tradeoff must be specified by the user's requirements on the accuracy and speed of the predictions.

The fast predictors we have developed are most appropriate for situations that require repeated predictions for a static system geometry. Such situations include object-dependent penalty design as in [1] and [8], where predictions are required for every voxel. Without fast techniques for resolution and noise prediction, these penalty designs methods would be too slow for practical implementations. Such fast predictors are also important for the study of computer observers [9], where repeated covariance estimates may be required.

Future work should compare other fast methods like those discussed in [12] to the methods presented here. Additionally, the methods presented here may not be appropriate for 180° SPECT, where responses may be highly asymmetric. Future fast resolution and covariance predictor studies should try to accommodate such operating modes.

REFERENCES

[1] J. W. Stayman and J. A. Fessler, "Regularization for uniform spatial resolution properties in penalized-likelihood image reconstruction," *IEEE Trans. Med. Imag.*, vol. 19, pp. 601–615, June 2000.

[2] H. H. Barrett, D. W. Wilson, and B. M. W. Tsui, "Noise properties of the EM algorithm: I. Theory," *Phys. Med. Biol.*, vol. 39, pp. 833–846, 1994.

[3] W. Wang and G. Gindi, "Noise analysis of MAP-EM algorithms for emission tomography," *Phys. Med. Biol.*, vol. 42, pp. 2215–2232, Nov. 1997.

[4] E. J. Soares, C. L. Byrne, and S. J. Glick, "Noise characterization of block-iterative reconstruction algorithms. I. Theory," *IEEE Trans. Med. Imag.*, vol. 19, pp. 261–270, Apr. 2000.

[5] J. A. Fessler, "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.

[6] 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.

[7] J. W. Stayman and J. A. Fessler, "Compensation for nonuniform resolution using penalized-likelihood reconstruction in space-variant imaging systems," *IEEE Trans. Med. Imag.*, vol. 23, pp. 269–284, Mar. 2004.

[8] J. Qi and R. M. Leahy, "A theoretical study of the contrast recovery and variance of MAP reconstructions with applications to the selection of smoothing parameters," *IEEE Trans. Med. Imag.*, vol. 18, pp. 293–305, Apr. 1999.

[9] P. Bonetto, J. Qi, and R. M. Leahy, "Covariance approximation for fast and accurate computation of channelized hotelling observer statistics," *IEEE Trans. Nucl. Sci.*, vol. 47, pp. 1567–1572, Aug. 2000.

[10] J. Qi and R. H. Huesman, "Theoretical study of lesion detectability of MAP reconstruction using computer observers," *IEEE Trans. Med. Imag.*, vol. 20, pp. 815–822, Aug. 2001.

[11] J. Qi and R. M. Leahy, "Resolution and noise properties MAP reconstruction for fully 3-D PET," *IEEE Trans. Med. Imag.*, vol. 19, pp. 493–506, May 2000.

[12] Y. Xing, I. T. Hsiao, and G. R. Gindi, "Efficient calculation of resolution and variance in 2D circular-orbit SPECT," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf.*, vol. 3, 2001, Paper M5B-5, pp. 1298–1302.

[13] Y. Xing and G. Gindi, "Rapid calculation of detectability in bayesian SPECT," in *Proc. IEEE Int. Symp. Biomed. Imag.*, July 2002, pp. 78–81.

[14] R. M. Lewitt, P. R. Edholm, and W. Xia, "Fourier method for correction of depth-dependent collimator blurring," *Proc. SPIE 1092, Med. Imaging III: Image Processing*, pp. 232–243, 1989.

[15] J. A. Fessler, "Approximate variance images for penalized-likelihood image reconstruction," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf.*, vol. 2, 1997, pp. 949–952.

[16] The Zubal Phantom. Image Processing and Analysis Group, Dept. Diagnostic Radiology and Electrical Engineering and the Yale School of Medicine, Yale Univ., New Haven, CT. [Online]. Available: http://noodle.med.yale.edu/zubal/

[17] G. Zubal, G. Gindi, M. Lee, C. Harrell, and E. Smith, "High resolution anthropomorphic phantom for Monte Carlo analysis of internal radiation sources," in *Proc. IEEE Symp. Computer-Based Medical Systems*, 1990, pp. 540–547.

[18] J. A. Fessler and H. Erdoğan, "A paraboloidal surrogates algorithm for convergent penalized-likelihood emission image reconstruction," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf.*, vol. 2, 1998, pp. 1132–1135.

[19] H. Erdoğan and J. A. Fessler, "Monotonic algorithms for transmission tomography," *IEEE Trans. Med. Imag.*, vol. 18, pp. 801–814, Sept. 1999.

[20] ——, "Ordered subsets algorithms for transmission tomography," *Phys. Med. Biol.*, vol. 44, pp. 2835–2851, Nov. 1999.