

PARALLEL COMPUTATIONAL METHODS FOR
MODEL-BASED TOMOGRAPHIC RECONSTRUCTION AND COHERENT
IMAGING

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Venkatesh Sridhar

In Partial Fulfillment of the
Requirements for the Degree

of

Doctor of Philosophy

May 2020

Purdue University

West Lafayette, Indiana

**THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL**

Dr. Charles A. Bouman, Co-chair

School of Electrical and Computer Engineering

Dr. Gregory T. Buzzard, Co-chair

Department of Mathematics

Dr. Samuel P. Midkiff

School of Electrical and Computer Engineering

Dr. Mark R. Bell

School of Electrical and Computer Engineering

Dr. Mary Comer

School of Electrical and Computer Engineering

Approved by:

Dr. Dimitrios Peroulis

Head, School of Electrical and Computer Engineering

Dedicated to my family.

ACKNOWLEDGMENTS

I profoundly thank my co-advisors Prof. Charles Bouman and Prof. Gregory Buzzard for their guidance and support in my PhD research. The opportunity to work on inter-disciplinary problems spanning multiple areas such as computational imaging, machine-learning, and parallel-programming would have not been possible without their passion and expertise. Prof. Bouman's high standards in research have often been tough to cope with, but have also made this experience rewarding. Further, his emphasis on explaining complex concepts in a brief manner and thinking beyond conventional wisdom have shaped this thesis. Prof. Buzzard's work on multi-agent consensus algorithms has laid the foundation for a major part of this thesis. His supervision has been crucial in developing mathematical theorems and proofs that are integral to this research.

I thank Prof. Samuel Midkiff for his valuable advice and his course on parallel programming, which has been instrumental for implementing computational methods developed in this research.

I am grateful to Sherman J. Kisner for being an astounding mentor and research collaborator. I have immensely benefited from his patient and hands-on guidance, specially during the initial stage of my research. Further, his ability to quickly turn around efficient and modular software has inspired me to meet similar standards.

I would like to express my deep gratitude towards Prof. Anand Raghunathan. Mid-way through my PhD program, I evinced interest in research intersecting the diverse areas of computational imaging and hardware acceleration. Despite my lack of prior experience in the latter, he reposed faith in my abilities to cover necessary ground. His course on SoC design is one of the most enjoyable courses that I've taken.

I am fortunate to have collaborated with Xiao Wang, who lead the supercomputing experiments in this research. Working closely with him has provided me practical experience in the field of high-performance computing.

I would like to appreciate Prof. Mark Bell and Prof. Mary Comer for their constructive feedback as members on my thesis committee. I would like to thank the U.S. Department of Homeland Security (DHS) Science and Technology Directorate for their research grants that have supported most of my thesis work.

I would like to specifically thank S.V.Venkatakrishnan for his passionate advice and encouragement during multiple moments in my research. Further, I am glad for the jovial banter, stimulating technical discussions, and the lively company of my former lab-mates Aditya, Dilshan, Zeeshan, Shruthi, and Suhas, as well as my current lab-mates Soumendu, Thilo, Camilo, Maliha, Emma, Madhuri, Abdu, Wenrui, Ziyu, and Diyu, that gave me much needed respite from work.

I am thankful for my elder brother, Vignesh Sridhar, whose consistent support and motivation has helped me overcome personal tough times. I am grateful to my parents, Dr. Geetha and Vaidyanathan Sridhar, for their unflinching support towards my graduate education and general well being. Last but not the least, I would like to thank all my friends who have enriched my experience at Purdue.

TABLE OF CONTENTS

	Page
LIST OF TABLES	viii
LIST OF FIGURES	ix
ABBREVIATIONS	xii
ABSTRACT	xiii
1 Model-Based Reconstruction for X-ray Diffraction Imaging	1
1.1 Introduction	1
1.2 XRD Forward Model	2
1.2.1 Fundamentals of Coherent Scatter Detection	2
1.2.2 Data Collection and Preprocessing	5
1.2.3 Point Spread Function	8
1.2.4 Discretized Forward Model	9
1.3 Statistical Model and Reconstruction Approach	11
1.3.1 Likelihood function	11
1.3.2 Prior model	12
1.3.3 Iterative coordinate descent	13
1.4 Results and Discussion	14
1.5 Conclusion	19
2 Distributed Iterative CT Reconstruction using Multi-Agent Consensus Equilibrium	20
2.1 Introduction	20
2.2 Distributed CT Reconstruction using Conventional Priors	23
2.2.1 CT Reconstruction using MAP Estimation	23
2.2.2 MACE framework	25
2.2.3 Partial Update MACE Framework	28

	Page
2.3 MACE with Plug-and-Play Priors	31
2.4 Experimental Results	34
2.4.1 Methods	35
2.4.2 MACE Reconstruction of 2-D CT Dataset	38
2.4.3 MACE Reconstruction for large 3D datasets	44
2.5 Conclusion	44
3 Fast Algorithms for Model-based Imaging through Turbulence	46
3.1 Introduction	46
3.2 Stastical Framework for DH-MBIR	49
3.2.1 DH Forward Model	49
3.2.2 MAP Estimation for DH Reconstruction	49
3.3 Accelerating MBIR	50
3.3.1 Fast Method for Reflectance Update	50
3.3.2 Fast Method for Phase-error Update	53
3.3.3 SIMD Parallelization of ICD updates	54
3.4 Experimental Results	57
3.4.1 Method	57
3.4.2 Results	58
3.5 Conclusion	62
REFERENCES	63
A MACE Formulation as a Fixed-point Problem	68
B Convergence of Partial-update MACE Approach	69
C Deriving the MACE-PnP Approach	75
D EM Surrogate Function	79
VITA	80

LIST OF TABLES

Table	Page
2.1 CT Dataset Description	35
2.2 MACE convergence (equits) for different values of ρ , $N=16$	37
2.3 MACE memory usage for the system-matrix (gigabytes) as a function of number of nodes, N	39
2.4 Computational performance for 3-D MACE reconstruction on the NERSC supercomputer as a function of #view-subsets	43
3.1 CPU Time per EM iteration	59
3.2 Speedup of Fast DH-MBIR over Exact DH-MBIR	59

LIST OF FIGURES

Figure	Page
1.1 Illustration of X-ray diffraction and transmission. The solid red line depicts the trajectory of photons scattered at angle θ , whereas the solid green line represents the direct path. The dotted red lines indicate the angle subtended by the detector at the voxel.	3
1.2 Schematic depicting the basic geometry of the XDi system. The measurement of scatter is localized to specific points in object-space, or scatter-centers, which are highlighted in red.	5
1.3 Schematic depicting the spatial spread of a scatter-center due to the finite aperture of the detector collimation. The zoomed in section on the left illustrates the PSF of the scatter-center on the imaging plane, where the different “rings” or contours correspond to different PSF values.	9
1.4 Basis spectra obtained through NNMF decomposition of the training data, when number of basis components, N_p is 4	10
1.5 Model-based scatter strength reconstructions of an actual bag, with its prescan image depicted in (a). All slices shown are coronal, i.e. looking down on the scanner tunnel. In (d) and (f), the voxel size is halved in all 3-dimensions. The number of basis spectra N_p was chosen to be 4.	17
1.6 (a) -(d): Reconstructed diffraction profiles of two liquids A and B, contained within the bottles of the scanned bag. (e): Histogram for classification of the two liquids based on XDi scans of 82 different bags; x-axis represents a linear projection of the 4 spectral coefficients and y-axis represents the fraction of scanned bags.	18

Figure	Page
2.1 Illustration of the MACE algorithm for distributing CT reconstruction across a parallel cluster of compute nodes. (a) An illustration of the MACE algorithm using a conventional prior model. The MACE algorithm works by splitting the data into view subsets and reconstructing them in parallel. The individual reconstructions are then merged in an iterative loop that results in the true MAP reconstruction for the full data set. (b) An illustration of the MACE-PnP algorithm which extends the MACE framework to the use of Plug-n-Play prior models to improve reconstruction quality. In this case, a denoiser is run in the MACE loop in order to implement an advanced prior model. The MACE and MACE-PnP algorithms distribute both computation and memory across parallel clusters of computers, thereby enabling the reconstruction of large tomographic data sets.	24
2.2 Single node reconstruction for (left) Low-Res. Ceramic Composite dataset (right) Baggage Scan dataset.	37
2.3 Comparison of reconstruction quality for MACE method using 16 parallel nodes each processing $(1/16)^{th}$ of the views, against centralized method. (a) and (c) Centralized method. (b) and (d) MACE. Notice that both methods have equivalent image quality.	38
2.4 MACE convergence for different number of nodes, N , using $\rho = 0.8$: (a) Low-Res. Ceramic composite dataset (b) Baggage Scan dataset. Notice that number of equits tends to gradually increase with number of parallel processing nodes, N	39
2.5 MACE-PnP reconstruction of Baggage Scan data set: (a) MACE PnP reconstruction using $N = 16$ nodes; (b) Zoomed-in regions of PnP versus conventional prior; (c) MACE-PnP convergence as a function of number of nodes N . Notice that PnP prior produces better image quality with reduced streaking and artifacts. In addition, the number of equits required for convergence of the MACE-PnP does not tend to increase significantly with number of nodes N	40
2.6 MACE speedup as a function of the number of nodes, for different datasets and prior models. Importantly, note that in the case of PnP priors, we achieve a near linear speedup for both datasets.	41
2.7 Comparison of quality between (a),(c) fully converged result and (b),(d) the MACE reconstruction using 8 view-subsets on the NERSC supercomputer. Notice that both have equivalent image quality.	42

Figure	Page
2.8 MACE convergence for the High-Res. ceramic reconstruction on the NERSC supercomputer as a function of view-subsets (reconstruction size $1280 \times 1280 \times 1200$).	43
3.1 Illustration of a Digital Holography system for remote-sensing. A laser source illuminates the target and a lens focuses the reflected light onto the FPA. The detected field is optically heterodyned with a strong reference field to form a hologram. However, atmospheric turbulence between the target and lens can corrupt the pupil-plane field by inducing phase-errors. These phase-errors must be removed prior to recovering the target image from the hologram.	48
3.2 Approximation of the wrapped-phase penalty function $\rho(u - \phi_j), j \in \partial s$, for the s -th pixel ICD update. Sub-figures (a) and (b) show how the approximation varies with different values of (ϕ_j, ϕ_s)	55
3.3 SIMD Parallelization of ICD updates: (a) A tiled grid where pixels of the same color have independent ICD updates (b) SIMD processing units of a CPU can be used to update multiple pixels of the same color at a time using fast vector add and multiply operations.	56
3.4 Example depicting object reflectance, r and simulated reflection coefficient, g . Unlike $ g ^2$ (right), the reflectance r (left) is speckle-free and smooth.	58
3.5 Effect of fast updates for estimating reflectance, r , and phase-error, ϕ , on the convergence rate of DH-MBIR. The number of iterations for DH-MBIR convergence is unchanged when we use fast updates for ϕ estimation instead of exact updates, but increases when we use fast updates for r estimation in place of exact updates.	60
3.6 Comparison of reconstruction quality for 6 different simulated datasets. For each dataset, the reflectance estimate without any phase recovery is shown the top row of sub-figures (a) and (b) respectively, and the corresponding phase-errors in the pupil-plane are shown in the top row of sub-figures (c) and (d). In all sub-figures, the middle row shows reconstruction using the exact MBIR approach while the bottom rows shows reconstruction using the fast MBIR approach. For most datasets, the reconstructions using the fast MBIR method are almost indistinguishable from the exact MBIR method.	61

ABBREVIATIONS

MBIR	Model-based Iterative Reconstruction
XRD	X-ray Diffraction
MAP	Maximum-a-posteriori
SNR	Signal-to-Noise ratio
NNMF	Non-negative Matrix Factorization
MRF	Markov random field
ICD	Iterative Coordinate Descent
CT	Computed Tomography
MACE	Multi-Agent Consensus Equilibrium
PnP	Plug-n-play
MPI	Message Parsing Interface
SV	Super-voxel
DH	Digital Holography
EM	Expectation-Maximization
SIMD	Single Instruction Multiple data
FPA	Focal plane array
ISR	Intelligence, reconnaissance and surveillance

ABSTRACT

Sridhar, Venkatesh Ph.D., Purdue University, May 2020. Parallel Computational Methods for Model-Based Tomographic Reconstruction and Coherent Imaging. Major Professors: Charles A. Bouman, Gregory T. Buzzard.

Non-destructive imaging modalities for evaluating the internal properties of materials can be formulated as physics-driven inverse problems. Model-based Iterative reconstruction (MBIR) methods that integrate a forward model of the imaging system and a prior model of the object being imaged can provide superior reconstruction quality relative to conventional methods. However, making MBIR feasible for practical applications faces two key challenges. First, we require efficient computational methods for MBIR that allow large-scale reconstructions in real-time. Second, we must develop forward models that accurately capture the physics and geometry of the imaging system, and, support the use of advanced denoisers that enhance image quality as prior models.

This thesis attempts to address the aforementioned challenges and is divided into three main chapters, each corresponding to a different inverse imaging application.

