GPU-Based Sparse Bayesian Learning for Adaptive Transmission Tomography

by

HyungJu Jeon

Department of Electrical and Computer Engineering
Duke University

Date: _____

Approved:

_____
Lawrence Carin, Supervisor

_____
Andrew Hilton

_____
Galen Reeves

Thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science in the Department of Electrical and Computer Engineering
in the Graduate School of Duke University
2014

Abstract

GPU-Based Sparse Bayesian Learning for Adaptive Transmission Tomography

by

HyungJu Jeon

Department of Electrical and Computer Engineering
Duke University

Date: _____

Approved:

_____
Lawrence Carin, Supervisor

_____
Andrew Hilton

_____
Galen Reeves

An abstract of a thesis submitted in partial fulfillment
of the requirements for the degree of Master of Science
in the Department of Electrical and Computer Engineering
in the Graduate School of Duke University
2014

Abstract

The aim of this thesis is to investigate a GPU-based scalable image reconstruction algorithm for transmission tomography based on a Gaussian noise model for the log transformed and calibrated measurements. The proposed algorithm is based on sparse Bayesian learning (SBL) which promotes sparsity of the imaged object by introducing additional latent variables, one for each pixel/voxel, and learning them from the data using an hierarchical Bayesian model. We address the computational bottleneck of SBL which arises in the computation of posterior variances. Two scalable methods for efficient estimation of variances were studied and tested: the first is based on a matrix probing technique; and the second method is based on a Monte Carlo estimator. Finally, we study adaptive data acquisition methods, where instead of using a standard scan around the object, the source locations are selected based on the learned information from previously available measurements, leading to fewer projections.

Contents

List of Tables

List of Figures

List of Abbreviations and Symbols

Symbols

$\mathbf{x}$	Attenuation coefficient vector in column-major order
$\mathbf{H}$	System matrix (Forward projection)
$\mathbf{y}$	Normalized log-transformed measurements
I_0	Calibration measurements
$\mathbf{\Sigma}$	Posterior covariance.
N	Number of pixels
M	Number of rays
N_s	Number of samples/vectors used in diagonal estimator
$\mathcal{H}$	Shannon entropy

Abbreviations

CT	Computed Tomography
SBL	Sparse Bayesian Learning
GPU	Graphic Processing Unit
GMRF	Gaussan Markov Random Field
RMSE	Root Mean Squared Error
EM	Expectation-Maximization
FBP	Filtered-Back-Procjection
IG	Information Gain

1

Introduction

Tomographic image reconstruction is the process of estimating an object from measurements of its line integrals along different angles [1]. Perhaps one of the most widely known examples is x-ray computed tomography (CT) [2], which is used in various medical and security applications [1]. Tomographic reconstruction is an ill-posed problem [3] in the sense that multiple solutions exist that are consistent with the data. To improve image quality, it is often desirable to incorporate prior knowledge, which includes the statistical characteristics of the measured data, and properties that are expected of the image.

The images in transmission tomography represent attenuation per unit length of the impinging beam due to energy absorption inside the imaged object. An important property of these images is that they can be well approximated by a sparse representation in some transform domain. In some applications, sparsity is present directly in the native image domain [4], but more commonly it is present in the pixel-difference domain or in some transform domain, such as the wavelet, curvlet or shearlet transforms.

There are generally two types of approaches for tomographic image reconstruc-

tion. The first type consists of one-shot algorithms such as filtered back-projection [1] and its extensions, which rely on analytic formulas. These algorithms cannot incorporate the type of prior knowledge mentioned above and also produce prominent artifacts when some of the data are missing. The second type consists of iterative algorithms based on minimizing some cost function. The latter enable to incorporate prior knowledge about the image by adding penalties that promote sparsity in some representation. In addition, iterative algorithms are far more robust to missing data than one-shot algorithms.

In this work, we further develop a sub-class of statistical methods known as sparse Bayesian learning. The studied algorithm incorporates prior knowledge about the nature of the measured signals which includes an object-dependent noise variance that originate in the log transformed Poisson measurements and Beer's law [5]. It also includes a model for the sparsity of the image in the pixel/voxel difference domain. What sets the SBL apart from prior art in CT is the fact that it can automatically learn the balance between data-fidelity and the penalty due to prior knowledge (automatic relevance determination), and thus does not require any tuning of parameters. In addition, it also allows to compute variances or Bayesian confidence intervals. An important motivation for SBL is adaptive sensing/experimental design, where the measurements are selected based on the learned information from previously available measurements, and using information theoretic measures to select the "best" measurement. The latter requires the posterior covariance matrix which is provided by SBL, but not by standard reconstruction methods for CT.

The contributions of this work are in the study of computationally efficient GPU-based methods for estimating the posterior variances required in SBL and estimating information theoretic measures required in adaptive sensing. We focus on the particular structure of the system matrix for transmission tomography and study which method provides the best accuracy and speed. We also study the practical use of

adaptive sensing for CT.

2

Model and Methods

2.1 Model

Transmission computed tomography (TCT) can be stated as an inverse problem in which a multi-dimensional distribution $f(\mathbf{x})$ is estimated from its line integrals. In the ideal setting, each measurement at a detector is proportional to the number of photons arriving at that detector. A common model for the noise in X-ray measurements assumes that the mean number of photons received at a detector I_d can be computed using Beer's Law: $I_d = I_0 \exp(-\int_L f(\mathbf{x})\, dl)$, where I_0 are mean number of photons emitted from the source. For large counts, tbe normalized post-log measurements, $y = \log \frac{I_0}{I_d}$, follow approximately a Gaussian distribution with mean $\bar{y}$ and the variance σ_y^2 given by the Eq. 2.1 [5]

$$\bar{y} = \int_L \alpha\, dl \qquad \sigma_y^2 = \frac{\exp(\bar{y})}{N_0} \tag{2.1}$$

Discretizing the line integrals and $f(\mathbf{x})$ is given in matrix form

$$\mathbf{y} = \mathbf{Hf} + \epsilon \tag{2.2}$$

where $\mathbf{f}$ is the sampled $f(\mathbf{x})$ on a Cartesian grid and rearranged as a vector. $\mathbf{f}$ is the linear attenuation constant inside the object in unit of mm^{-1}. $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_M]^\top$ is the system matrix where each row $\mathbf{h_i}$ corresponds to a line integral, also called projection. One approach to compute H_{ij}, which we use here, is to calculate the length of intersection between i-th ray and j-th pixel. The X-ray CT measurements are assumed to be monoenergetic and without low photon counts. The distribution of log-transformed measurement $\mathbf{y}$ can be approximated by a Gaussian distribution

$$\mathbf{y} \sim \mathcal{N}(\mathbf{Hx}, \mathbf{B}^{-1}), \quad \mathbf{B} = \mathrm{diag}(I_0 \exp(-\mathbf{Hx})), \tag{2.3}$$

$\mathrm{diag}(\mathbf{v})$ denotes a diagonal matrix with the diagonal elements given by the vector $\mathbf{v}$.

2.1.1 Sparse Bayesian Learning

Sparse Bayesian learning (SBL) [6] utilizes a zero-mean Gaussian probability distribution to promote the sparsity. In our model, the object, x accordingly, is assumed to have a sparse representation, $\mathbf{s} = \mathbf{Dx}$ under some operator $\mathbf{D}$. We propose the prior

$$p(\mathbf{x}|\boldsymbol{\alpha}) = \mathcal{N}(\mathbf{x}; \mathbf{0}, (\mathbf{D}^\top \mathbf{A} \mathbf{D})^{-1}), \quad \mathbf{A} = \mathrm{diag}(\boldsymbol{\alpha}) \tag{2.4}$$

For a fixed $\boldsymbol{\alpha}$, the posterior distribution $p(\mathbf{x}|\mathbf{y}, \boldsymbol{\alpha})$ can be computed using Bayes' rule and is given by

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}(\boldsymbol{\alpha}), \mathbf{P}^{-1}(\boldsymbol{\alpha})) \tag{2.5}$$

where

$$\boldsymbol{\mu} = \mathbf{P}^{-1} \mathbf{H}^\top \mathbf{B} \mathbf{y} \tag{2.6}$$

$$\mathbf{P} = (\mathbf{H}^\top \mathbf{B} \mathbf{H} + \mathbf{D}^\top \mathbf{A} \mathbf{D}) \tag{2.7}$$

Direct computation of $\mathbf{P}^{-1}$ is not practical in most cases since the full covariance matrix can not be explicitly stored in memory, preconditioned conjugate gradient

(CG) [7] methods can be used to efficiently solve the equation $\mathbf{P}\boldsymbol{\mu} = \mathbf{H}^\top\mathbf{B}\mathbf{y}$ efficiently. Moreover, by carefully choosing $\mathbf{D}$ (section 2.1.2), entire operations required in this method (i.e., $\mathbf{H}, \mathbf{H}^\top, \mathbf{D}$) can be computed on-the-fly(section 2.2) without storing any large matrix, making this method feasible even for very large scale problems.