In the first chapter of this thesis, we propose a novel 4D model-based iterative reconstruction (MBIR) algorithm for low-angle coherent-scatter X-ray Diffraction (XRD) tomography that can substantially increase the SNR. Our forward model is based on a Poisson photon counting model that incorporates a spatial point-spread function, detector energy response and energy-dependent attenuation correction. Our prior model uses a Markov random field (MRF) together with a reduced spectral bases set determined using non-negative matrix factorization. Our algorithm efficiently computes the Bayesian estimate by exploiting the sparsity of the measurement data. We demonstrate the ability of our method to achieve sufficient spatial resolution

from sparse photon-starved measurements and also discriminate between materials of similar densities with real datasets.

In the second chapter of this thesis, we propose a multi-agent consensus equilibrium (MACE) algorithm for distributing both the computation and memory of MBIR for Computed Tomographic (CT) reconstruction across a large number of parallel nodes. In MACE, each node stores only a sparse subset of views and a small portion of the system matrix, and each parallel node performs a local sparse-view reconstruction, which based on repeated feedback from other nodes, converges to the global optimum. Our distributed approach can also incorporate advanced denoisers as priors to enhance reconstruction quality. In this case, we obtain a parallel solution to the serial framework of Plug-n-play (PnP) priors, which we call MACE-PnP. In order to make MACE practical, we introduce a partial update method that eliminates nested iterations and prove that it converges to the same global solution. Finally, we validate our approach on a distributed memory system with real CT data. We also demonstrate an implementation of our approach on a massive supercomputer that can perform large-scale reconstruction in real-time.

In the third chapter of this thesis, we propose a method that makes MBIR feasible for real-time single-shot holographic imaging through deep turbulence. Our method uses surrogate optimization techniques to simplify and speedup the reflectance and phase-error updates in MBIR. Further, our method accelerates computation of the surrogate-updates by leveraging cache-prefetching and SIMD vector processing units on a single CPU core. We analyze the convergence and real CPU time of our method using simulated datasets, and demonstrate its dramatic speedup over the original MBIR approach.

1. MODEL-BASED RECONSTRUCTION FOR X-RAY DIFFRACTION IMAGING

1.1 Introduction

X-ray Diffraction (XRD) Tomography is emerging as an important imaging modality for transportation security applications. Recently, baggage scanners equipped with both dual-energy Computed Tomography (CT) and XRD modalities have been deployed at airports for the inspection of carry-on luggage. XRD scanners reconstruct the voxel-wise spectral profile which is characteristic of molecular composition of the material [1–3]. In contrast to conventional CT imaging, this gives XRD the ability to even distinguish between materials with similar densities based on their diffraction profiles.

However, current X-ray diffraction imaging (XDi) system designs often face challenges in achieving sufficient spatial resolution and signal-to-noise ratio (SNR) due to the limited number of detected photons. One important consideration in XDi scanner design is the tradeoff between low-angle scatter detection versus wide-angle scatter detection. Wide-angle detection systems can accept scatter at high angles thereby achieving higher photon counts. However, wide-angle systems also require attenuation correction from an auxiliary X-ray transmission scan. Low-angle detectors have the important advantage that both the scattered photons and the directly transmitted photons travel along similar paths. This allows autonomous correction for attenuation, but it also reduces the photon count, making it more difficult to achieve the needed signal-to-noise ratio and resolution.

In this thesis, we deal with a 3rd generation XDi scanner that implements the Multiple Inverse Fan-Beam (MIFB) geometry [2]. In stark comparison to 1st generation XDi topologies which scan only a single object point in time, 3rd generation

systems have the capability to acquire highly-parallel measurements [2]. This particular advantage translates into drastic reductions in scan times of checked baggage. We propose a novel 4D model-based iterative reconstruction (MBIR) algorithm for low-angle scatter XRD that can substantially increase the SNR by fully accounting for both the forward model of the scanner and the prior model of the image [4]. Our forward model is based on a Poisson photon counting model that utilizes a spatial point-spread function to incorporate the scanner-geometry and corrects for the energy-dependant attenuation of the X-ray spectrum along the path from source to detector. Our prior model is based on the combination of a Markov random field (MRF) spatial prior together with a reduced non-negative spectral bases set. Using these models, we compute the maximum a posteriori (MAP) estimate to render a 4D reconstruction in space and momentum spectrum. The computation of the MAP estimate is based on an iterative voxel-wise update strategy similar to that considered by *Bouman et al.* [5] and exploits the sparsity in photon-count measurements. We validate the performance of our reconstruction technique with actual baggage data.

1.2 XRD Forward Model

1.2.1 Fundamentals of Coherent Scatter Detection

XDi systems utilize a range of X-ray photon energies that result in coherent scatter. In this type of X-ray interaction with matter, the incident photon does not have enough energy to ionize electrons from the atomic shell of the medium. This leads to a change in direction of the photon, but its energy remains the same.

Figure 1.1 illustrates the diffraction of an incident X-ray beam by a single voxel. Photons that are deflected by an angle θ are sensed by a detector. Furthermore, let us denote the solid angle subtended by the detector as Ω . Notice that this solid angle is dependent on both the voxel location and the detector position.

Let us assume that the voxel contains a homogenous material. Furthermore, let us assume that this material is amorphous or polycrystalline with crystal grains at

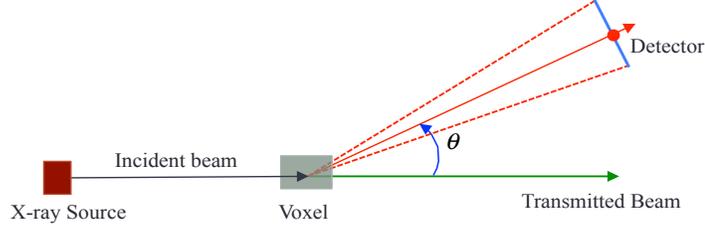


Fig. 1.1. Illustration of X-ray diffraction and transmission. The solid red line depicts the trajectory of photons scattered at angle θ , whereas the solid green line represents the direct path. The dotted red lines indicate the angle subtended by the detector at the voxel.

random orientations that are small compared to the size of a voxel. In this general case, intensity of the scattered X-rays will generally be proportional to the quantity

$$X_g(E, \theta) = n_p \frac{d\sigma_p}{d\Omega}(E, \theta)$$

where n_p is the number density of constituent particles in the voxel (units of *particles/cm*³), and $\frac{d\sigma_p}{d\Omega}$ is the differential cross-section of the constituent particles (units of *cm*²/*(sr – particle)*). Notice that the differential cross-section represents the scatter cross-section per steradian of scatter angle. Since the material composition of the object can change from voxel to voxel, the differential cross-section is a function of r as well as the energy of the photon E and the scatter angle θ . From this definition, we see that X has units of $1/(\text{sr} - \text{cm})$.

For the aforesaid type of material, it is commonly assumed that X is only a function of the momentum transfer, which is defined as

$$q = \frac{E}{hc} \sin\left(\frac{\theta}{2}\right). \quad (1.1)$$

This is a very important constraint since this means that we need must only reconstruct the quantity

$$X(q) = X_g(E, \theta),$$

where q is given as a function of E and θ .

Let $\lambda(E)$ be the detected photon rate for photons at a energy E . Then in general, $\lambda(E)$ is given by

$$\lambda(E) = I_s(E) A_I(E) X(q) A_D(E) \Omega V \quad (1.2)$$

where $I_s(E)$ is the source intensity at energy E , $A_I(E)$ is the source to voxel attenuation, $A_D(E)$ is the voxel to detector attenuation, Ω is the solid angle subtended by the detector, and V is the volume of the voxel. Here the photon intensity is $I_s(E)$ has units of *photons/cm²*, and both $A_I(E)$ and $A_D(E)$ are unitless, so $\lambda(E)$ has units of *photons*.

For our applications, θ is very low, typically in the range of 0 to 50 milli-radians. This implies that the attenuation along the path from source to detector is approximately same as that experienced by the directly transmitted beam [3,6]. So in the low angle case this approximation can be written as

$$I_s(E) A_I(E) A_D(E) \approx f(E) , \quad (1.3)$$

where $f(E)$ is the spectrum of the detected X-ray photons along the direct path. In practice, $f(E)$ must be measured for each beam since the attenuation will be object dependent. The details on how $f(E)$ is measured are described in the following section.

The product of the solid angle of scatter, and the voxel volume form a single constant that is not a function of energy given by

$$\beta = \Omega V .$$

In practice, β can be determined through a calibration procedure since it is only a function of the scanner geometry and not a function of the object being scanned.

So then the final relationship between the unknown X and the measured photon rate at energy E is given by

$$\lambda(E) = \beta f(E) X(q) \quad (1.4)$$

1.2.2 Data Collection and Preprocessing

Figure 1.2 illustrates the geometry of the XDi system. An array of sources are used to image a 2D plane through the object, which in this case, is the x - z plane. Each source is collimated into a fan of X-ray pencil beams that pass through the object. However, Figure 1.2 merely shows what happens along one such pencil beam. A dual-energy X-ray detector measures the flux of the directly transmitted beam. An array of scatter detectors are collimated so that each detector measures the scatter at one point along the pencil beam. The collimation is performed so that the angle of scatter, θ , is relatively small and approximately constant for all the detectors.

We will refer to the point of intersection of the pencil beam and the detector collimation as a scatter center. The scatter center represents the location of material being measured. We will index each scatter center by o , and we note that the scatter center is specified by a combination of the source s and the detector d . Equivalently, each scatter center o uniquely specifies a source-detector pair (s, d) . In addition, the object being imaged is moved through the XDi system with a belt. So each scatter center belongs to a slice through the object.

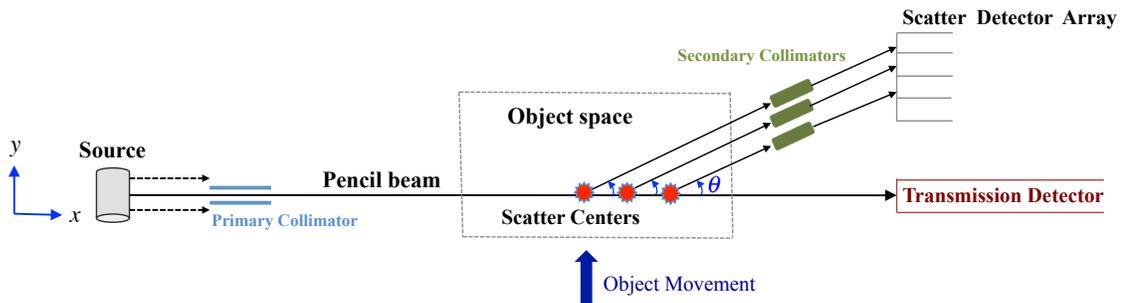


Fig. 1.2. Schematic depicting the basic geometry of the XDi system. The measurement of scatter is localized to specific points in object-space, or scatter-centers, which are highlighted in red.

1.2.2.1 XDi List-mode Data

In order to maximize signal-to-noise, the detectors count photons and report an energy for each detected photon. This data is reported in list-mode format; so a list of photon counts with associated energy are reported for each scatter center.

More specifically, for a given scatter-center, o , the list-mode measurement data provides us three pieces of information represented by $\{N_o, \theta_o, E_o\}$. Here, N_o denotes the number of detected photons from the o -th scatter center, and $E_o \in \mathbb{R}^{N_o}$ is a vector whose entry $E_{o,i}$ represents the energy of the i -th detected photon, $1 \leq i \leq N_o$. The scatter-angle may deviate slightly from the nominal angle $\bar{\theta}$ and its value for the o -th scatter center is denoted by $\theta_o \in \mathbb{R}^+$.

For each photon energy, $E_{o,i}$, we may compute an associated momentum transfer value that accounts for that the specific detector's scatter angle.

$$q_{o,i} = E_{o,i} \frac{\sin(\theta_o/2)}{hc} \quad (1.5)$$

The momentum transfer value is then quantized by mapping it to the closest of a set of N_M values given by $\{\tilde{q}_m\}_{m=1}^{N_M}$. Let $M_{o,i}$ be this discrete quantization level. Then the photon count for the o -th scatter-center and m -th momentum bin, $\lambda_{o,m}$, can then be expressed as

$$\lambda_{o,m} = \sum_{i=1}^{N_o} \delta(M_{o,i} - m) , \quad (1.6)$$

where the function $\delta(\cdot)$ is defined as 1 when argument is 0, and 0 otherwise. So intuitively, $\lambda_{o,m}$ is the number of photons from scatter center o that fall into momentum transfer bin m .

Assuming that N_M is sufficiently large, detected photons from a given scatter-center o that fall within the same momentum bin m , have approximately the same energy. Accounting for the detector-specific scatter angle θ_o , we can express this discrete energy $\tilde{E}_{o,m}$ as follows

$$\tilde{E}_{o,m} = \tilde{q}_m \frac{hc}{\sin(\theta_o/2)} \quad (1.7)$$

1.2.2.2 Attenuation Correction

For each scatter-center o , we estimate the momentum-spectrum of photons along the direct transmission path, represented by the vector $I_o \in \mathbb{R}^{N_M}$. However, note that this spectrum, I_o , accounts for the energy-dependent attenuation undergone by the detected photons. So, the m -th entry of I_o is expressed as

$$I_{o,m} = f_o(\tilde{E}_{o,m}) \quad (1.8)$$

where $f_o(\cdot)$ is a function specific to the incident pencil-beam, since the attenuating medium varies for different beam-paths. The above function is common to all scatter-centers that share the same transmission path.