In SBL, the values for γ are chosen by maximizing marginal distribution (type-II MAP) in Eq. 2.8

$$p(\mathbf{y}|\boldsymbol{\gamma}) = \frac{1}{\left(\sqrt{2\pi}\right)^N}|\mathbf{B}^{-1} + \mathbf{H}\left(\mathbf{D}^\top\mathbf{A}\mathbf{D}\right)^{-1}\mathbf{H}^\top|^{-1/2} \exp\left\{-\frac{1}{2}\mathbf{y}^\top\left(\mathbf{B}^{-1} + \mathbf{H}\left(\mathbf{D}^\top\mathbf{A}\mathbf{D}\right)^{-1}\mathbf{H}^\top\right)^{-1}\mathbf{y}\right\} \tag{2.8}$$

Values of $\boldsymbol{\gamma}$ that maximizes marginal distribution (Eq.2.8), are analytically intractable and therefore an EM algorithms is used to update $\boldsymbol{\gamma}$.

$$(E) \quad \mathbf{x}^{(\mathbf{t+1})} : \mathbf{P}^{(\mathbf{t})}\mathbf{x}^{(\mathbf{t+1})} = \mathbf{H}^\top\mathbf{B}\mathbf{y} \tag{2.9}$$

$$(M) \quad \alpha_i^{(t+1)} : \left[\left[\mathbf{D}\boldsymbol{\Sigma}^{(\mathbf{t})}\mathbf{D}^\top\right]_{ii} - [\mathbf{D}\mu]_i^2\right]^{-1} \tag{2.10}$$

where $[\cdot]_i$ and $[\cdot]_{ii}$ stands for i-th element in the vector or on the main diagonal of the matrix

2.1.2 Prior choice

In Bayesian inference, the prior model $p(\mathbf{x})$ represents a probabilistic description about the solution or its properties known before acquiring any data.

Our *a priori* assumption on the attenuation coefficient in CT comes from statistical property of natural images: wavelet coefficient or simple derivative of natural images follow super-Gaussian distribution [8, 9]. In other words, the majority of simple derivative or wavelet coefficient are close to 0 and non-zero terms are significantly large. This suggest that images are (1) divided into several piecewise smooth regions and has (2) few of edges. So our choice of prior is now limited to one that

has properties above and can be expressed in quadratic form following Eq.2.4, The most simple prior would be the one that penalizes the difference in each direction separately:

$$\mathbf{D} = \begin{bmatrix} \mathbf{D_h} \\ \mathbf{D_v} \end{bmatrix} \tag{2.11}$$

$$\mathbf{D_h}(x_{i,j}) = x_{i,j} - x_{i-1,j} \tag{2.12}$$

$$\mathbf{D_v}(x_{i,j}) = x_{i,j} - x_{i,j-1} \tag{2.13}$$

with the prior of the form

$$p(\mathbf{x}|\boldsymbol{\alpha}) \propto \exp\left(-\frac{1}{2}\mathbf{x}\mathbf{D}^\top\mathbf{A}\mathbf{D}\mathbf{x}\right) \tag{2.14}$$

$$= \exp\left(-\frac{1}{2}\sum_i \alpha_i \left(\mathbf{D_h}^2(x_i) + \mathbf{D_v}^2(x_i)\right)\right) \tag{2.15}$$

The resulting prior is a weighted version of thin-membrane prior which is commonly used in data interpolation [10]. However, instead of piecewise linear field, the thin membrane prior prefers piecewise constant leveled surface and incurs large bias due to edge smoothing [11, 12]. Another commonly used prior in data interpolation is the thin-plate prior. The thin-plate penalizes the curvature in the image. Thin-plate prior utilizes second order partial derivative and can be expressed as follows:

$$p(\mathbf{x}|\boldsymbol{\alpha}) \propto \exp\left(-\frac{\alpha}{2}\sum_i \left(\mathbf{D_{hh}}^2(x_i) + \mathbf{D_{vv}}^2(x_i) + 2\mathbf{D_{hv}}^2(x_i)\right)\right) \tag{2.16}$$

$$\mathbf{D_{hh}}(x_{i,j}) = x_{i-1,j} + x_{i+1,j} - 2x_{i,j} \tag{2.17}$$

$$\mathbf{D_{vv}}(x_{i,j}) = x_{i,j-1} + x_{i,j+1} - 2x_{i,j} \tag{2.18}$$

$$\mathbf{D_{hv}}(x_{i,j}) = x_{i+1,j+1} - x_{i,j+1} - x_{i+1,j} + x_{i,j} \tag{2.19}$$

Thin-plate prior can successfully recovers piecewise smooth sources characteristic without incurring large bias errors, however, in both thin-membrane and thin-plate

prior, the resulting $\mathbf{D}$ is not a square matrix and non-invertible, thereby making hyperparameter update step in Eq.**??** much more difficult.

We show here that by making a simple modification to thin-plate prior one can achieve square, invertible $\mathbf{D}$ which has similar smoothing effect. First we will replace the diagonal derivative term in thin-plate prior with product of horizontal and vertical second order derivative and set different weight α_i for each x

$$p(\mathbf{x}|\boldsymbol{\alpha}) \propto \exp\left(-\frac{1}{2}\sum_i \alpha_i \left(\mathbf{D_{hh}}^2(x_i) + \mathbf{D_{vv}}^2(x_i) + 2\mathbf{D_{hh}}\mathbf{D_{vv}}(x_i)\right)\right) \tag{2.20}$$

$$= \exp\left(-\frac{1}{2}\sum_i \alpha_i \left(\mathbf{D_{hh}}(x_i) + \mathbf{D_{vv}}(x_i)\right)^2\right) \tag{2.21}$$

$$= \exp\left(-\frac{1}{2}\sum_{i,j} \alpha_{i,j} \left(x_{i-1,j} + x_{i+1,j} + x_{i,j-1} + x_{i,j+1} - 4x_{i,j}\right)^2\right) \tag{2.22}$$

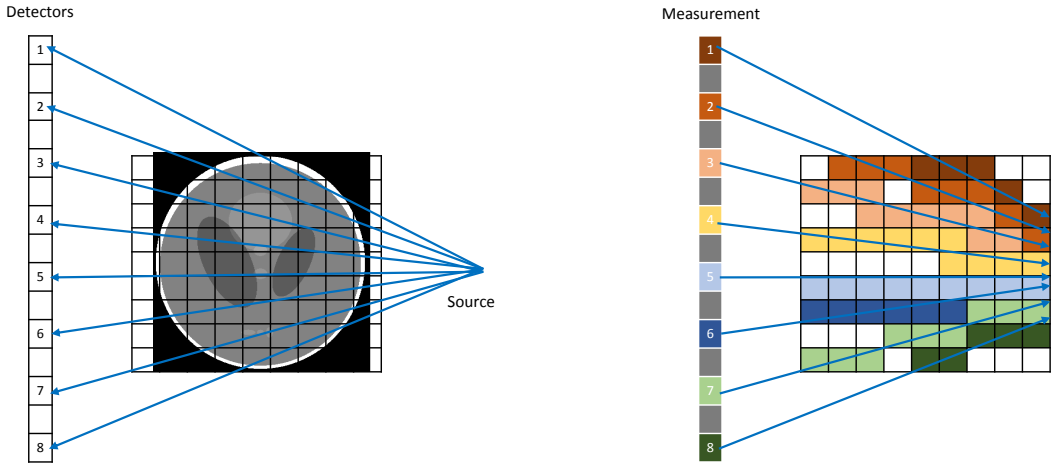
$$= \exp\left(-\frac{1}{2}\sum_{i,j} \alpha_i \left(4x_i - \sum_{j\in N(i)} x_j\right)^2\right) \tag{2.23}$$

where $N(i)$ are neighbors of i. As can be seen, this prior is effectively taking the difference between center and the average of its neighbor and penalizes it. The difference operator $\mathbf{D}$ now becomes

$$\mathbf{D} = \left[\mathbf{D_h} + \mathbf{D_v} + \mathbf{D_h}^\top + \mathbf{D_v}^\top\right] \tag{2.24}$$

As can be seen, the proposed prior has square form and is invertible when appropriate boundary conditions are given.

The difference between the behavior of proposed prior and that of thin-plate prior can be characterized by second derivative test discriminant: $\mathbf{D_{hh}}\mathbf{D_{vv}} - \mathbf{D_{hv}^2}$. Around the critical point, if $\mathbf{D_{hh}}\mathbf{D_{vv}} - \mathbf{D_{hv}^2} > 0$ (i.e. local extrema), the proposed prior penalizes the difference even stronger, resulting in stronger denoising effect. On the other hand, saddle points, ($\mathbf{D_{hh}}\mathbf{D_{vv}} - \mathbf{D_{hv}^2} < \mathbf{0}$), are less penalized.



(a) Forward Projection (b) Backwardrward Projection

FIGURE 2.1: Illustration of Forward and Backward Projection in 2D Fan-beam CT

2.2 Computation of Forward Operator

In discrete CT imaging model, the system matrix $\mathbf{H}$ represents the ray tracing along projection rays. Figure 2.1 illustrates both forward ($\mathbf{H}$) and backward ($\mathbf{H}^\top$) projection in 2D Fan-beam CT.

Given the large number of pixels (voxels) and rays, it is crucial to compute forward/backward projection efficiently. Although the system matrix in CT imaging is often sparse with $O(M\sqrt{N})$ non zeros elements, due to the large scale of the problem it is often difficult to store full matrix and is usually computed on-the-fly. A straight-forward ray tracing would require $O(N)$ computation for each ray given $N = n \times n$ pixels in square 2D image. Instead we used the center line-intersection method, also known as the Siddons method [13, 14].