We derive $f_o(\cdot)$ from the dual-energy flux measured by the transmission detector. This is accomplished through techniques similar to *Beam-Hardening* correction for dual-energy CT. A common approach is to select two basis materials with markedly different attenuation characteristics, and then decompose the medium along the direct beam path into these basis materials [7,8]. For this approach, $f_o(E)$ can be determined in accordance with Beer's law, as follows

$$f_o(E) = \tilde{I}_s(E) \exp \left\{ - \sum_{j=1}^2 d_{j,o} \mu_j(E) \right\} \quad (1.9)$$

where $d_{j,o}$ and $\mu_j(E)$ represent the effective thickness and attenuation coefficient of the j^{th} basis material respectively, and $\tilde{I}_s(E)$ denotes the source X-ray spectrum.

Estimating $d_{j,o}$, involves a two-step procedure. Let the pair $\{T_{L,o}, T_{H,o}\} \in \mathbb{R}^2$ represent the dual-energy flux measured for the transmission path of the o -th scatter-center. This is dependent on the object being scanned. Let $\{\tilde{T}_{L,o}, \tilde{T}_{H,o}\}$ represent the corresponding measurements from an air-calibration scan. First, we generate the standard dual-energy projections, $p_{L,o}$ and $p_{H,o}$, defined by equation (1.10). Next, we apply an empirical 2nd order polynomial correction method that gives us $d_{j,o}$ as shown in equation (1.11).

$$p_{L,o} = \log \left[\frac{\tilde{T}_{L,o}}{T_{L,o}} \right] \quad \text{and} \quad p_{H,o} = \log \left[\frac{\tilde{T}_{H,o}}{T_{H,o}} \right] \quad (1.10)$$

$$d_{j,o} = \gamma_{j,1} p_{L,o} + \gamma_{j,2} p_{H,o} + \gamma_{j,3} p_{L,o} p_{H,o} + \gamma_{j,4} p_{L,o}^2 + \gamma_{j,5} p_{H,o}^2, \quad j \in \{1, 2\} \quad (1.11)$$

The polynomial coefficients $\gamma_{j,*}$ are obtained empirically through experiments with real objects, which are made from various compositions of the two basis materials.

1.2.3 Point Spread Function

The practical limitations of source and detector collimation result in scatter centers that are each spatially spread out, as opposed to narrow points in space. As shown in Figure 1.3, the finite aperture of the detector collimation results in the elongation of the scatter-center along the path of the incident pencil beam. Furthermore, the aperture of the source collimation, or equivalently the cross-section of the pencil beam, causes the scatter-center to also spread out along the transverse direction of the pencil beam. The overall spatial distribution of the scatter-center on the imaging plane, illustrated by the zoomed-in section of Figure 1.3, is modeled as a *point-spread function* (PSF). In this figure, we denote the PSF for the o -th scatter center as $h_o(x, z)$, where (x, z) are coordinates of any given point on the imaging plane. The PSF is only dependent on the scanner geometry, and is modeled through ray-trace simulations of the XDi system.

We render a sparse matrix H , whose entries $H_{o,r}$ represent the contribution of the r -th voxel to the spatial region encompassed by the o -th scatter center. $H_{o,r}$ is computed by finely sampling the point-spread function of the o -th scatter center, $h_o(x, z)$, at multiple points within the voxel r , and then taking the average of these PSF values. Finally, H is normalized such that

$$\sum_{r=1}^{N_r} H_{o,r} = 1,$$

where N_r is the total number of voxels.

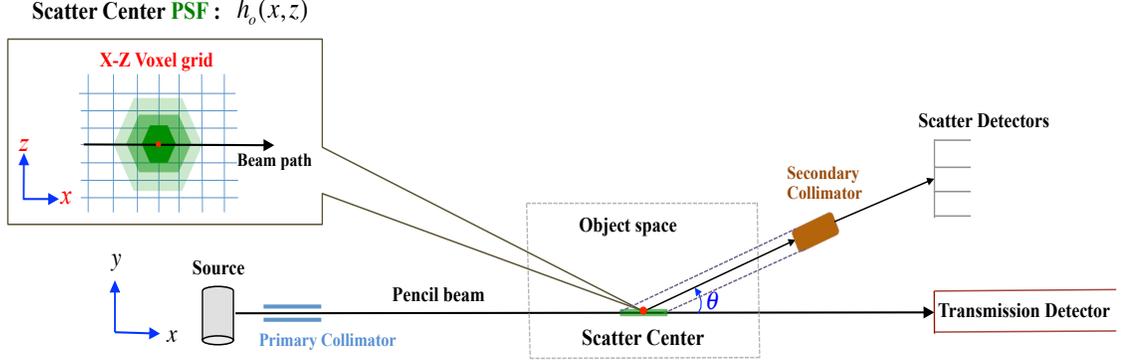


Fig. 1.3. Schematic depicting the spatial spread of a scatter-center due to the finite aperture of the detector collimation. The zoomed in section on the left illustrates the PSF of the scatter-center on the imaging plane, where the different “rings” or contours correspond to different PSF values.

1.2.4 Discretized Forward Model

Let $X \in \mathbb{R}^{N_r \times N_M}$ represent the unknown matrix to be reconstructed, where its entries $X_{r,m}$ denote the diffraction profile of the r -th voxel at momentum-transfer bin m . The scatter photon count received from the o -th scatter-center and m -th momentum-bin, $\lambda_{o,m}$, is a Poisson random variable. Along the same lines as equation (1.4), we can express the mean of $\lambda_{o,m}$ as follows

$$E[\lambda_{o,m}|X] = \sum_{m'=1}^{N_M} P_{o,m,m'} \beta_o I_{o,m'} \sum_{r=1}^{N_r} H_{o,r} X_{r,m'} \quad (1.12)$$

where β_o is a geometry-dependent calibration factor that is specific to the scatter-center, and $P_o \in \mathbb{R}^{N_M \times N_M}$, is a matrix that accounts for the non-ideal response of the detector to various photon energies. Its entries, $P_{o,m,m'}$, represent the probability that a photon’s detected momentum transfer falls within bin m , given that its true momentum transfer pertains to bin m' . P_o is empirically calibrated for each detector of the XDi system and is independent of the object being scanned. It is typically a lower triangular matrix, because photons impinging on a detector are most likely detected at energies roughly equal to or lower than its true energy.

1.2.4.1 Reconstruction of XRD spectrum on reduced subspace

Reconstructing the diffraction profile $X_{r,*}$, at every voxel r , over all N_M momentum-bins is too computationally expensive. A particular way to reduce this computational cost is to reduce the dimension of X in momentum-space. We assert that each row of X can be adequately represented by a reduced set of spectral features. The technique we apply to solve this dimensional reduction problem, coupled with a positivity constraint on X , is non-negative matrix factorization (NNMF).

We model X as a product of two matrices, \tilde{X} and T , both of which have nonnegative elements. \tilde{X} is a tall-and-thin matrix of size $N_r \times N_p$ and T is a short-and-wide matrix of size $N_p \times N_M$, where $N_p \ll N_M$.

$$X = \tilde{X}T . \quad (1.13)$$

Each row of T represents a basis spectrum used to represent the momentum-spectra of X . The plots in Figure 1.4 are the rows of T when $N_p = 4$. So now, for every voxel r , instead of reconstructing the entire diffraction profile $\{X_{r,m}\}_{m=1}^{N_M}$, we only need to recover the N_p coefficients of the basis spectra, $\{\tilde{X}_{r,p}\}_{p=1}^{N_p}$

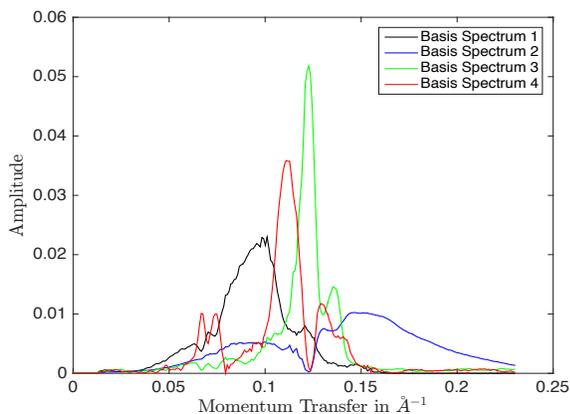


Fig. 1.4. Basis spectra obtained through NNMF decomposition of the training data, when number of basis components, N_p is 4

Currently we use a training data set comprising the diffraction profiles of various materials of interest, acquired through an offline measurement device. Then we quickly render a T matrix by NNMF decomposition [9] of the training data, and normalize T such that each row sums to 1.

$$\sum_{m=1}^{N_M} T_{p,m} = 1 \quad \forall p$$

1.3 Statistical Model and Reconstruction Approach

Our framework for XDi reconstruction is based on standard *maximum a posteriori* (MAP) estimation. Let λ represent the matrix of photon-counts detected from each scatter-center and momentum bin. The MAP estimate of X is formulated as

$$\begin{aligned} \hat{X} &= \tilde{X}T, \quad \text{where} \\ \tilde{X} &\leftarrow \underset{\tilde{X} \geq 0}{\operatorname{argmin}} \left\{ -\log p(\lambda|\tilde{X}) - \log p(\tilde{X}) \right\} \end{aligned} \quad (1.14)$$

In the following sections we shall first discuss the likelihood and prior models, $p(\lambda|\tilde{X})$ and $p(\tilde{X})$ respectively, and then move on to describing an iterative optimization strategy to solve equation (1.14).

1.3.1 Likelihood function

The measured photon count from scatter center o and with a momentum-transfer bin m is modeled as a poisson distribution shown below

$$p(\lambda_{o,m}|\tilde{X}) = e^{-\bar{\lambda}_{o,m}} \frac{(\bar{\lambda}_{o,m})^{\lambda_{o,m}}}{\lambda_{o,m}!}, \quad (1.15)$$

where $\bar{\lambda}_{o,m}$ is the mean photon count given by

$$\bar{\lambda}_{o,m} = \sum_{m'=1}^{N_M} P_{o,m,m'} \beta_o I_{o,m'} \sum_{r=1}^{N_r} \sum_{p=1}^{N_p} H_{o,r} \tilde{X}_{r,p} T_{p,m'}. \quad (1.16)$$

We accumulate the pre-computed coefficients that characterize the detector-response, attenuation correction, and the NNMF spectral bases in the above equation into a single parameter η as expressed below

$$\eta_{p,o,m} \triangleq \beta_o \sum_{m'=1}^{N_M} P_{o,m,m'} I_{o,m'} T_{p,m'} \quad (1.17)$$

Now, we can conveniently re-write the mean photon count as

$$\bar{\lambda}_{o,m} = \sum_{r,p} H_{o,r} \tilde{X}_{r,p} \eta_{p,o,m} . \quad (1.18)$$

The likelihood term assuming independence between measurements is then given by,

$$\begin{aligned} -\log p(\lambda|\tilde{X}) &= \sum_{o,m} -\log p(\lambda_{o,m}|\tilde{X}) \\ &= \sum_{o,m} \left\{ \sum_{r,p} H_{o,r} \tilde{X}_{r,p} \eta_{p,o,m} - \lambda_{o,m} \log \left(\sum_{r,p} H_{o,r} \tilde{X}_{r,p} \eta_{p,o,m} \right) \right\} + c(\lambda) . \end{aligned} \quad (1.19)$$

The term $c(\lambda)$ groups all additive terms that are not a function of X , which can therefore be dropped from the cost function.

1.3.2 Prior model

The prior distribution of \tilde{X} is modeled as a Markov random field (MRF) with a symmetric 26-point neighborhood system in 3-D space. The prior model of \tilde{X} incorporates a pair-wise Gibbs distribution with its general form given by

$$p(\tilde{X}) = \frac{1}{z} \prod_{p=1}^{N_p} \exp \left\{ -\frac{1}{\sigma} \sum_{(r,s) \in \mathcal{C}} b_{r,s} \rho \left(\tilde{X}_{r,p} - \tilde{X}_{s,p} \right) \right\} \quad (1.20)$$

where \mathcal{C} denotes the set of all pair-wise cliques in \tilde{X} , and $\rho(\cdot)$ is the positive and symmetric *potential function*. The parameter σ determines the overall level of smoothing, and z is a normalizing constant. The specific prior chosen for our application was a

Q-generalized Gaussian Markov random field, or Q-GGMRF [10] prior, where shape of the potential function is controlled by the parameters q and c as expressed below

$$\rho(\Delta) = |\Delta|^p \frac{|\Delta/c|^{q-p}}{1 + |\Delta/c|^{q-p}} \quad (1.21)$$

The penalty levied on differences between neighboring pixel varies as $\rho(\Delta) \approx c^{p-q}|\Delta|^q$ for very small differences where $|\Delta| \ll c$, and $\rho(\Delta) \approx |\Delta|^p$ in the range where $|\Delta| \gg c$. This approximate piece-wise behavior of the potential function permits us to tune our prior through a suitable choice of parameters p and q to preserve both low-contrast details as well as edge characteristics. Common choice of parameters is $p = 2$, and q in the interval $[1, 2]$. Note that these values result in a convex potential function, which ultimately ensures an overall convex cost function. Selecting $q = 1$ further simplifies the prior and its derivative, and thus expedites computation.

1.3.3 Iterative coordinate descent

To solve the multi-variable optimization problem posed by equation (1.14), we use the iterative coordinate descent (ICD) strategy which minimizes the cost function $g(\tilde{X})$ through a series of successive 1D minimizations [5,11]. Substituting expressions for the log likelihood and log prior in equation (1.14), the global cost function is

$$g(\tilde{X}) = \sum_{o,m} \left\{ \sum_{r,p} H_{o,r} \tilde{X}_{r,p} \eta_{p,o,m} - \lambda_{o,m} \log \left(\sum_{r,p} H_{o,r} \tilde{X}_{r,p} \eta_{p,o,m} \right) \right\} + \frac{1}{\sigma} \sum_p \sum_{(r,s) \in \mathcal{C}} b_{r,s} \rho(\tilde{X}_{r,p} - \tilde{X}_{s,p}) . \quad (1.22)$$

Note that since the above cost function is convex, our optimization algorithm ultimately converges to the global minimum. Here each ICD update minimizes the cost function with respect to a single optimization variable keeping all the other $N_r N_p - 1$ variables fixed. The update for the voxel s and spectral feature q is given by

$$\tilde{X}_{s,q} \leftarrow \underset{u \geq 0}{\operatorname{argmin}} g_l(u) \quad , \text{ where} \quad (1.23)$$

$$\begin{aligned}
g_l(u) = & \sum_{o,m} \left\{ u H_{o,s} \eta_{q,o,m} - \lambda_{o,m} \log \left[\left(u - \tilde{X}_{s,q} \right) H_{o,s} \eta_{q,o,m} + \sum_{r,p} H_{o,r} \tilde{X}_{r,p} \eta_{p,o,m} \right] \right\} \\
& + \frac{1}{\sigma} \sum_{r \in \partial s} b_{s,r} \rho \left(u - \tilde{X}_{r,q} \right)
\end{aligned} \tag{1.24}$$

and ∂s denotes the neighborhood of voxel s and $g_l(u)$ is the localized cost function. The solution to equation (1.23) does not have a closed form expression. However, because $g_l(\cdot)$ is convex, the solution can be obtained by finding the root of the local cost function's derivative $g'_l(u)$. By exploiting the sparse nature of both the PSF matrix H and measurements λ , the computation of $g'_l(u)$ can be accelerated. Algorithm 1 provides the pseudo-code for the above ICD optimization approach.

1.4 Results and Discussion

To validate our forward model and ICD optimization algorithm, we perform reconstructions on XRD measurements acquired from actual baggage. Further we investigate the effect of various parameters such as the voxel resolution, level of regularization contributed by the prior, and the dimensionality of the reduced spectral subspace, N_p , on the quality of reconstruction.

A key benefit of introducing the point spread function to the reconstruction is that it allows flexibility in the definition of the voxel grid in object space. More, specifically we can reconstruct at any spatial resolution we choose. However, beyond some spatial sampling rate, we will no longer recover any additional spatial resolution.

Fig. 1.5 depicts various reconstructions of a test bag containing two bottles filled to capacity with two different liquids. The conventional X-ray transmission image of the bag taken prior to the scan is shown in (a). The two liquids in this bag have similar densities. Their scatter-strengths, which is the sum of the diffraction spectra over all momentum-bins, differ by less than 15%.

Back-projection of the photon-count measurements onto the voxel grid using a high resolution PSF matrix H is shown in Fig. 1.5(b). The reconstructed images Fig. 1.5(c) to (f) represent the scatter-strength, which in this case is the sum of \tilde{X} over all

Algorithm 1 ICD Algorithm for MAP estimation

1: # **Notation:** o - scatter-center index, m - momentum-bin index, p - NNMF basis index, r - voxel index

2: **for** every (p, o) **do**

3: **for** every m **do**

4: $\eta_{p,o,m} = \beta_o \sum_{m'} P_{o,m,m'} I_{o,m'} T_{p,m'}$

5: **end for**

6: $\tilde{n}_{p,o} = \sum_m \eta_{p,o,m}$

7: **end for**

8: **for** every o **do**

9: $\mathcal{M}_o = \{m \mid \lambda_{o,m} > 0\}$

10: **for** every $m \in \mathcal{M}_o$ **do**

11: $S_{o,m} = \sum_{r,p} H_{o,r} \tilde{X}_{r,p} \eta_{p,o,m}$ ▷ state vector

12: **end for**

13: **end for**

14: **while** not converged **do**

15: **for** every voxel s and NNMF component q **do**

16: $v \leftarrow \tilde{X}_{s,q}$

17: $\mathcal{O} \leftarrow \{o \mid H_{o,s} \neq 0\}$

18: # Root-search on derivative of cost function specified by (1.24) :

19: $\tilde{X}_{s,q} \leftarrow \{u \mid g'_l(u) = 0\}$ where

20:
$$g'_l(u) = \sum_{o \in \mathcal{O}} \left\{ H_{o,s} \tilde{\eta}_{q,o} - \sum_{m \in \mathcal{M}_o} \frac{\lambda_{o,m} H_{o,s} \eta_{q,o,m}}{(u - \tilde{X}_{s,q}) H_{o,s} \eta_{q,o,m} + S_{o,m}} \right\} + \sum_{r \in \partial s} \frac{b_{s,r}}{\sigma} \rho' \left(u - \tilde{X}_{r,q} \right).$$

21: **for** every (o, m) where $H_{o,s} \neq 0$ **do**

22: $S_{o,m} \leftarrow S_{o,m} + (\tilde{X}_{s,q} - v) H_{o,s} \eta_{q,o,m}$

23: **end for**

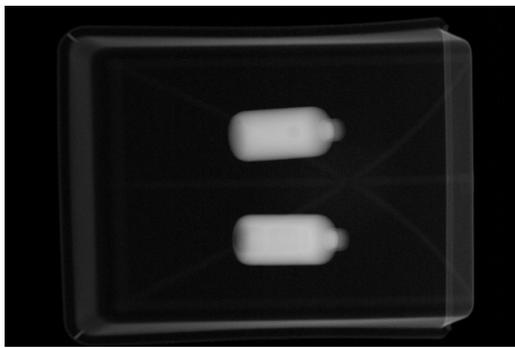
24: **end for**

25: **end while**

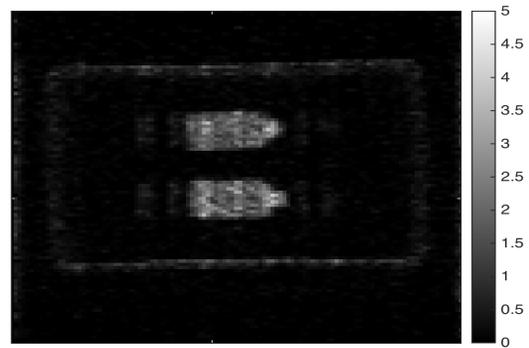
N_p basis components, because each basis spectrum sums to 1. First, we investigate if we can extract finer spatial information from the XDi bag scans by decreasing the voxel size. Results Fig. 1.5(c) and (d) pertain to reconstructions performed at the standard voxel size of the scanner. We then tried doubling resolution in each spatial dimension and re-defined the PSF matrix accordingly. We were able to achieve enhanced image quality as illustrated by the resulting high-resolution reconstructions in Fig. 1.5(d) and (e). Next, we examine the effect of the prior on image quality. A comparison of results Fig. 1.5(d) vs (f), and Fig. 1.5(c) vs (e) demonstrates that a higher level of regularization results in greater smoothness and removal of high-contrast artifacts.

The results in Fig. 1.6 are based on the NNMF basis spectra shown in Fig. 1.4. As opposed to the overall scatter strength, Fig. 1.6(a) and (b) separately show the reconstructed coefficients for the 2nd and 3rd NNMF basis spectra respectively. Further, Fig. 1.6(c) and (d) shows the reconstructed momentum spectrum of the two liquids in the scanned bag. The spectra are averaged over the volume of each bottle and displayed for reconstructions with different values of N_p . In general, most liquids have relatively smooth diffraction profiles, rather than a rugged profile with several sharp peaks that solid crystals and powders usually exhibit. Typically, accounting for a higher number of basis spectra should lead to major improvements in the reconstructed spectral profile and cause it to converge towards the true spectrum. In our current example, the spectral reconstructions do not vastly vary between using four and six basis components, because the predominant spectral features in the actual data resemble the 2nd basis spectrum.

The reconstruction was performed over 97 bags containing liquids A and B, similar to that shown in 1.5(a). The XRD measurements for these bags were acquired from various scanners and scan-times. We designed a linear classification algorithm that identifies the liquids from their reconstructions. Fifteen bags were reserved for training our classification algorithm, while the other 82 bags were used for testing. More specifically, from the former set of bags, we used every voxel that is internal



(a) Transmission prescan image



(b) Back-Projection with PSF, high resolution

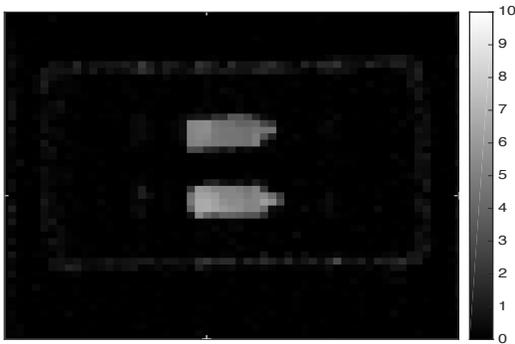
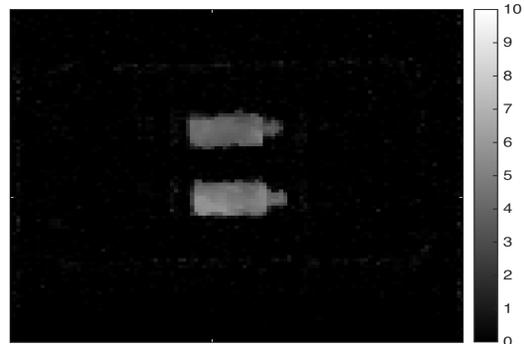
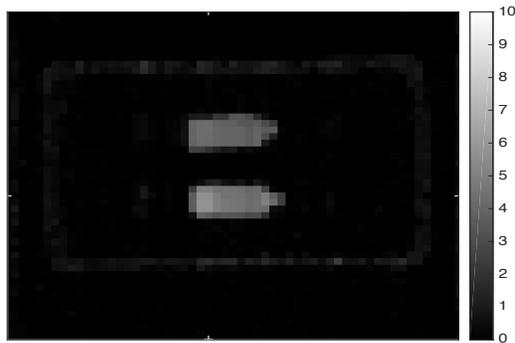
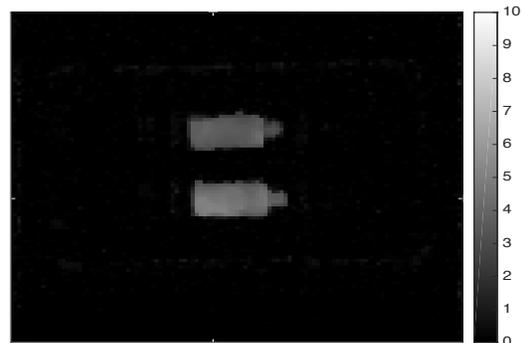
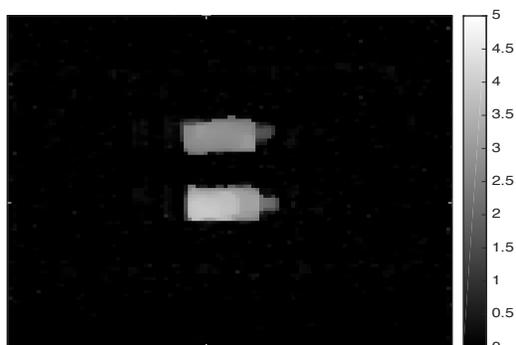
(c) Reconstructed, low resolution, $\sigma = 1.2$ (d) Reconstructed, high resolution, $\sigma = 1.2$ (e) Reconstructed, low resolution, $\sigma = 0.6$ (f) Reconstructed, high resolution, $\sigma = 0.6$

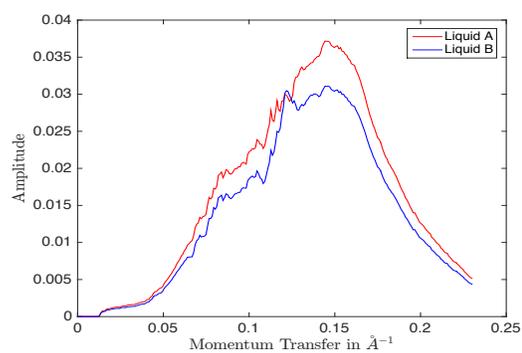
Fig. 1.5. Model-based scatter strength reconstructions of an actual bag, with its prescan image depicted in (a). All slices shown are coronal, i.e. looking down on the scanner tunnel. In (d) and (f), the voxel size is halved in all 3-dimensions. The number of basis spectra N_p was chosen to be 4.