In Siddon's method, each ray is parameterized in vector form: $\overrightarrow{x_o} + t\overrightarrow{v}$. Then the value of t at which the ray crosses the first voxel is computed separately for each axis by dividing the distance between starting coordinate and the next closest boundary by the magnitude of direction in corresponding axis. If t in x-axis($tMaxX$) is smaller

than that in y-axis($tMaxY$), in other words, if intersection in x axis occurs first, we update the current location of the ray and increase $tMaxX$ by $tDeltaX$ which represents the amount the ray has to move in order to pass one voxel horizontally. This process is repeated until the ray reaches the end of the grid. For forward projection, the product of length of intersection with each voxel (t) and attenuation coefficient on that voxel, is summed along the ray and stored. On the other hand, for backward projection, the product of intersection and the measurement intensity is stored in each voxel.

One of the advantages of forward projection using Siddon's method is that the memory required for each ray is relatively small and operation over each voxel consists of few simple computations. This makes forward projection very efficient on systems such as Graphics Processing Unit(GPU) where the task can be distributed among large number of processing cores with each core having relatively low computational power.

2.3 Estimation of Posterior Covariance Matrix

While the posterior covariance in Bayesian inference may provide additional useful informations such as uncertainty quantification, estimating posterior covariance is generally considered infeasible in large scale problems since full covariance matrix can not be explicitly stored in memory. However, in many cases we only need certain elements of the covariance matrix. Of particular interests are the diagonal of the covariance matrix (i.e. marginal variance) which arises in the hyperparameters update step of SBL model (Eq.**??**) and in experimental design [6, 15].

Several techniques has been developed for computing exact variances in Gaussian Markov random field (GMRF) using belief propagation [16, 17]. However, most of these technique are still not very scalable and are limited to restricted class of models.

Another recent approach to estimate variance in general model include Lanczos

algorithm [18] and has been studied in the context of variational Bayes [15,19]. While it can estimate the rough structure with only few iterations, accurate estimate can be made only with a very large number of iterations. Moreover due to the finite precision system, Lanczos vectors loses orthogonality after a certain number of iterations and need to be reorthogonalized. Such reorthogonalization step requires one to store the entire sequence of Lanczos vectors and may dominate the overall calculation when the number of required Lanczos vectors are large.

Here, we examined two techniques to estimate elements of posterior covariance matrix when the precision matrix ($\mathbf{P} = \mathbf{\Sigma}^{-1}$) is not explicitly given in a matrix form but the product of matrix-vector multiplication with some arbitrary vector can be efficiently computed on-the-fly. The first method involves solving sequence of linear equation using conjugate gradient and the second method is based on Monte Carlo estimator using perturbed samples drawn from simple Gaussian distribution. Both of these techniques are highly parallelizable and scalable.

2.3.1 Diagonal Estimator using low-rank matrix

In theory, it is possible to extract the diagonal of inverse of given matrix $\mathbf{\Sigma} = \mathbf{P^{-1}}$, $\mathbf{\Sigma} \in \mathbb{R}^{N \times N}$ by solving sequence of linear equations: $\mathbf{\Sigma}\hat{P}_i = e_i$ where e_i is the i-th standard basis vector and $\hat{P}_i$ is the i-th column of estimated diagonal matrix. However, this requires solving N linear equations using iterative method, each requiring $O(N)$ computation complexity [7]. Instead, one can design a low-rank matrix $\mathbf{VV'}$, with probing matrix $\mathbf{V} \in \mathbb{R}^{N \times M}$ where $M \ll N$ and use it in place of $\mathbf{I} = [e_1, e_2, \ldots, e_N]$.

Algorithm.1 describes general algorithm of diagonal of inverse estimator.

Algorithm 1 Diagonal Estimator

Require: $\mathcal{P}(x) : R^n \rightarrow R^n$ such that $b = \mathbf{P}x$.
$\mathbf{V} \in \mathbb{R}^{N \times M}$, $\mathbf{V} = [V_1, V_2, \ldots, V_M]$
Number of iteration steps s

Output: D^s denotes the approximated diagonal in vector form at step s

1: $D^0 = 0$
2: **for** $k = 1 \cdots s$ **do**
3: Solve linear equation $\mathbf{P}x_k = V_k$ using iterative method
4: $t_k = t_{k-1} + x_k \odot V_k$
5: $q_k = q_{k-1} + V_k \odot V_k$
6: $D^k = t_k \oslash q_k$
7: **end for**

where $\oslash$ and $\odot$ represents element-wise division and multiplication respectively. After s steps of approximations, the i-th element of approximation D^s can be expressed as follows

$$D_i^s = \frac{\sum_{k=1}^{s} V_k(i) \sum_{j=1}^{n} \sigma_{ij} V_k(j)}{\sum_{k=1}^{s} (V_k(i))^2} \tag{2.25}$$

$$= \sigma_{ii} + \sum_{i \neq j} \sigma_{ij} \frac{\sum_{k=1}^{s} V_k(i) V_k(j)}{\sum_{k=1}^{s} (V_k(i))^2} \tag{2.26}$$

σ_{ij} denotes (i, j)-th element of $\mathbf{\Sigma}$ and $V_k(i)$ denote i-th element of vector V_k (k-th column of $\mathbf{V}$). As can be seen, exact diagonal can be extracted when

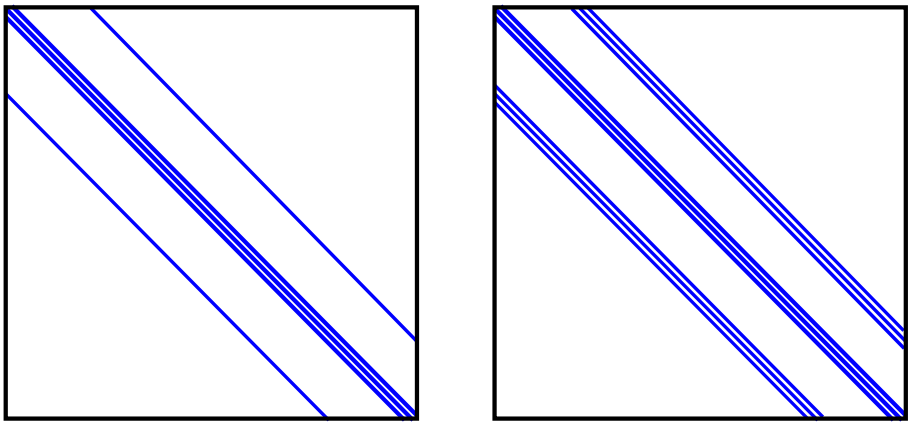
$$\sum_{i \neq j} a_{ij} \frac{\sum_{k=1}^{s} V_k(i) V_k(j)}{\sum_{k=1}^{s} (V_k(i))^2} \simeq 0 \tag{2.27}$$

To get the accurate estimation, it is crucial to design appropriate probing matrix $\mathbf{V}$ that satisfies Eq.2.27.

First, one can use random vectors drawn from normal Gaussian distribution(e.g. $V_k(i) \sim \mathcal{N}(0, 1)$) as the column of $\mathbf{V}$. Unbiased stochastic estimator based on random vectors drawn independently from normal distribution has been first proposed by Hutchinson in estimating the trace of a matrix and utilized in estimating diagonal of a matrix and/or matrix inverse [20–22]. However, unless the target matrix is highly diagonally dominant, stochastic estimator requires large number of probing vectors

to attain high accuracy solution. Bekas *et al.* utilized columns of Hadamard matrix instead of random vectors to overcome this, however, its effectiveness are limited to the case where the target matrix is banded.

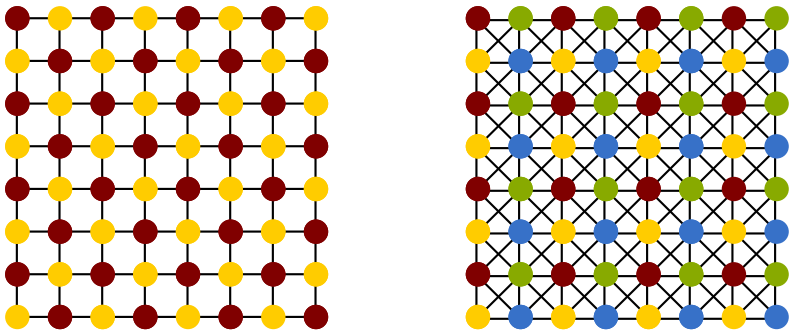
More sophisticated version of estimator exploits the sparsity pattern of target matrix and graph coloring algorithm [22–24]. Given the sparsity pattern of a matrix, adjacency graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ can be constructed where an edge between vertex i and j exists ($\{i, j\} \in \mathcal{E}$) only when $a_{i,j}$ in target matrix is non-zero. Then using graph coloring algorithm, the vertexes can be 'colored' such that each vertex and its neighboring vertexes have different colors.



(a) $N = 4$ (b) $N = 8$

FIGURE 2.2: Example sparsity pattern of matrix with neighbor N

Fig 2.2 shows example sparsity pattern of a matrix. Sparsity pattern of the coavariance matrix can also provide additional information. For example, if the covariance matrix has a sparsity pattern similar to that of Fig.2.3.1, this suggests that the each element is correlated only with its horizontal and vertical neighbors. Also, such sparsity pattern in precision matrix, implies that the underlying model is thin-plate model GMRF.