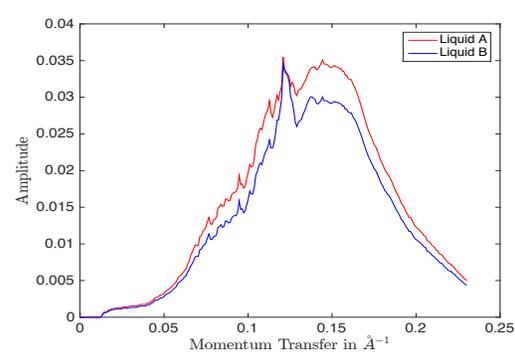
(a) Reconstructed coefficient for 2nd NNMF basis spectrum



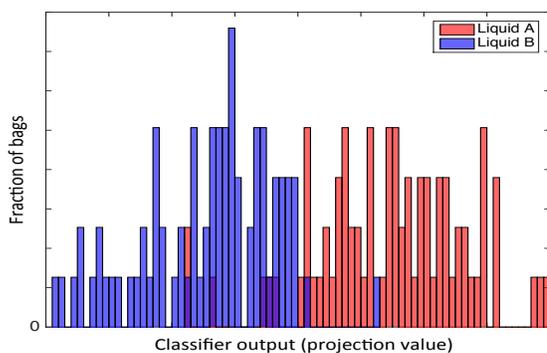
(b) Reconstructed coefficient for 3rd NNMF basis spectrum



(c) Reconstructed spectrum, $N_p = 4$



(d) Reconstructed spectrum, $N_p = 6$



(e) Classification of the 2 liquids from multiple XDi scans (x-y axis values removed deliberately)

Fig. 1.6. (a) -(d): Reconstructed diffraction profiles of two liquids A and B, contained within the bottles of the scanned bag. (e): Histogram for classification of the two liquids based on XDi scans of 82 different bags; x-axis represents a linear projection of the 4 spectral coefficients and y-axis represents the fraction of scanned bags.

to either liquid bottle as a training sample for our classifier. Subsequently, from the training data, we determined an optimal linear discriminant based on all the four NNMF basis coefficients that separates the two liquids. From Fig. 1.6(e), we observe that there is an optimal threshold for the linear-projection output of the classifier that clearly distinguishes between the two liquids. We deliberately do not include details pertaining to the optimal threshold value and corresponding detection and false-alarm rate results of our classification method.

1.5 Conclusion

In this chapter of the thesis, we presented a 4-D model-based iterative reconstruction (MBIR) method for a low-angle detection XDi scanner. We proposed a novel forward model that includes a system-dependent spatial point-spread function, corrects for the energy-dependent X-ray attenuation, and also captures the non-ideal energy-response of the detectors. To reduce computational cost, we represented the unknown spectra with a reduced spectral bases. Reconstructions with real bag data show that it is possible to achieve sufficient spatial resolution despite very low photon counts. Further, we demonstrated the ability of our approach to distinguish between materials of similar densities based on the reconstructed basis spectral coefficients.

2. DISTRIBUTED ITERATIVE CT RECONSTRUCTION USING MULTI-AGENT CONSENSUS EQUILIBRIUM

2.1 Introduction

Tomographic reconstruction algorithms can be roughly divided into two categories: analytical reconstruction methods [12, 13] and regularized iterative reconstruction methods [14] such as model-based iterative reconstruction (MBIR) [15–18]. MBIR methods have the advantage that they can improve reconstructed image quality particularly when projection data are sparse and/or the X-ray dosage is low. This is because MBIR integrates a model of both the sensor and object being imaged into the reconstruction process [15, 17–19]. However, the high computational cost of MBIR often makes it less suitable for solving large reconstruction problems in real-time.

One approach to speeding MBIR is to precompute and store the system matrix [20–22]. In fact, the system matrix can typically be precomputed in applications such as scientific imaging, non-destructive evaluation (NDE), and security scanning where the system geometry does not vary from scan to scan. However, for large tomographic problems, the system matrix may become too large to store on a single compute node. Therefore, there is a need for iterative reconstruction algorithms that can distribute the system matrix across many nodes in a large cluster.

More recently, advanced prior methods have been introduced into MBIR which can substantially improve reconstruction quality by incorporating machine learning approaches. For example, Plug-n-play (PnP) priors [18, 19], consensus equilibrium (CE) [23], and RED [24] allow convolutional neural networks (CNN) to be used in prior modeling. Therefore, methods that distribute tomographic reconstruction across large compute clusters should also be designed to support these emerging approaches.

In order to make MBIR methods useful, it is critical to parallelize the algorithms for fast execution. Broadly speaking, parallel algorithms for MBIR fall into two categories: fine-grain parallelization methods that are most suitable for shared-memory (SM) implementation [20,21,25,26], and course-grain parallelization methods that are most suitable for distributed-memory (DM) implementation [27–29]. So for example, SM methods are best suited for implementation on a single multi-core CPU processor or a GPU, while DM methods are better suited for computation across a large cluster of compute nodes. Further, DM methods ensure that the overhead incurred due to inter-node communication and synchronization does not dominate the computation.

In particular, some DM parallel methods can handle large-scale tomographic problems by distributing the system-matrix across multiple nodes, while others do not. For example, the DM algorithm of Wang et al. [27] parallelizes the reconstruction across multiple nodes, but it requires that each node have a complete local copy of the system matrix. Alternatively, the DM algorithms of Linyuan et al. [28] and Cui et al. [29] could potentially be used to parallelize reconstruction across a cluster while distributing the system matrix across the nodes. However, the method of Linyuan [28] is restricted to the use of a total variation (TV) prior [30, 31] and in experiments has required 100s of iterations for convergence, which is not practical for large problems. Alternatively, the the method of Cui [29] is for use in unregularized PET reconstruction.

In this thesis chapter, we present a Multi-agent Consensus Equilibrium (MACE) reconstruction algorithm that distributes both the computation and memory of iterative CT reconstruction across a large number of parallel nodes [32–34]. The MACE approach uses the consensus equilibrium [23] framework to break the reconstruction problem into a set of subproblems which can be solved separately and then integrated together to achieve a high-quality reconstruction. By distributing computation over a set of compute nodes, MACE enables the solution of reconstruction problems that would otherwise be too large to solve.

Figure 2.1 illustrates our two approaches to this distributed CT reconstruction problem. While both the approaches integrate multiple sparse-view reconstructions across a compute cluster into a high-quality reconstruction, they differ based on how the prior model is implemented. Figure 2.1(a) depicts our basic MACE approach that utilizes conventional edge-preserving regularization [15, 35] as a prior model and converges to the maximum-a-posteriori (MAP) estimate. Figure 2.1(b) shows our second approach called MACE-PnP which allows for distributed CT reconstruction using plug-and-play (PnP) priors [18, 19]. These PnP priors substantially improve reconstructed image quality by implementing the prior model using a denoising algorithm based on methods such as BM3D [36] or deep residual CNNs [37]. We prove that MACE-PnP provides a parallel algorithm for computing the standard serial PnP reconstruction of [18].

A direct implementation of MACE is not practical because it requires repeated application of proximal operators that are themselves iterative. In order to overcome this problem, we introduce the concept of *partial updates*, a general approach for replacing any proximal operator with a non-iterative update. We also prove the convergence of this method for our application.

Our experiments are divided into two parts and are based on real CT datasets from synchrotron imaging and security scanning. In the first part, we use the MACE algorithm to parallelize 2D CT reconstructions across a distributed CPU cluster of 16 compute nodes. We show that MACE both speeds up reconstruction while drastically reducing the memory footprint of the system matrix on each node. We incorporate regularization in the form of either conventional priors such as Q-GGMRF, or alternatively, advanced denoisers such as BM3D that improve reconstruction quality. In the former case, we verify that our approach converges to the Bayesian estimate [15, 17], while in the latter case, we verify convergence to the PnP solution of [18, 23].

In the second part of our experiments, we demonstrate an implementation of MACE on a large-scale supercomputer that can reconstruct a large 3D CT dataset. For this problem, the MACE algorithm is used in conjunction with the super-voxel

ICD (SV-ICD) algorithm [20] to distribute computation over 1200 compute nodes, consisting of a total of 81,600 cores. Importantly, in this case the MACE algorithm not only speeds reconstruction, it enables reconstruction for a problem that would otherwise be impossible since the full system matrix is too large to store on a single node.

2.2 Distributed CT Reconstruction using Conventional Priors

2.2.1 CT Reconstruction using MAP Estimation

We formulate the CT reconstruction problem using the maximum-a-posteriori (MAP) estimate given by [38]

$$x^* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} f(x; \beta) , \quad (2.1)$$

where f is the MAP cost function defined by

$$f(x; \beta) = -\log p(y|x) - \log p(x) + \text{const} ,$$

y is the preprocessed projection data, $x \in \mathbb{R}^n$ is the unknown image of attenuation coefficients to be reconstructed, and const represents any possible additive constant.

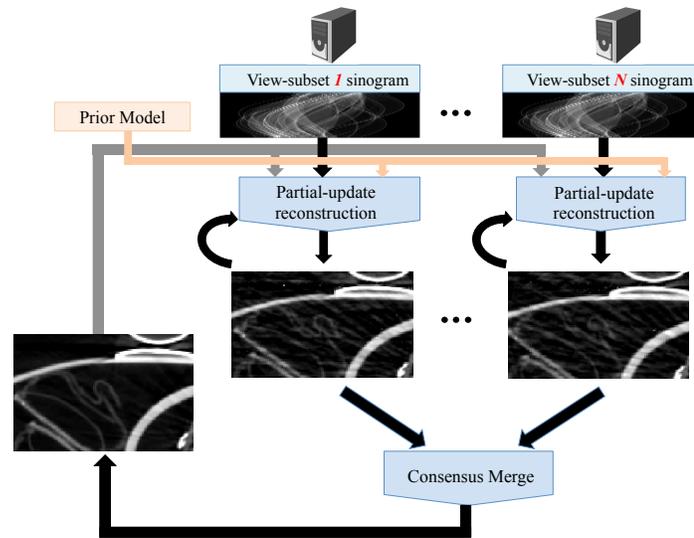
For our specific problem, we choose the forward model $p(y|x)$ and the prior model $p_\beta(x)$ so that

$$f(x; \beta) = \sum_{k=1}^{N_\theta} \frac{1}{2} \|y_k - A_k x\|_{\Lambda_k}^2 + \beta h(x) , \quad (2.2)$$

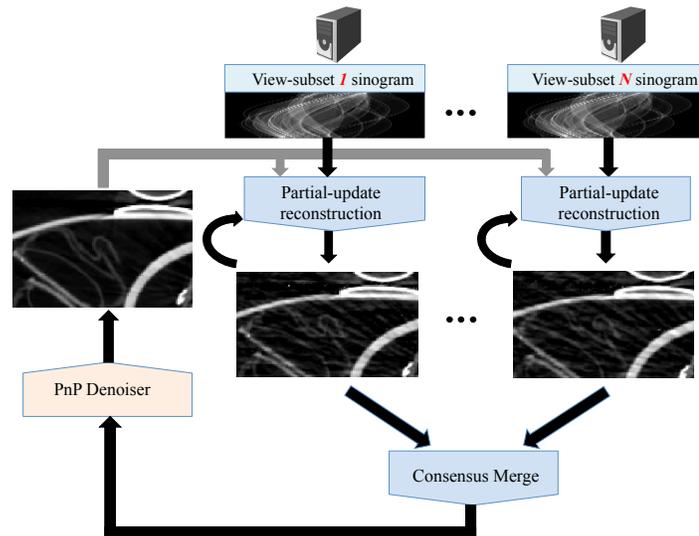
where $y_k \in \mathbb{R}^{N_D}$ denotes the k^{th} view of data, N_θ denotes the number of views, $A_k \in \mathbb{R}^{N_D \times n}$ is the system matrix that represents the forward projection operator for the k^{th} view, and $\Lambda_k \in \mathbb{R}^{N_D \times N_D}$ is a diagonal weight matrix corresponding to the inverse noise variance of each measurement. Also, the last term

$$\beta h(x) = -\log p(x) + \text{const} ,$$

represents the prior model we use where β can be used to control the relative amount of regularization. In this section, we will generally assume that $h(\cdot)$ is convex, so then



(a) MACE



(b) MACE-PnP

Fig. 2.1. Illustration of the MACE algorithm for distributing CT reconstruction across a parallel cluster of compute nodes. (a) An illustration of the MACE algorithm using a conventional prior model. The MACE algorithm works by splitting the data into view subsets and reconstructing them in parallel. The individual reconstructions are then merged in an iterative loop that results in the true MAP reconstruction for the full data set. (b) An illustration of the MACE-PnP algorithm which extends the MACE framework to the use of Plug-n-Play prior models to improve reconstruction quality. In this case, a denoiser is run in the MACE loop in order to implement an advanced prior model. The MACE and MACE-PnP algorithms distribute both computation and memory across parallel clusters of computers, thereby enabling the reconstruction of large tomographic data sets.

f will also be convex. A typical choice of h which we will use in the experimental section is the Q-Generalized Gaussian Markov Random Field (Q-GGMRF) prior [35] that preserves both low contrast characteristics as well as edges.