(a) $N = 4$ (b) $N = 8$

FIGURE 2.3: Colored adjacency graph on Cartesian grid with neighbor N

In Fig.2.3, colored adjacency graph corresponding to each case in Fig 2.2 is illustrated. It is important to note that the number of the colors used in Fig. 2.3 is the minimum. However, in general cases, finding the minimum number of colors required to color a given adjacency graph is known to be an NP-hard problem [25]. Once the coloring is done, the probing matrix $\mathbf{V}$ can be constructed as follows :

$$V_c(i) = \begin{cases} \rho & \text{if } Color(i) = c \\ 0, & \text{otherwise} \end{cases} \tag{2.28}$$

for values of ρ, Tang used 1 and Malioutov used ± 1. Although both method estimate diagonal reasonably well in practice, the method proposed by Malioutov *et. al* has clear advantage since it provides unbiased estimate of diagonal. Tang's method, however, are biased and also prone to error especially when the off diagonal elements do not decay fast enough.

In theory, computation for each probing vector can be done independently in parallel, and by using iterative solver (e.g. Conjugate gradient) estimation can be done efficiently even on very large scale problem.

While the diagonal of covariance are used the most in many situation, it is often advantageous to compute the off-diagonal elements as well. In this work, we propose a simple extension to existing methods that enables us to estimate off-diagonal element

of the matrix inverse without additional cost.

Recall that in original estimator using low-rank matrix, probing matrix $\mathbf{V}$ are chosen so that the numerator on the second term in Eq.2.26 converges to zeros for off-diagonal element (e.g. $\sum_{k=1}^{s} V_k(i)V_k(j) = 0$). If we can somehow make $\sum_{k=1}^{s} V_k(i)V_k(i) = 0$ and $\sum_{k=1}^{s} V_k(i)V_k(j) \neq 0$ for some (i, j) pair, we can then extract off-diagonal element, instead of diagonal elements. In order to achieve this without changing probing matrix, we will use shifted vector $V_k^{[m]}$ in component-wise multiplication on step 4, and 5 in Algorithm.1 where $V_k^{[m]}$ can be obtained by shifting up the V_k vector element-wise by m.

$$D^s_{[m],i} = \frac{\sum_{k=1}^{s} V_k^{[m]}(i) \sum_{j=1}^{n} \sigma_{ij} V_k(j)}{\sum_{k=1}^{s} V_k^{[m]}(i) V_k(i)} \tag{2.29}$$

$$= \frac{\sum_{k=1}^{s} V_k(i+m) \sum_{j=1}^{n} \sigma_{ij} V_k(j)}{\sum_{k=1}^{s} V_k(i+m) V_k(i)} \tag{2.30}$$

$$= \sigma_{i,i+m} + \sum_{j \neq i+m} \sigma_{ij} \frac{\sum_{k=1}^{s} V_k(i+m)V_k(j)}{\sum_{k=1}^{s} V_k(i+m)V_k(i)} \tag{2.31}$$

Since the computation bottle neck in this algorithm is in solving linear equation $\mathbf{P}x = V_k$, extracting off-diagonal elements can be done at the same time with diagonal elements at very little additional cost.

2.3.2 Sampling based Monte-Carlo Estimator

Posterior covariance matrix can be estimated using Monte Carlo estimator once we have samples drawn from this posterior distribution. While it is difficult to sample directly from the complex Gaussian posterior distribution, Papandreou showed that one can efficiently sample it by perturbing independent factors from distribution [26, 27]. Fig.**??** describes algorithm for drawing samples $\tilde{\mathbf{z}} \sim \mathcal{N}(0, \mathbf{\Sigma})$ from the posterior distribution Eq.2.7 ($\mathbf{\Sigma} = (\mathbf{H}^\top \mathbf{B}\mathbf{H} + \mathbf{D}^\top \mathbf{A}\mathbf{D})^{-1}$) using perturbation.

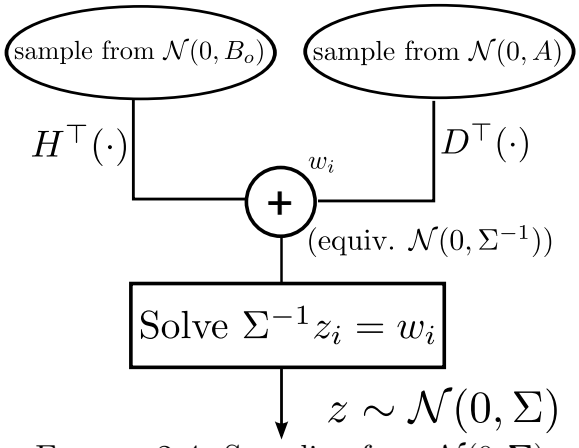


FIGURE 2.4: Sampling from $\mathcal{N}(0, \mathbf{\Sigma})$

samples from both of the distributions $\mathcal{N}(0, \mathbf{B})$ and $\mathcal{N}(0, \mathbf{B})$ can be drawn easily since both $\mathbf{B}$ and $\mathbf{A}$ are diagonal. Linear system $\mathbf{\Sigma}^{-1} z_i = w_i$ can be solved using preconditioned conjugate gradient method [7]. Once the samples $\tilde{\mathbf{z}} \sim \mathcal{N}(0, \mathbf{\Sigma})$ are drawn, Monte Carlo estimator for the posterior covariance matrix can be computed as follows:

$$\hat{\mathbf{\Sigma}} = \frac{1}{N} \sum_{i=1}^{N} z_i z_i^\top \tag{2.32}$$

As with previous methods, sampling based estimator also yields unbiased estimation and is highly scalable and parallelizable since each sample can be drawn independently and processed in parallel. Since the variance estimate follows chi-square distribution with degree of freedom equal to the number of samples N_s, its relative error $r = \sqrt{\frac{2}{N_s}}$ is independent from the problem size.

2.4 Experimental Design

As discussed in Ch.2.3, the ability to estimate the elements in the covariance matrix can be used in Bayesian sequential experimental design. In CT imaging, sequential experimental design can be applied to optimize the sampling scheme where given

the present data measurement system $\mathbf{H_{old}}$, most useful set of measurement $\mathbf{H_{new}}$ can be chosen. The term 'usefulness 'can vary depending on the context. Here, the usefulness (score) of the determined by the information gain.

$$IG(\mathbf{y_{new}}) := \mathcal{H}(p(\mathbf{x}|\mathbf{y_{old}})) - \mathcal{H}(p(\mathbf{x}|\mathbf{y_{old}}, \mathbf{y_{new}})) \tag{2.33}$$

$$= \log \left|\mathbf{B_{new}^{-1}} + \mathbf{H_{new}} \boldsymbol{\Sigma}_{\boldsymbol{old}} \mathbf{H_{new}}^{\top}\right| + \log |\mathbf{B_{new}}| \tag{2.34}$$

$$= \log \left|\boldsymbol{\Phi}^{-1}\right| + \log |\mathbf{B_{new}}| \tag{2.35}$$

Information gain in Eq.2.34 measures the decrease in entropy of the system after new measurement are taken into the model, and among candidates of new measurement system $\mathbf{H_{new}}$, one that maximizes information gain will be taken and appended to the system $\mathbf{H}$. This process of maximizing the determinant of information matrix is often called Bayesian D-optimality. Other criteria such as Bayesian A-optimality (trace of information matrix) can also be used. However, the computation of information gain often difficult for large system due to posterior covariance $\boldsymbol{\Sigma}$ inside the determinant.

Here, we will utilize the sampling based posterior variance estimator proposed by Papandreou *et. al* [27] to estimate information gain. Given the samples $\tilde{s} \sim \mathcal{N}(0, \boldsymbol{\Phi}^{-1})$ one can estimate $\log |\boldsymbol{\Phi}|$ using following relationship

$$\mathbb{E}\left[\exp\left(\frac{1}{2}\tilde{\mathbf{s}}^{\top}(\boldsymbol{\Phi} - \mathbf{P})\tilde{\mathbf{s}}\right)\right] = \frac{1}{2}\frac{|\boldsymbol{\Phi}|}{|\mathbf{P}|} \tag{2.36}$$

for some matrix $\mathbf{P}$ that approximates $\boldsymbol{\Phi}$ reasonably well. Using sample mean, this becomes

$$\frac{1}{N_s}\sum_{i}^{N_s}\exp\left(\frac{1}{2}\tilde{s_i}^{\top}(\boldsymbol{\Phi} - \mathbf{P})\tilde{s_i}\right) \simeq \frac{1}{2}\frac{|\boldsymbol{\Phi}|}{|\mathbf{P}|} \tag{2.37}$$

$$\log |\boldsymbol{\Phi}| \simeq \log |\mathbf{P}| - 2\log N_s - 2\log \sum_{i}^{N_s}\exp\left(\frac{1}{2}\tilde{s_i}^{\top}(\boldsymbol{\Phi} - \mathbf{P})\tilde{s_i}\right) \tag{2.38}$$

Gaussian samples $\tilde{s} \sim \mathcal{N}(0, \boldsymbol{\Phi}^{-1})$ can be drawn by adding two sample drawn and perturbed independently. This process is described in Fig.2.5

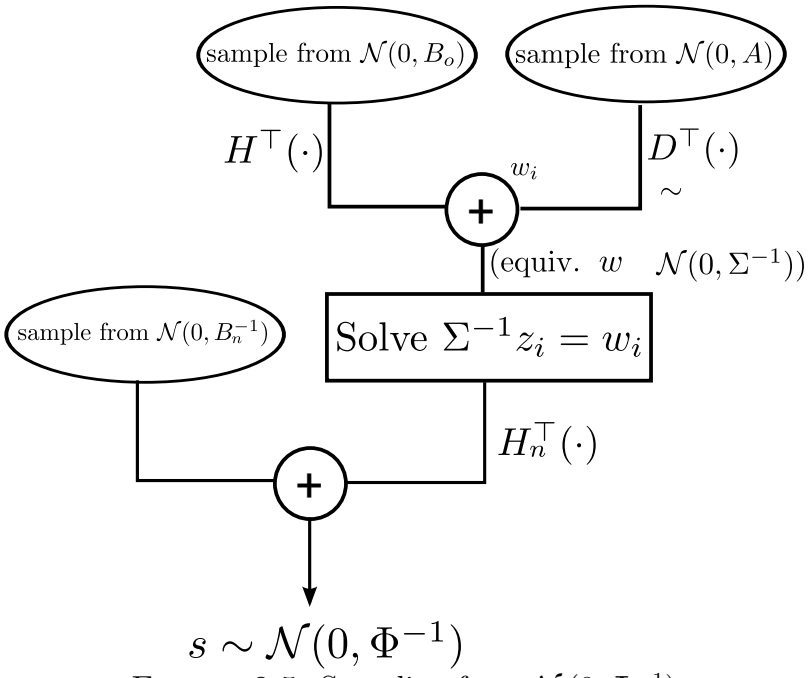


FIGURE 2.5: Sampling from $\mathcal{N}(0, \mathbf{\Phi}^{-1})$

where sampling from $\mathcal{N}(0, \mathbf{\Sigma})$ can be done using algorithm in Fig.2.4 and sampling from $\mathcal{N}(0, \mathbf{B}^{-1})$ is trivial as $\mathbf{B}$ is diagonal.

3

Results

Using proposed algorithms, we solve CT imaging problems where measurements are taken from 2D fan-beam CT system. All matrix-vector multiplications and forward/backward projection implemented using the Siddon's algorithm are implemented using CUDA C. The system we used throughout the experiment is equipped with 4 Intel(R) Xeon(R) CPU E7-4820 @ 2.00GHz and NVIDIA Kepler GK110 graphical processing unit.

3.1 Experimental Setup

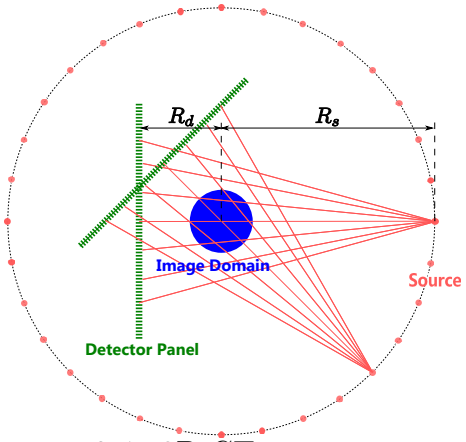
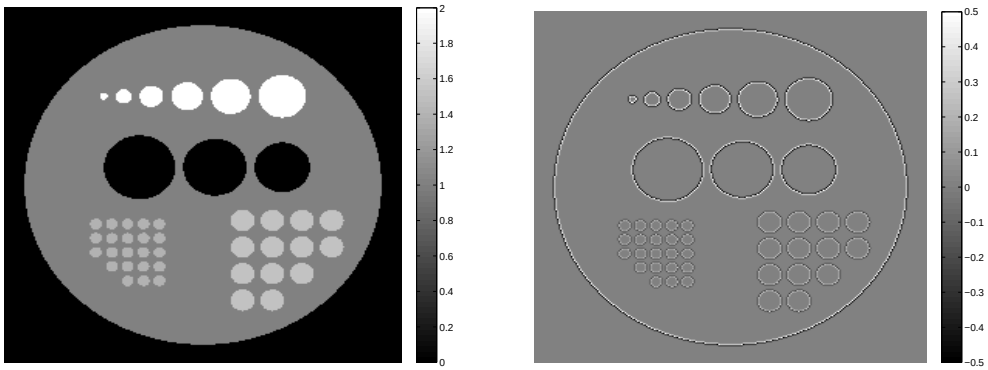


FIGURE 3.1: 2D CT system geometry

Fig.3.1 describes geometry of 2D fan-beam CT system. Each pixel in image domain contains discretized attenuation coefficient relative to water.

3.1.1 Datasets

Throughout the work, two datasets are used to test and analyze the result. The first dataset is the synthetic image with circular image domain.



(a) Image domain ($nz = 57\%$) (b) Difference domain ($nz = 7\%$)

FIGURE 3.2: Synthetic image and % of non-zero terms

Synthetic image have circular region of varying size and attenuation inside the

image domain. The image is not sparse in its native basis (i.e pixel domain), however, it does have sparse representation which can be obtained using difference operator described in Eq.2.24. In synthetic image reconstruction system, total of 360 source locations are even spread around the image and the number of detectors per each source was 780 ($M = 280,800$). log-transformed measurement using synthetic was obtained by using forward matrix to computed line-integral $H_i\mathbf{x}$ along each ray then adding zero mean Gaussian noise with variance $\sigma_i^2 = \frac{\exp(-H_i\mathbf{x})}{I_0}$ independently.

The second data set we used is an experimental data taken from Experimental x-ray Lab at Duke university. The target object is an acrylic glass phantom with linear attenuation coefficient value 1.5 times higher than that of water. The glass phantom contains clusters of rods with different size. The x-ray source specifications are 60kVp, 50mA, 25ms and a 0.55mm Cerium filter was used. Measurements were taken from 360 views, each with 1780 detectors.

3.2 Forward/Backward Projection

Forward/backward projections based on Siddon's method are implemented on both C and CUDA C code. For GPU parallelization, each ray is assigned to a single thread. Number of blocks and thread has been optimized to achieve maximum occupancy. Time required to perform forward/backward projection on each implementation is reported on Table.3.1. Times are measured by performing each operation 100

Table 3.1: Time required to perform forward/backward operation

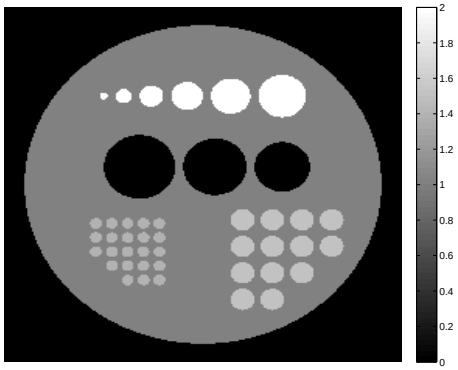
		CPU (C)	GPU (CUDA)
$\mathbf{H}$	Time (s)	2.42	0.022
	Gain	×1	×110
$\mathbf{H}^\top$	Time (s)	2.56	0.022
	Gain	×1	×115

times on each implementation and taking average. While the result from CPU C

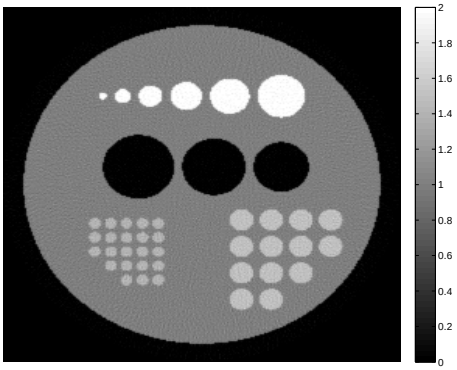
implementation doe not employ any sort of parallelization and therefore this result may not be a fair comparison, GPU implementation speeds up the entire operation by more than hundredfold which is still more than 3 times faster than theoretical maximum speed achievable using entire 32 cores of our system. Also, the $RMSE$ of the resulting vectors between two implementation was in the order of 10^{-4}.

3.3 Synthetic Data Reconstruction

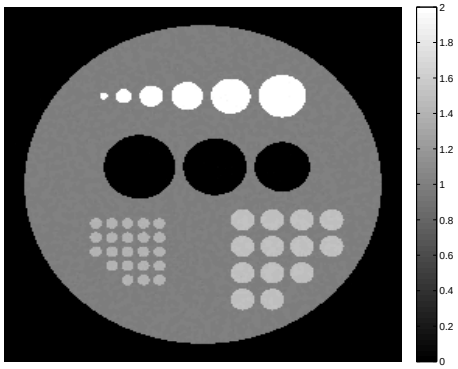
To examine the behavior of the proposed model, we first tested algorithm on small size synthetic images. On this test, entire reconstruction algorithm was implemented on CPU and instead of estimating variance, full covariance matrix was obtained using matrix inversion via Cholesky decomposition. SBL reconstruction is then compared to the most common analytical reconstruction algorithm: filtered-back-projection (FBP). 50 EM iterations have been performed to the final image.