In order to parallelize our problem, we will break up the MAP cost function into a sum of auxiliary functions, with the goal of minimizing these individual cost functions separately. So we will represent the MAP cost function as

$$f(x; \beta) = \sum_{i=1}^N f_i(x; \beta) ,$$

where

$$f_i(x; \beta) = \sum_{k \in J_i} \frac{1}{2} \|y_k - A_k x\|_{\Lambda_k}^2 + \frac{\beta}{N} h(x) , \quad (2.3)$$

and the view subsets J_1, \dots, J_N partition the set of all views into N subsets.¹ In this thesis, we will generally choose the subsets J_i to index interleaved view subsets, but the theory we develop works for any partitioning of the views. In the case of interleaved view subsets, the view subsets are defined by

$$J_i = \{ m : m \bmod N = i, m \in \{1, \dots, N_\theta\} \} .$$

2.2.2 MACE framework

In this section, we introduce a framework, which we refer to as multi-agent consensus equilibrium (MACE) [23], for solving our reconstruction problem through the individual minimization of the terms $f_i(x)$ defined in (2.3).² Importantly, minimization of each function, $f_i(\cdot)$, has exactly the form of the MAP CT reconstruction problem but with the sparse set of views indexed by J_i . Therefore, MACE integrates the results of the individual sparse reconstruction operators, or agents, to produce a consistent solution to the full problem.

¹By partition, we mean that $\cup_{i=1}^N J_i = \{1, 2, \dots, N_\theta\}$ and $\cap_{i=1}^N J_i = \emptyset$.

²In this section, we suppress the dependence on β for notational simplicity.

To do this, we first define the agent, or in this case the proximal map, for the i^{th} auxiliary function as

$$F_i(x) = \operatorname{argmin}_{z \in \mathbb{R}^n} \left\{ f_i(z) + \frac{\|z - x\|^2}{2\sigma^2} \right\}, \quad (2.4)$$

where σ is a user selectable parameter that will ultimately effect convergence speed of the algorithm. Intuitively, the function $F_i(x)$ takes an input image x , and returns an image that reduces its associated cost function f_i and is close to x .

Our goal will then be to solve the following set of MACE equations

$$F_i(x^* + u_i^*) = x^* \text{ for } i = 1, \dots, N, \quad (2.5)$$

$$\sum_{i=1}^N u_i^* = 0, \quad (2.6)$$

where $x^* \in \mathbb{R}^n$ has the interpretation of being the consensus solution, and each $u_i^* \in \mathbb{R}^n$ represents the force applied by each agent that balances to zero.

Importantly, the solution x^* to the MACE equations is also the solution to the MAP reconstruction problem of equation (2.1) (see Theorem 1 of [23]). In order to see this, notice that since f_i is convex we have that

$$\partial f_i(x^*) + \frac{x^* - (x^* + u_i^*)}{\sigma^2} \ni 0, \quad i = 1, \dots, N,$$

where ∂f_i is the sub-gradient of f_i . So by summing over i and applying (2.6) we have that

$$\partial f(x^*) \ni 0.$$

Which shows that x^* is a global minimum to the convex MAP cost function. We can prove the converse in a similar manner.

We can represent the MACE equilibrium conditions of (2.5) and (2.6) in a more compact notational form. In order to do this, first define the stacked vector

$$\mathbf{v} = \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix}, \quad (2.7)$$

where each component $v_i \in \mathbb{R}^n$ of the stack is an image. Then we can define a corresponding operator \mathbf{F} that stacks the set of agents as

$$\mathbf{F}(\mathbf{v}) = \begin{pmatrix} F_1(v_1) \\ \vdots \\ F_N(v_N) \end{pmatrix}, \quad (2.8)$$

where each agent F_i operates on the i -th component of the stacked vector. Finally, we define a new operator $\mathbf{G}(\mathbf{v})$ that computes the average of each component of the stack, and then redistributes the result. More specifically,

$$\mathbf{G}(\mathbf{v}) = \begin{pmatrix} \bar{\mathbf{v}} \\ \vdots \\ \bar{\mathbf{v}} \end{pmatrix}, \quad (2.9)$$

where $\bar{\mathbf{v}} = \frac{1}{N} \sum_{i=1}^N v_i$.

Using this notation, the MACE equations of (2.5) and (2.6) have the much more compact form of

$$\mathbf{F}(\mathbf{v}^*) = \mathbf{G}(\mathbf{v}^*), \quad (2.10)$$

with the solution to the MAP reconstruction is given by $x^* = \bar{\mathbf{v}}^*$ where $\bar{\mathbf{v}}^*$ is the average of the stacked components in \mathbf{v}^* .

The MACE equations of (2.10) can be solved in many ways, but one convenient way to solve them is to convert these equations to a fixed-point problem, and then use well known methods for efficiently solving the resulting fixed-point problem [23]. It can be easily shown (see appendix A) that the solution to the MACE equations are exactly the fixed points of the operator $\mathbf{T} = (2\mathbf{F} - \mathbf{I})(2\mathbf{G} - \mathbf{I})$ given by

$$\mathbf{T}\mathbf{w}^* = \mathbf{w}^*, \quad (2.11)$$

where then $\mathbf{v}^* = (2\mathbf{G} - \mathbf{I})\mathbf{w}^*$.

We can evaluate the fixed-point \mathbf{w}^* using the *Mann* iteration [39]. In this approach, we start with any initial guess and apply the following iteration,

$$\mathbf{w}^{(k+1)} = \rho\mathbf{T}\mathbf{w}^{(k)} + (1 - \rho)\mathbf{w}^{(k)}, k \geq 0, \quad (2.12)$$

which can be shown to converge to \mathbf{w}^* for $\rho \in (0, 1)$. The Mann iteration of (2.12) has guaranteed convergence to a fixed-point \mathbf{v}^* if \mathbf{T} is non-expansive. Both F_i , as well as its reflection, $2F_i - I$, are non-expansive, since each proximal map F_i belongs to a special class of operators called *resolvents* [39]. Also, we can easily show that $2\mathbf{G} - \mathbf{I}$ is non-expansive. Consequently, \mathbf{T} is also non-expansive and (2.12) is guaranteed to converge.

In practice, \mathbf{F} is a parallel operator that evaluates N agents that can be distributed over N nodes in a cluster. Alternatively, the \mathbf{G} operator has the interpretation of a reduction operation across the cluster followed by a broadcast of the average across the cluster nodes.

2.2.3 Partial Update MACE Framework

A direct implementation of the MACE approach specified by (2.12) is not practical, since it requires a repeated application of proximal operators $F_i, i = 1, \dots, N$, that are themselves iterative. Consequently, this direct use of (2.12) involves many nested loops of intense iterative optimization, resulting in an impractically slow algorithm.

In order to overcome the above limitation, we propose a *partial-update MACE* algorithm that permits a faster implementation without affecting convergence. In partial-update MACE, we replace each proximal operator with a fast non-iterative update in which we partially evaluate the proximal map, F_i , using only a single pass of iterative optimization.

Importantly, a partial computation of $F_i(\cdot; \sigma)$ resulting from a single pass of iterative optimization will be dependent on the initial state. So, we use the notation $\tilde{F}_i(\cdot; \sigma, X_i)$ to represent the partial-update for $F_i(\cdot; \sigma)$, where $X_i \in \mathbb{R}^n$ specifies the initial state. Analogous to (2.8), $\tilde{\mathbf{F}}(\mathbf{v}; \sigma, \mathbf{X})$ then denotes the partial update for stacked operator $\mathbf{F}(\mathbf{v}; \sigma)$ from an initial state $\mathbf{X} \in \mathbb{R}^{nN}$.

Algorithm 2 provides the pseudo-code for the Partial-update MACE approach. Note that this algorithm strongly resembles equation (2.12) that evaluates the fixed-point of map $(2\mathbf{F} - \mathbf{I})(2\mathbf{G} - \mathbf{I})$, except that \mathbf{F} is replaced with its partial update as shown in line 7 of the pseudo-code. Note that this framework allows a single pass of any optimization technique to compute the partial update. In this thesis, we will use a single pass of the the Iterative Coordinate Descent (ICD) optimization method [17,20], that greedily updates each voxel, but single iterations of other optimization methods can also be used.

Algorithm 2 Partial-update MACE with conventional priors

- 1: *Initialize:*
 - 2: $\mathbf{w}^{(0)} \leftarrow$ any value $\in \mathbb{R}^{nN}$
 - 3: $\mathbf{X}^{(0)} = \mathbf{G}(\mathbf{w}^{(0)})$
 - 4: $k \leftarrow 0$
 - 5: **while** not converged **do**
 - 6: $\mathbf{v}^{(k)} = (2\mathbf{G} - \mathbf{I}) \mathbf{w}^{(k)}$
 - 7: $\mathbf{X}^{(k+1)} = \tilde{\mathbf{F}}(\mathbf{v}^{(k)}; \sigma, \mathbf{X}^{(k)})$ ▷ Approximate $\mathbf{F}(\mathbf{v}^{(k)})$
 - 8: $\mathbf{w}^{(k+1)} = 2\mathbf{X}^{(k+1)} - \mathbf{v}$ ▷ $\approx (2\mathbf{F} - \mathbf{I})(2\mathbf{G} - \mathbf{I})\mathbf{w}^{(k)}$
 - 9: $\mathbf{w}^{(k+1)} \leftarrow \rho\mathbf{w}^{(k+1)} + (1 - \rho)\mathbf{w}^{(k)}$ ▷ Mann update
 - 10: $k \leftarrow k + 1$
 - 11: **end while**
 - 12: *Solution:*
 - 13: $x^* = \bar{\mathbf{w}}^{(k)}$ ▷ Consensus solution
-

In order to understand the convergence of the partial-update algorithm, we can formulate the following update with an augmented state as

$$\begin{aligned} \begin{bmatrix} \mathbf{w}^{(k+1)} \\ X^{(k+1)} \end{bmatrix} &= \begin{bmatrix} \rho & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2\tilde{\mathbf{F}}(\mathbf{v}^{(k)}; X^{(k)}) - \mathbf{v}^{(k)} \\ \tilde{\mathbf{F}}(\mathbf{v}^{(k)}; X^{(k)}) \end{bmatrix} \\ &+ \begin{bmatrix} 1 - \rho & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w}^{(k)} \\ X^{(k)} \end{bmatrix}, \end{aligned} \tag{2.13}$$

where $\mathbf{v}^{(k)} = (2\mathbf{G} - \mathbf{I})\mathbf{w}^{(k)}$. We specify \tilde{F} more precisely in Algorithm 3. In Theorem 2.2.1, we show that any fixed-point $(\mathbf{w}^*, \mathbf{X}^*)$ of this partial-update algorithm is a solution to the exact MACE method of (2.11). Further, in Theorem 2.2.2 we show that for the specific case where f_i defined in (2.3) is strictly quadratic, the partial-update algorithm has guaranteed convergence to a fixed-point.

Theorem 2.2.1 *Let $f_i : \mathbb{R}^n \rightarrow \mathbb{R}, i = 1, \dots, N$ be a strictly convex and differentiable function. Let F_i denote the proximal map of f_i . Let $\tilde{F}_i(v; x)$ denote the partial update for $F_i(v)$ as specified by Algorithm 3. Then any fixed-point $(\mathbf{w}^*, \mathbf{X}^*)$ of the Partial-update MACE approach represented by (2.13) is a solution to the exact MACE approach specified by (2.11).*

Proof Proof is in Appendix B. ■

Theorem 2.2.2 *Let B be a positive definite $n \times n$ matrix, and let $f_i : \mathbb{R}^n \rightarrow \mathbb{R}^n, i = 1, \dots, N$ each be given by*

$$f_i(x) = \sum_{k \in J_i} \frac{1}{2} \|y_k - A_k x\|_{\Lambda_k}^2 + \frac{\beta}{N} x^T B x.$$

Let F_i denote the proximal map of f_i . Let $\tilde{F}_i(v; x, \sigma), v, x \in \mathbb{R}^n$, denote the partial update for proximal operation $F_i(v; \sigma)$ as shown in Algorithm 3. Then equation (2.13) can be represented by a linear transform

$$\begin{bmatrix} \mathbf{w}^{(k+1)} \\ X^{(k+1)} \end{bmatrix} = \mathbf{M}_{\sigma^2} \begin{bmatrix} \mathbf{w}^{(k)} \\ X^{(k)} \end{bmatrix} + \mathbf{c}(\mathbf{y}, \mathbf{A}, \rho),$$

where $\mathbf{M}_{\sigma^2} \in \mathbb{R}^{2Nn \times 2Nn}$ and $\mathbf{c} \in \mathbb{R}^{2Nn}$. Also, for sufficiently small $\sigma > 0$, any eigenvalue of the matrix \mathbf{M}_{σ^2} is in the range $(0, 1)$ and hence the iterates defined by (2.13) converge in the limit $k \rightarrow \infty$ to $(\mathbf{w}^, \mathbf{X}^*)$, the solution to the exact MACE approach specified by (2.11).*

Proof Proof is in Appendix B. ■

Consequently, in the specific case of strictly convex and quadratic problems, the result of Theorem 2.2.2 shows that despite the partial-update approximation to the MACE approach, we converge to the exact consensus solution. In practice, we use non-Gaussian MRF prior models for their edge-preserving capabilities, and so the global reconstruction problem is convex but not quadratic. However, when the priors are differentiable, they are generically locally well-approximated by quadratics, and our experiments show that we still converge to the exact solution even in such cases.