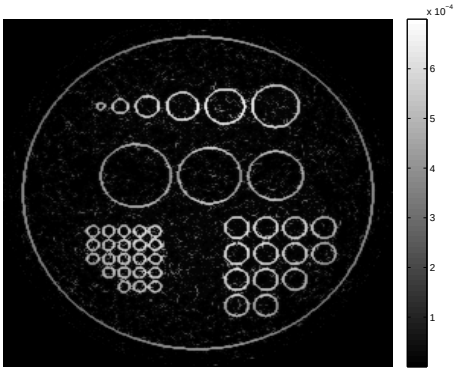
(a) True image



(b) FBP reconstruction ($RMSE = 13.7$)



(c) SBL reconstruction ($RMSE = 2.2$)



(d) Posterior standard deviation

FIGURE 3.3: Synthetic data reconstruction

Compared to FBP reconstruction, SBL reconstruction had less noise and the edges were sharp which suggest that piecewise-smoothness enforcing property of the proposed prior. This is more evident in cross-sectional view of reconstruction Fig.3.4

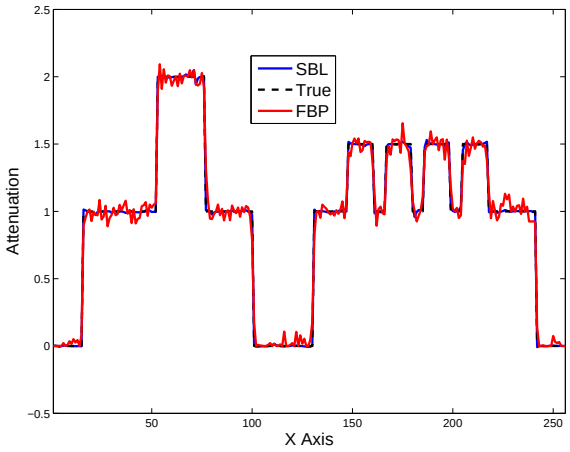


FIGURE 3.4: Cross-sectional view of synthetic data reconstruction

It is also important to note that even on first few iterations where the hyperparameter $\boldsymbol{\alpha}$ are not yet optimized, SBL performed better than FBP.

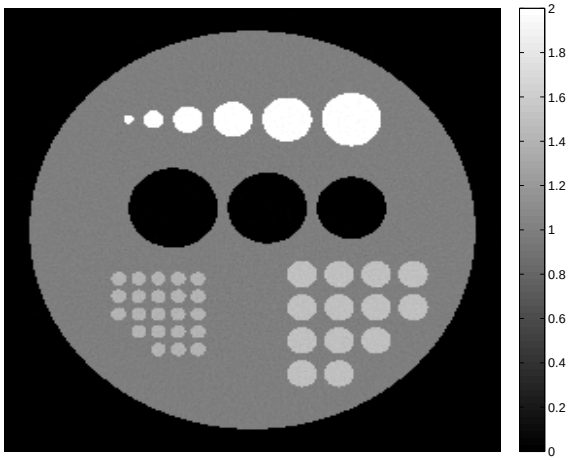
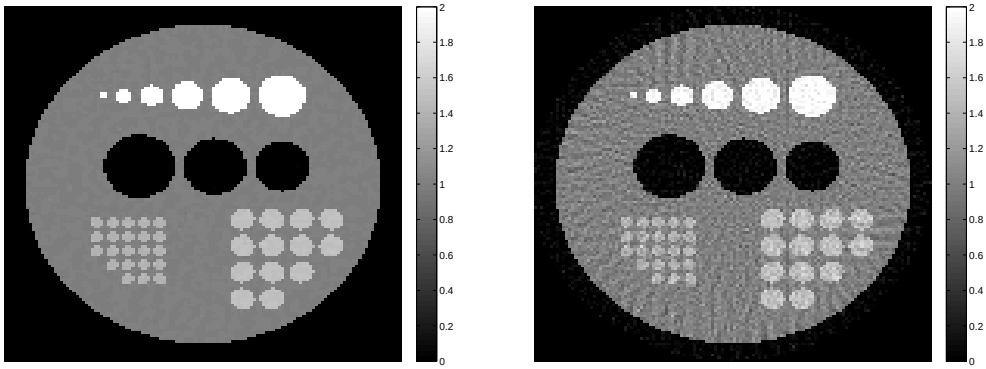


FIGURE 3.5: SBL reconstuction after 1 iteration ($RMSE = 4.2$)

It is known that analytic FBP reconstruction suffer from streaking artifacts when the measurements are noisy and incomplete while SBL reconstruction using wavelet is relatively robust to incomplete dataset and thereby applicable to compressed sensing in CT [1,28]. Measurements have been subsampled by the factor of 4 by choosing 90 view locations uniformly around the image. Result Fig.3.6 shows clear advantage of

SBL based reconstruction over FBP algorithm and together with the other results, could be used justify our selection of the difference prior.



(a) SBL reconstruction (b) FBP reconstruction

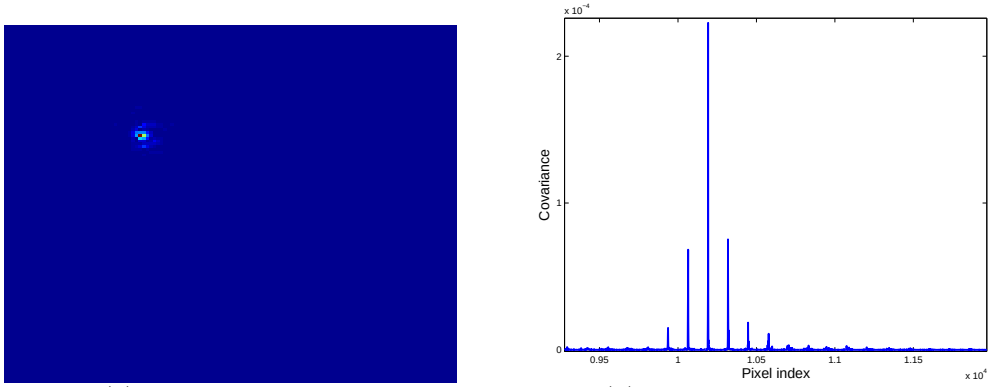
FIGURE 3.6: Synthetic data reconstruction with view subsampling (1/4 views)

3.4 Variance Estimation

Two classes of diagonal estimator have been implemented on GPU and their performance is evaluated. In diagonal estimator based on low-rank probing matrix, following probing matrix were used: (1) stochastic estimator proposed by Tang *et. al* [22]; (2) graph coloring method using with indicator variable (0,1) [22]; (3) graph coloring method using i.i.d random variable $\rho = \pm 1$ proposed by Malioutov [23, 24] In all implementation, linear equation $\mathbf{\Sigma x}_k = V_k$ was solved using preconditioned conjugate gradient on GPU. For the preconditioner $\mathbf{M}$, Jacobi preconditioner $\mathbf{M} = diag^{-1}(\mathbf{\Sigma})$ was used.

While the stochastic estimator and the sampling based diagonal estimators can be implemented straightforwardly, both of the graph coloring method requires one to know *a priori* the general sparsity pattern of the target matrix. Although our prior precision matrix follow GMRF with length 2 on Cartesian Grid, unlike the work in [23, 24], our observation is not localized and therefore the sparsity pattern of posterior covariance matrix could not be estimated solely based on prior precision

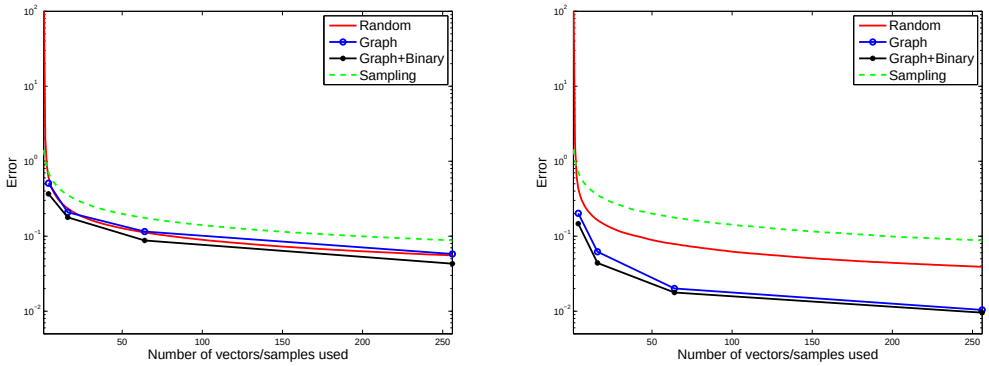
matrix. To examine the sparsity pattern of the covariance matrix, full covariance matrix has been obtained from small-scale synthetic data.



(a) 2D representation (b) plot of covariance value

FIGURE 3.7: 2D and plot representation of a column of full covariance matrix

Fig.3.7 shows a value of a column of a covariance matrix. As can be seen, covariance matrix is diagonally dominant, in other words, it has high variance term and the covariance decays rapidly as the separation between pixels increase. Also, outside the small localized region around the target pixel, covariance is very small or near zero. While this may provide general idea on how the sparsity pattern of full covariance matrix may look like, it is difficult to infer exact correlation length would be. Therefore instead of using standard graph coloring algorithm to find minimum number of required colors, we used $n_c \times n_c$ patch with $N_s (= n_c^2)$ different colors so that within those patches, all pixels are assigned to a different colors.



(a) First iteration (b) After 50th iteration

FIGURE 3.8: Performance of diagonal estimator at different EM iterations

Fig.3.8 shows the relative error of each variance estimator compared to direct matrix inversion vs. number of vectors/samples used. Here, blue line(Graph) corresponds to graph coloring method proposed by Tang, and black (Graph+binary) line corresponds Malioutov's method using ± 1 as a color indicator variable. The performance of graph coloring based estimators are expected to deteriorate at the initial iterations as initially, $\boldsymbol{\alpha}$ are initialized to 0.1, and the structure of covariance matrix depends largely on $\mathbf{H}^\top\mathbf{B}\mathbf{H}$ term which is not localized and therefore can not exploit the sparsity pattern. However, on later iterations, $\boldsymbol{\alpha}$ grows and $\mathbf{D}^\top\mathbf{A}\mathbf{D}$ becomes dominant term and the sparsity pattern is now becomes evident, making graph coloring based estimator preferable to other approaches. In addition to low error rate, graph coloring based estimator have other two other advantages. First, the preconditioned conjugate gradient involved in solving linear equation $\boldsymbol{\Sigma}\mathbf{x}_k = V_k$ converges much faster on graph-coloring based estimator (Fig3.9). This can be explained by the fact that a column of probing matrix $\mathbf{V}$ only contains small number of non-zero elements ($\#nz = \frac{N}{N_s}$).

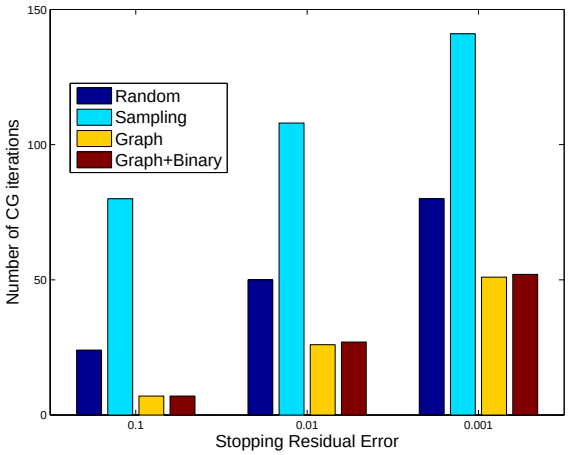


FIGURE 3.9: Number of CG iterations required for different number of vectors/samples

Second, while the convergence rate of error is inversely proportional to $\sqrt{N_s}$ for all techniques, on graph-coloring based estimator, the number of CG iterations decreases when more colors are used. Fig.3.10 shows the number of CG iteration required in solving $\mathbf{\Sigma x}_k = V_k$ using Malioutov's graph coloring method with different number of colors.

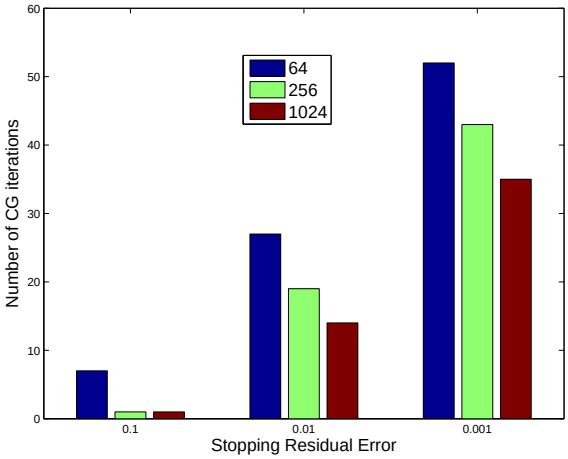


FIGURE 3.10: Number of CG iterations required for Graph-coloring based estimator

Real world experimental data with measurements consist of 640,800 rays (360 views; 1780 detectors) is reconstructed in high resolution image (480×480) using

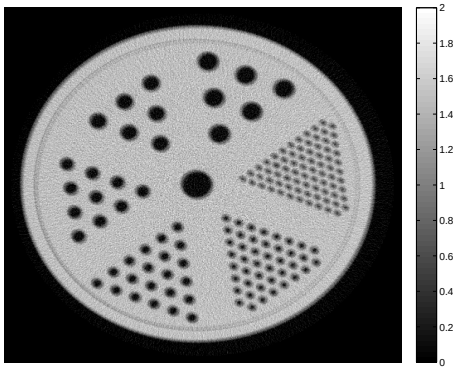
using Malioutov's Graph coloring estimator. Diagonal elements $\left[\mathbf{D\Sigma^{(t)}D^\top}\right]_{ii}$ may be estimated using Malioutov's Graph coloring estimator

$$diag(\mathbf{D\Sigma^{(t)}D^\top}) \approx \sum_{k=1}^{N_s} \mathbf{V_k} \odot \mathbf{Dx_k} \tag{3.1}$$

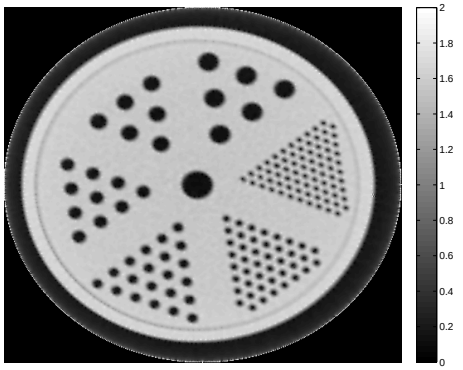
where $\mathbf{x_k}$ is the solution to

$$\mathbf{MP^t x_k} = \mathbf{MD^\top V_k} \tag{3.2}$$

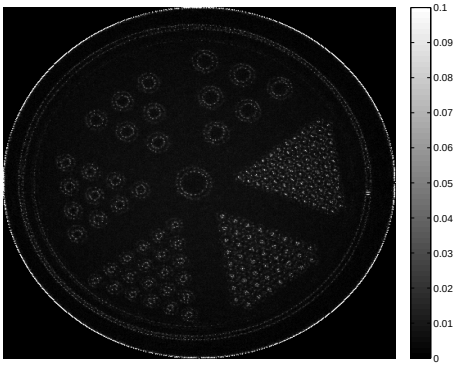
where $\mathbf{M}$ is the Jacobi preconditioner. $N_s = 64$ colors were used to estimate the variance, and total 50 iterations were performed to get the final reconstruction.



(a) FBP reconstruction



(b) SBL reconstruction

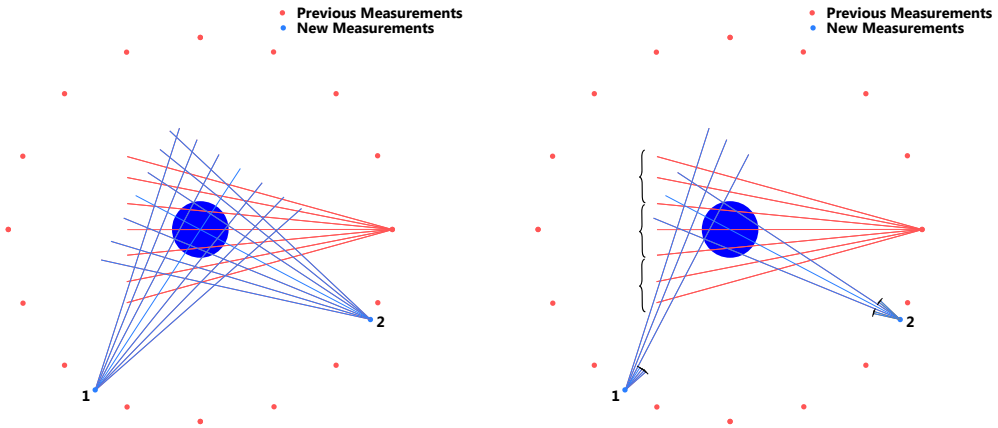


(c) Posterior standard deviation

FIGURE 3.11: Synthetic data reconstruction

3.5 Adaptive Sensing

To test the effectiveness of adaptive sensing using information gain(IG), two setups for adaptive sensing was tested on CPU with direct matrix inversion.



(a) Adaptive view selection (b) Adaptive partition selection

FIGURE 3.12: Adaptive sensing setup

On both setup, system is initialized with uniformly distributed sparse measurements (20 of 360 views). Once the IG for each candidates of selections are computed source location with highest IG is chosen and appended to the model. In setup 2. rays within each view is split into three partitions of rays. Then, similar to adaptive view selection, one can choose which partition to append to the model next. In addition to adaptive selection, sequential model update using randomly selected measurements (views/partitions) are performed. For view selection scheme, 20 new selections are made and for partition selection, 60 partitions are chosen.