Algorithm 3 ICD-based Partial-update for proximal operation $F_i(v)$ using an initial

state x

- 1: Define $\varepsilon_s = [0, \dots, 1, \dots, 0]^t \in \mathbb{R}^n$ ▷ entry 1 at s -th position
 - 2: $z = x$ ▷ Copy initial state
 - 3: **for** $s = 1$ to n **do**
 - 4: $\alpha_s = \operatorname{argmin}_\alpha \left\{ f_i(z + \alpha \varepsilon_s) + \frac{1}{2\sigma^2} \|z + \alpha \varepsilon_s - v\|^2 \right\}$
 - 5: $z \leftarrow z + \alpha_s \varepsilon_s$
 - 6: **end for**
 - 7: $\tilde{F}_i(v; x) = z$ ▷ Partial update
-

2.3 MACE with Plug-and-Play Priors

In this section, we generalize our approach to incorporate *Plug-n-Play* (PnP) priors implemented with advanced denoisers [18]. Since we will be incorporating the prior as a denoiser, for this section we drop the prior terms of in equation (2.2) by setting $\beta = 0$. So let $f(x) = f(x; \beta = 0)$ denote the CT log likelihood function of (2.2) with $\beta = 0$ and no prior term, and let $F(x)$ denote its corresponding proximal map. Then Buzzard et al. in [23] show that the PnP framework of [18] can be specified by the following equilibrium conditions

$$F(x^* - \alpha^*; \sigma) = x^* \tag{2.14}$$

$$H(x^* + \alpha^*) = x^*, \tag{2.15}$$

where $H : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is the plug-n-play denoiser used in place of a prior model. This framework supports a wide variety of denoisers including BM3D and residual CNNs that can be used to improve reconstruction quality as compared to conventional prior models [18, 19].

Let $f_i(x) = f_i(x; \beta = 0)$ to be the log likelihood terms from (2.3) corresponding to the sparse view subsets, and let $F_i(x)$ be their corresponding proximal maps. Then in Appendix C we show that the PnP result specified by (2.14) and (2.15) is equivalent to the following set of equilibrium conditions.

$$F_i(x^* + u_i^*; \sigma) = x^*, \quad i = 1, \dots, N, \quad (2.16)$$

$$H(x^* + \alpha^*) = x^*, \quad (2.17)$$

$$\sum_{i=1}^N u_i^* + \alpha^* = 0. \quad (2.18)$$

Again, we can solve this set of balance equations by transforming into a fixed point problem. One approach to solving equations (2.16) – (2.18) is to add an additional agent, $F_{N+1} = H$, and use the approach of Section 2.2.2 [40]. However, here we take a slightly different approach in which the denoising agent is applied in series, rather than in parallel.

In order to do this, we first specify a rescaled parallel operator \mathbf{F} and a novel consensus operator \mathbf{G}_H , given by

$$\mathbf{F} = \begin{pmatrix} F_1(v_1; \sqrt{N}\sigma) \\ \vdots \\ F_N(v_N; \sqrt{N}\sigma) \end{pmatrix} \text{ and } \mathbf{G}_H(\mathbf{v}) = \begin{pmatrix} H(\bar{\mathbf{v}}) \\ \vdots \\ H(\bar{\mathbf{v}}) \end{pmatrix}, \quad (2.19)$$

where $\bar{\mathbf{v}} = \sum_{i=1}^N v_i / N$.

In Theorem 2.3.1 below we show that we can solve the equilibrium conditions of (2.16) – (2.18) by finding the fixed-point of the map $\mathbf{T}_H = (2\mathbf{F} - \mathbf{I})(2\mathbf{G}_H - \mathbf{I})$. In practice, we can implement the G_H first computing an average across a distributed cluster, then applying our denoising algorithm, followed by broadcasting back a denoised, artifact-free version of the average.

Theorem 2.3.1 *Let $F_i, i = 1, \dots, N$, denote the proximal map of function f_i . Let maps \mathbf{F} and \mathbf{G}_H be defined by (2.19). Let \hat{x}^* denote N vertical copies of x^* . Then, $(x^*, \mathbf{u}^*) \in \mathbb{R}^n \times \mathbb{R}^{nN}$ is a solution to the equilibrium conditions of (2.16) – (2.18) if and only if the point $\mathbf{w}^* = \hat{x}^* - N\mathbf{u}^*$ is a fixed-point of map $\mathbf{T}_H = (2\mathbf{F} - \mathbf{I})(2\mathbf{G}_H - \mathbf{I})$ and $\mathbf{G}_H(\mathbf{w}^*) = \hat{x}^*$.*

Proof Proof is in Appendix C. ■

When \mathbf{T}_H is non-expansive, we can again compute the fixed-point $\mathbf{w}^* \in \mathbb{R}^{nN}$ using the Mann iteration

$$\mathbf{w}^{(k+1)} = \rho(2\mathbf{F} - \mathbf{I})(2\mathbf{G}_H - \mathbf{I})\mathbf{w}^{(k)} + (1 - \rho)\mathbf{w}^{(k)}. \quad (2.20)$$

Then, we can compute the x^* that solves the MACE conditions (2.16) – (2.18), or equivalently, the PnP conditions (2.14) – (2.15), as $x^* = H\bar{\mathbf{w}}^*$. Importantly, (2.20) provides a parallel approach to solving the PnP framework since the parallel operator \mathbf{F} typically constitutes the bulk of the computation, as compared to consensus operator \mathbf{G}_H .

In the specific case when the denoiser H is *firmly non-expansive*, such as a proximal map, we show in Lemma 2.3.2 that \mathbf{T}_H is non-expansive. While there is no such guarantee for any general H , in practice, we have found that this Mann iteration converges. This is consistent with previous experimental results that have empirically observed convergence of PnP [19, 23] for a wide variety of denoisers including BM3D [36], non-local means [41], or Deep residual CNNs [37, 40].

Lemma 2.3.2 *If \mathbf{F} and \mathbf{G}_H are defined by (2.19), and H is firmly non-expansive, then $\mathbf{T}_H = (2\mathbf{F} - \mathbf{I})(2\mathbf{G}_H - \mathbf{I})$ is non-expansive and the Mann iteration of (2.20) converges to the fixed point of \mathbf{T}_H .*

Proof The proof is in Appendix C. ■

The Algorithm 4 below shows the partial update version of the Mann iteration from equation (2.20). This version of the algorithm is practical since it only requires a single update of each proximal map per iteration of the algorithm.

Algorithm 4 Partial-update MACE with PnP priors

- 1: *Initialize:*
 - 2: $\mathbf{w}^{(0)} \leftarrow$ any value $\in \mathbb{R}^{nN}$
 - 3: $\mathbf{X}^{(0)} = \mathbf{G}_H(\mathbf{w}^{(0)})$
 - 4: $k \leftarrow 0$
 - 5: **while** not converged **do**
 - 6: $\mathbf{v}^{(k)} = (2\mathbf{G}_H - \mathbf{I}) \mathbf{w}^{(k)}$
 - 7: $\mathbf{X}^{(k+1)} = \tilde{\mathbf{F}}(\mathbf{v}^{(k)}; \sigma, \mathbf{X}^{(k)})$ ▷ Approximate $\mathbf{F}(\mathbf{v}^{(k)})$
 - 8: $\mathbf{w}^{(k+1)} = 2\mathbf{X}^{(k+1)} - \mathbf{v}$ ▷ $\approx (2\mathbf{F} - \mathbf{I})(2\mathbf{G}_H - \mathbf{I})\mathbf{w}^{(k)}$
 - 9: $\mathbf{w}^{(k+1)} \leftarrow \rho\mathbf{w}^{(k+1)} + (1 - \rho)\mathbf{w}^{(k)}$ ▷ Mann update
 - 10: $k \leftarrow k + 1$
 - 11: **end while**
 - 12: *Solution:*
 - 13: $x^* = H\bar{\mathbf{w}}^{(k)}$ ▷ Consensus solution
-

2.4 Experimental Results

Our experiments are divided into two parts corresponding to 2D and 3D reconstruction experiments. In each set of experiments, we analyze convergence and determine how the number of view-subsets affects the speedup and parallel-efficiency of the MACE algorithm.

Table 2.1(a) lists parameters of the 2D data sets. The first 2D data set was collected at the Lawrence Berkeley National Laboratory Advanced Light Source synchrotron and is one slice of a scan of a ceramic matrix composite material. The second 2D data set was collected on a GE Imatron multi-slice CT scanner and was reformatted into parallel beam geometry. For the 2D experiments, reconstructions are done with a image size of 512×512 , and the algorithm is implemented on a distributed compute cluster of 16 CPU nodes using the standard *Message Parsing Interface* (MPI) protocol. Source code for our distributed implementation can be found in [42].

Table 2.1(b) lists parameters of the 3-D parallel-beam CT dataset used for our supercomputing experiment. Notice that for the 3D data set, the reconstructions are computed with an array size of 1280×1280 , effectively doubling the resolution of the reconstructed images. This not only increases computation, but makes the system matrix larger, making reconstruction much more challenging. For the 3D experiments, MACE is implemented on the massive NERSC supercomputer using 1200 multi-core CPU nodes belonging to the Intel Xeon Phi Knights Landing architecture, with 68 cores on each node.

Table 2.1.

CT Dataset Description

(a) 2-D Datasets

Dataset	#Views	#Channels	Image size
Low-Res. Ceramic Composite (LBNL)	1024	2560	512×512
Baggage Scan (ALERT)	720	1024	512×512

(b) 3-D Dataset

Dataset	#Views	#Channels	Volume size
High-Res. Ceramic Composite (LBNL)	1024	2560	$1280 \times 1280 \times 1200$

2.4.1 Methods

For the 2D experiments, we compare our distributed MACE approach against a single-node method. Both the single-node and MACE methods use the same ICD algorithm for reconstruction. In the case of the single-node method, ICD is run on a single compute node that stores the complete set of views and the entire system matrix. Alternatively, for the MACE approach computation and memory is distributed

among N compute nodes, with each node performing reconstructions using a subset of views. The MACE approach uses partial updates each consisting of 1 pass of ICD optimization.

We specify the computation in units called *Equits* [43]. In concept, 1 equit is the equivalent computation of 1 full iteration of centralized ICD on a single node. Formally, we define an equit as

$$\# \text{ Equits} = \frac{(\# \text{ of voxel updates})}{(\# \text{ of voxels in ROI}) * (\# \text{ of view subsets})} .$$

For the case of a single node, 1 equit is equivalent to 1 full iteration of the centralized ICD algorithm. However, equits can take fractional values since non-homogeneous ICD algorithms can skip pixel updates or update pixels multiple times in a single iteration. Also notice this definition accounts for the fact the the computation of an iteration is roughly proportional to the number of views being processed by the node. Consequently, on a distributed implementation, 1 equit is equivalent to having each node perform 1 full ICD iteration using its subset of views.

Using this normalized measure of computation, the speedup due to parallelization is given by

$$\text{Speedup}(N) = N \times \frac{(\# \text{ of equits for centralized convergence})}{(\# \text{ of equits for MACE convergence})} ,$$

where again N is the number of nodes used in the MACE computation. From this we can see that the speedup is linear in N when the number of equits required for convergence is constant.

In order to measure convergence of the iterative algorithms, we define the NRMSE metric as

$$\text{NRMSE}(x, x^*) = \frac{\|x - x^*\|}{\|x^*\|} .$$

where x^* is the fully converged reconstruction.

All results using the 3D implemented on the NERSC supercomputer used the highly parallel 3-D Super-voxel ICD (SV-ICD) algorithm described in detail in [20]. The SV-ICD algorithm employees a hierarchy of parallelization to maximize utilization of each cluster nodes. More specifically, the 1200 slices in the 3D data set are

processed in groups of 8 slices, with each group of 8 slices being processed by a single node. The 68 cores in each node then perform a parallelized version of ICD in which pixels are processed in blocks called super-voxels. However, even with this very high level of parallelism, the SV-ICD algorithm has two major shortcomings. First, it can only utilize 150 nodes in the super-computing cluster, but more importantly, when the system matrix is too large to store on a single node, high resolution 3D reconstruction is impossible without the MACE algorithm.