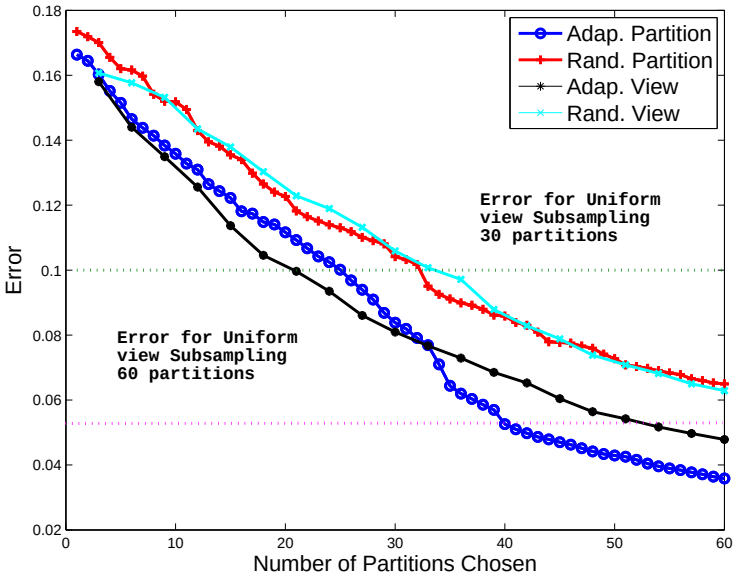


FIGURE 3.13: Performance of each adaptive sampling scheme

Fig.3.13 shows clearly the advantages of using adaptive selection over random selections or uniform subsampling. Finally information gain estimation using sampling method as proposed by Papandreou [27] is tested on GPU. The main difficulty in computing information gain arises in computing $\log|\mathbf{\Phi^{-1}}| = \log\left|\mathbf{B_{new}^{-1}} + \mathbf{H_{new}}\mathbf{\Sigma}_{\boldsymbol{old}}\mathbf{H_{new}^{\top}}\right|$

$$\log|\mathbf{\Phi}| \simeq \log|\mathbf{P}| - 2\log N_s - 2\log\sum_{i}^{N_s}\exp\left(\frac{1}{2}\tilde{s}_i{}^{\top}(\mathbf{\Phi}-\mathbf{P})\tilde{s}_i\right) \tag{3.3}$$

In order to estimate $\log|\mathbf{\Phi^{-1}}|$ efficiently and accurately , it is crucial find $\mathbf{P}$ that approximates $\mathbf{\Phi}$ reasonably well. For our system, two candidates of $\mathbf{P}$ are examined: (1) if the first first term in $\mathbf{\Phi^{-1}}$ is the dominant term $\mathbf{\Phi^{-1}}$ is diagonally dominant, $\mathbf{P} = \mathbf{B}$; (2) otherwise, we can use our diagonal estimator to estimate the Jacobi preconditioner $diag(\mathbf{B_{new}^{-1}} + \mathbf{H_{new}}\mathbf{\Sigma}_{\boldsymbol{old}}\mathbf{H_{new}^{\top}})$

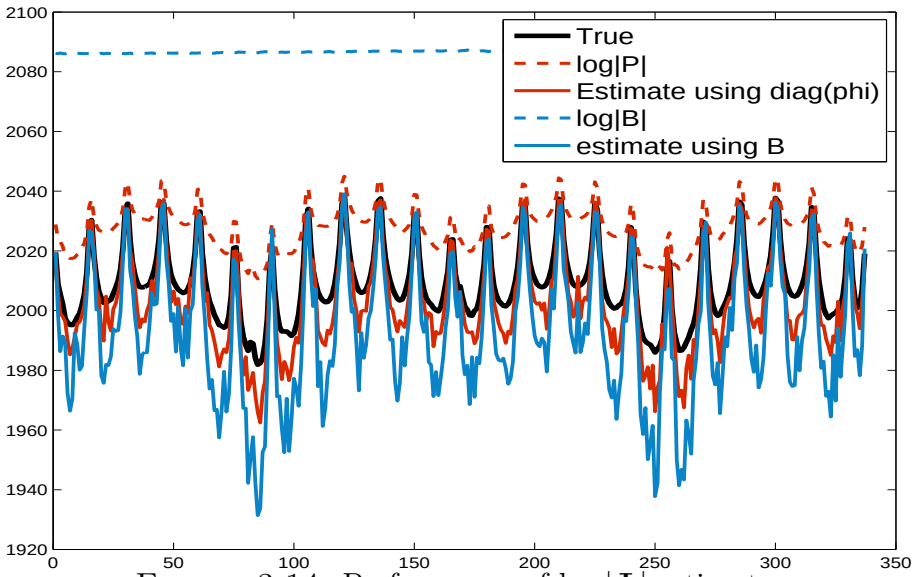


FIGURE 3.14: Performance of $\log|\mathbf{\Phi}|$ estimator

Fig.3.14 shows estimated $\log|\mathbf{\Phi^{-1}}|$ using two initial guesses. While using $\mathbf{B}$ as an initial guess is much easier as it is a diagonal matrix, the convergence rate is much slower and often suffer from numerical instability.

3.6 Conclusion and future works

Efficient CT image reconstruction algorithm based on popular sparse Bayesian learning model has been developed and test. Proposed algorithm use smooth penalties that promotes smooth image domain while preserving edge details and is applicable to many other image reconstruction or denoising problem.

Two scalable techniques for efficient variances estimation were studied. While the diagonal estimator using graph-coloring algorithm showed promising result in terms of both accuracy and speed, it is very problem specific and it still remains in question whether it can be applied to other imaging models. Sampling based estimator, on the other hand, suffered from slow convergence rate and low accuracy. However, it plays a critical role in an adaptive sensing CT system where the measurements are chosen sequentially based on the mutual information measure. Although the GPU based scalable experimental CT system has not been fully implemented and studied in this thesis, the results presented here clearly suggest the advantages of adaptive sensing in reducing the radiation dosage, and its viability.

Bibliography

[1] A. C. Kak and M. Slaney, *Principles of computerized tomographic imaging*. 2001.

[2] W. Kalender, "X-ray computed tomography," *Phys. Med. Biol.*, 2006.

[3] F. Natterer, *The mathematics of computerized tomography*. 1986.

[4] M. Li, H. Yang, and H. Kudo, "An accurate iterative reconstruction algorithm for sparse objects: application to 3D blood vessel reconstruction from a limited number of projections," *Phys. Med. Biol.*, 2002.

[5] M. Sonka and J. Fitzpatrick, "Handbook of medical imaging(Volume 2, Medical image processing and analysis)," 2000.

[6] M. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine," *J. Mach. Learn. Res.*, 2001.

[7] Y. Saad, *Iterative methods for sparse linear systems*. 2003.

[8] E. Simoncelli, "Modeling the joint statistics of images in the wavelet domain," in *SPIE's Int. Symp. Opt. Sci. Eng. Instrum.*, 1999.

[9] M. Seeger, "Bayesian inference and optimal design for the sparse linear model," *J. Mach. Learn. Res.*, 2008.

[10] R. Szeliski, "Bayesian modeling of uncertainty in low-level vision," *Int. J. Comput. Vis.*, 1990.

[11] S. J. Lee, I. T. Hsiao, G. R. Gindi, and T. Hsiao, "Quantitative effects of using thin-plate priors in Bayesian SPECT reconstruction," in *Opt. Sci. Eng. Instrumentation'97*, 1997.

[12] S. Lee, A. Rangarajan, and G. Gindi, "Bayesian image reconstruction in SPECT using higher order mechanical models as priors," *Med. Imaging, IEEE Trans.*, 1995.

[13] R. Siddon, "Fast calculation of the exact radiological path for a threedimensional CT array," *Med. Phys.*, 1985.

[14] G. Han, Z. Liang, and J. You, "A fast ray-tracing technique for TCT and ECT studies," in *Nucl. Sci. Symp. 1999. Conf. Rec. 1999 IEEE*, 1999.

[15] H. Nickisch and R. Pohmann, "Bayesian experimental design of magnetic resonance imaging sequences," in *Adv. Neural Inf. Process. Syst.*, pp. 1441–1448, 2009.

[16] D. Malioutov, J. Johnson, and A. Willsky, "Walk-sums and belief propagation in Gaussian graphical models," *J. Mach. Learn. Res.*, vol. 7, no. 2031-2064, 2006.

[17] Y. Weiss and W. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," *Neural Comput.*, 2001.

[18] C. Paige and M. Saunders, "LSQR: An algorithm for sparse linear equations and sparse least squares," *ACM Trans. Math. Softw.*, vol. 8, no. 1, pp. 43–71, 1982.

[19] M. W. M. Seeger and H. Nickisch, "Large scale Bayesian inference and experimental design for sparse linear models," *SIAM J. Imaging Sci.*, vol. 4, no. 1, pp. 166–199, 2011.

[20] M. Hutchinson, "A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines," *Commun. Stat. Comput.*, vol. 18, no. 3, 1989.

[21] C. Bekas, E. Kokiopoulou, and Y. Saad, "An estimator for the diagonal of a matrix," *Appl. Numer. Math.*, 2007.

[22] J. Tang and Y. Saad, "A probing method for computing the diagonal of a matrix inverse," *Numer. Linear Algebr. with Appl.*, pp. 1–15, 2010.

[23] D. Malioutov and J. Johnson, "Low-Rank Variance Approximation in GMRF Models : Single and Multiscale Approaches," *Signal Process. IEEE Trans.*, vol. 56, no. 10, pp. 4621–4634, 2008.

[24] D. Malioutov, "Low-rank variance estimation in large-scale GMRF models," in *Acoust. Speech Signal Process. 2006. ICASSP 2006 Proceedings. 2006 IEEE Int. Conf.*, vol. 2, 2006.

[25] M. Luby, "A simple parallel algorithm for the maximal independent set problem," *SIAM J. Comput.*, 1986.

[26] G. Papandreou and A. L. Yuille, "Gaussian sampling by local perturbations," in *Adv. Neural Inf. Process. Syst.*, pp. 1–9, 2010.

[27] G. Papandreou and A. Yuille, "Efficient variational inference in large-scale Bayesian compressed sensing," in *Comput. Vis. Work. (ICCV Work. 2011 IEEE Int. Conf.*, pp. 1332–1339, July 2011.

[28] Y. Kaganovsky, D. Li, and A. Holmgren, "Compressed sampling strategies for tomography," *JOSA A*, 2014.