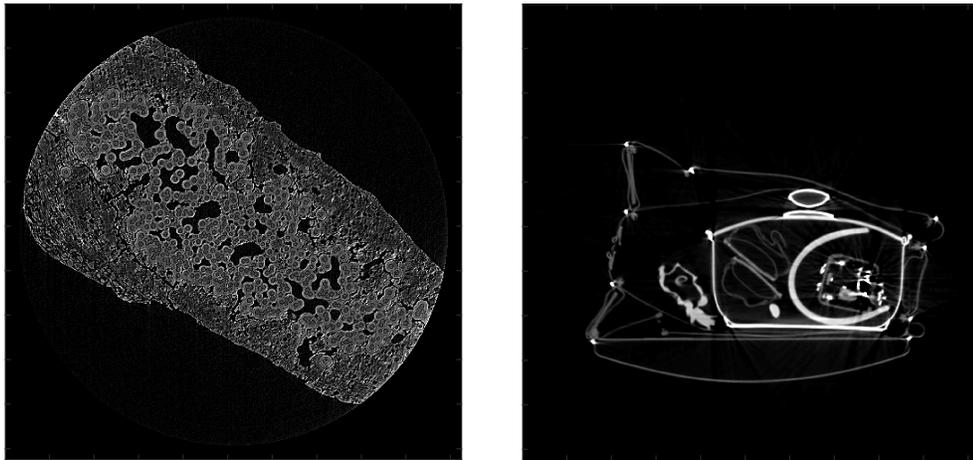
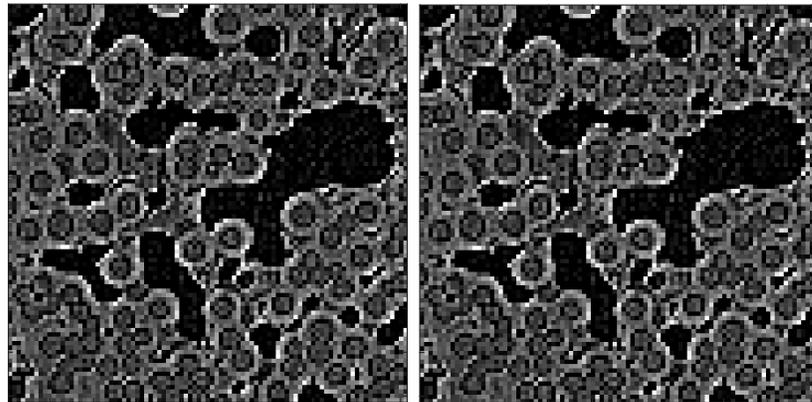


Fig. 2.2. Single node reconstruction for (left) Low-Res. Ceramic Composite dataset (right) Baggage Scan dataset.

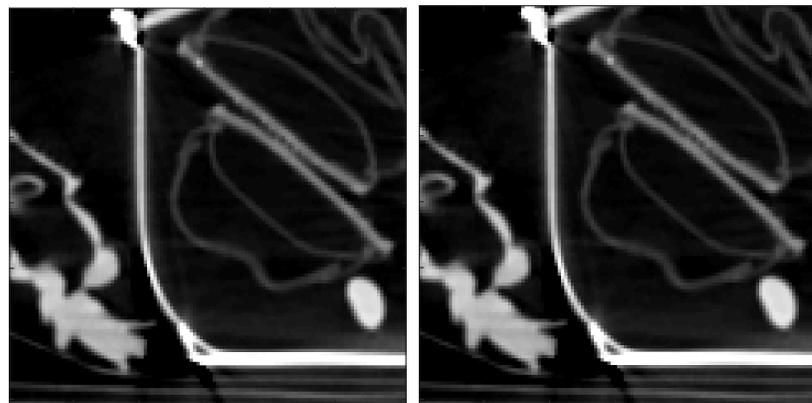
Table 2.2.

MACE convergence (equits) for different values of ρ , $N=16$.

Dataset	ρ				
	0.5	0.6	0.7	0.8	0.9
Low-Res. Ceramic	18.00	16.00	15.00	14.00	14.00
Baggage scan	12.64	10.97	10.27	9.52	12.40



(a) Single node reconstruction (b) MACE reconstruction, $N = 16$

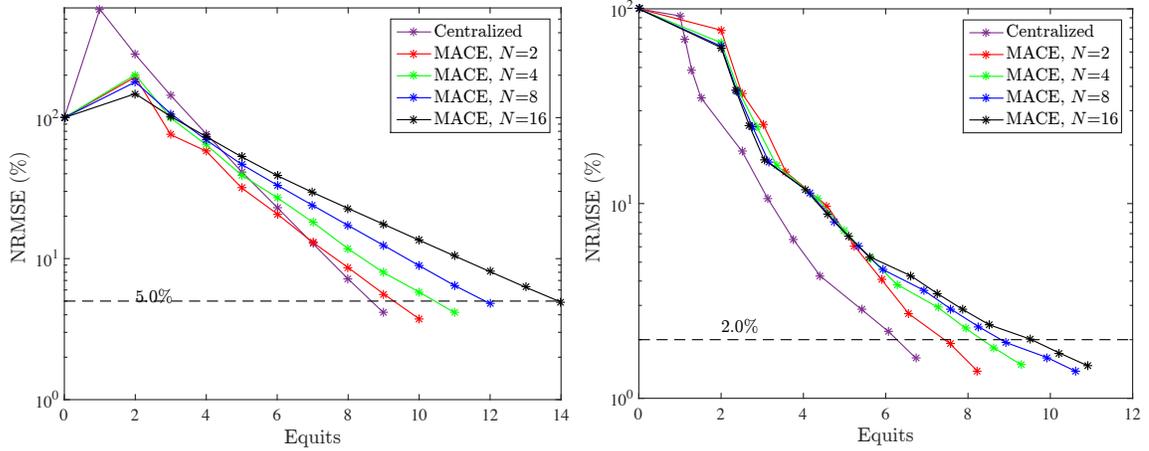


(c) Single node reconstruction (d) MACE reconstruction, $N = 16$

Fig. 2.3. Comparison of reconstruction quality for MACE method using 16 parallel nodes each processing $(1/16)^{th}$ of the views, against centralized method. (a) and (c) Centralized method. (b) and (d) MACE. Notice that both methods have equivalent image quality.

2.4.2 MACE Reconstruction of 2-D CT Dataset

In this section, we study the convergence and parallel efficiency of the 2D data sets of Table 2.1(a). Figure 2.2 shows reconstructions of the data sets, and Figure 2.3 compares the quality of the centralized and MACE reconstructions for zoomed-in regions of the image. The MACE reconstruction is computed using $N = 16$ compute nodes or equivalently $N = 16$ view-subsets. Notice that the MACE reconstruction is visu-



(a) Convergence for different #nodes, Low-Res. Ceramic dataset
 (b) Convergence for different #nodes, Baggage Scan dataset

Fig. 2.4. MACE convergence for different number of nodes, N , using $\rho = 0.8$: (a) Low-Res. Ceramic composite dataset (b) Baggage Scan dataset. Notice that number of equits tends to gradually increase with number of parallel processing nodes, N .

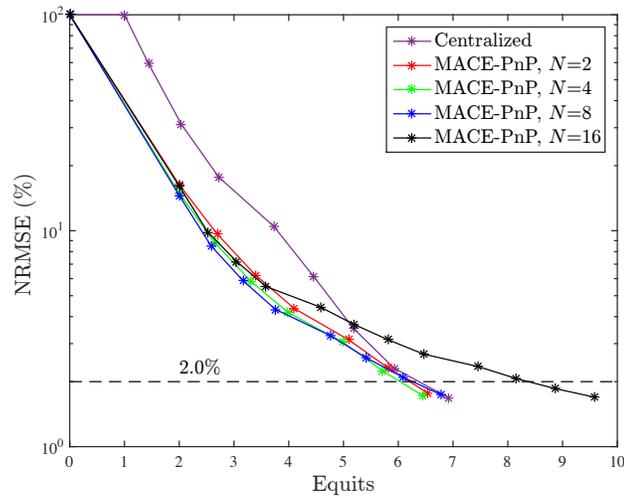
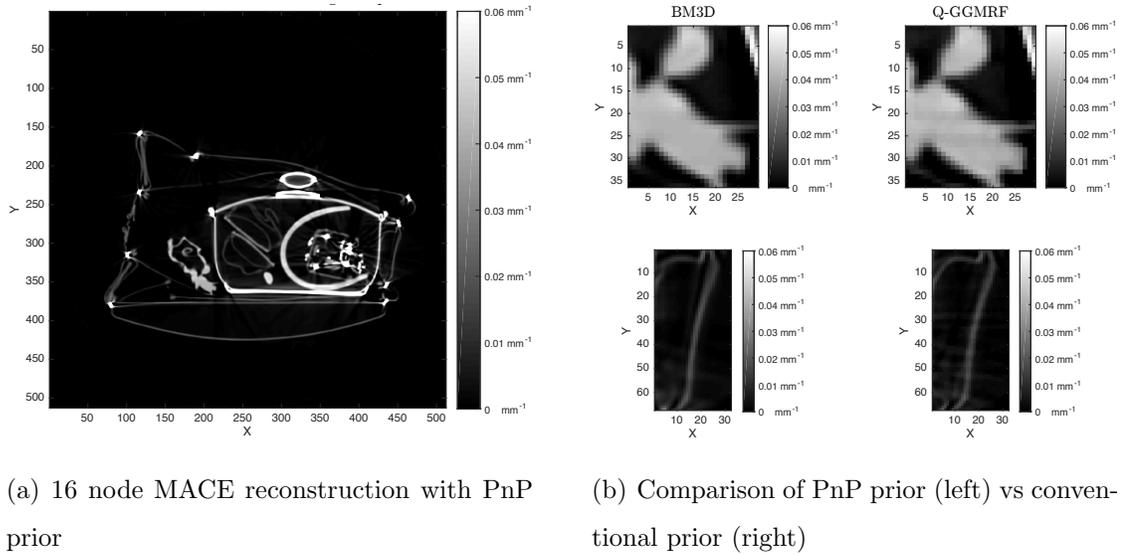
Table 2.3.

MACE memory usage for the system-matrix (gigabytes) as a function of number of nodes, N ¹

Dataset	N				
	1	2	4	8	16
Low-Res. Ceramic	14.83	7.42	3.71	1.86	0.93
Baggage scan	5.04	2.52	1.26	0.63	0.32

¹ System-matrix represented in sparse matrix format and floating-point precision

ally indistinguishable from the centralized reconstruction. However, for the MACE



(c) Convergence for different #nodes

Fig. 2.5. MACE-PnP reconstruction of Baggage Scan data set: (a) MACE PnP reconstruction using $N = 16$ nodes; (b) Zoomed-in regions of PnP versus conventional prior; (c) MACE-PnP convergence as a function of number of nodes N . Notice that PnP prior produces better image quality with reduced streaking and artifacts. In addition, the number of equits required for convergence of the MACE-PnP does not tend to increase significantly with number of nodes N .

reconstruction, each node only stores less than 7% of the full sinogram, dramatically reducing storage requirements.

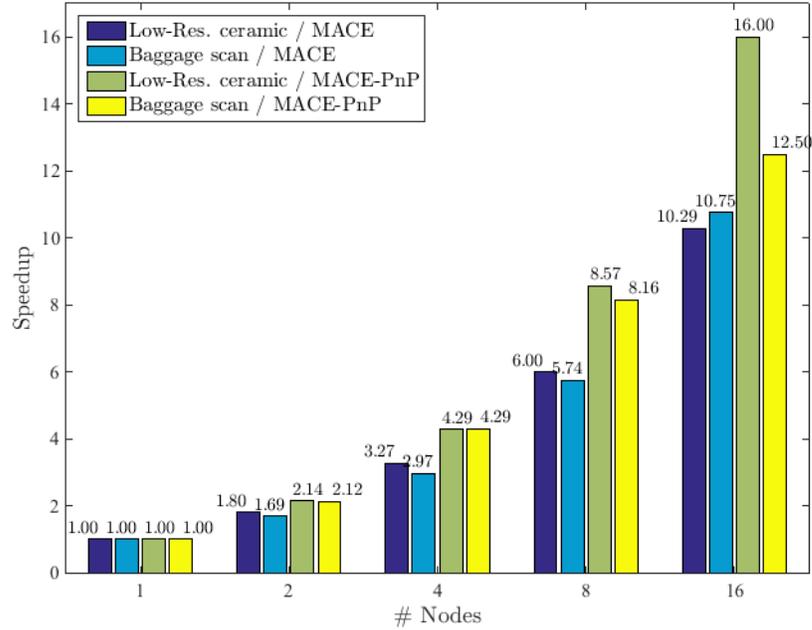


Fig. 2.6. MACE speedup as a function of the number of nodes, for different datasets and prior models. Importantly, note that in the case of PnP priors, we achieve a near linear speedup for both datasets.

Table 2.2 shows the convergence of MACE for varying values of the parameter ρ , with $N=16$. Notice that for both data sets, $\rho = 0.8$ resulted in the fastest convergence. In fact, in a wide range of experiments, we found that $\rho = 0.8$ was a good choice and typically resulted in the fastest convergence.

Figure 2.4 shows the convergence of MACE using a conventional prior for varying numbers of compute nodes, N . Notice that for this case of the conventional QGGMRF prior model, as N increased, the number of equits required for convergence tended to increase for both data sets.

Figure 2.5 shows results using the MACE with the PnP prior (MACE-PnP) with the Baggage Scan data set. Notice that the PnP prior results in improved image quality with less streaking and fewer artifacts. Also notice that with the PnP prior, the number of equits required for convergence shown in Figure 2.5(c) does not significantly increase with number of compute nodes, N .

Figure 2.6 summarizes this important result by plotting the parallel speed up as a function of number of nodes for both data sets using both priors. Notice that the MACE-PnP algorithm results in approximately linear speedup for $N \leq 8$ for both data sets, and near linear speedup up to $N = 16$. We conjecture that the advance prior tends to speed convergence due the stronger regularization constraint it provides.

Table 2.3 shows the system matrix memory as a function of the number of nodes, N . Note that MACE drastically reduces the memory usage per node by a factor of N .

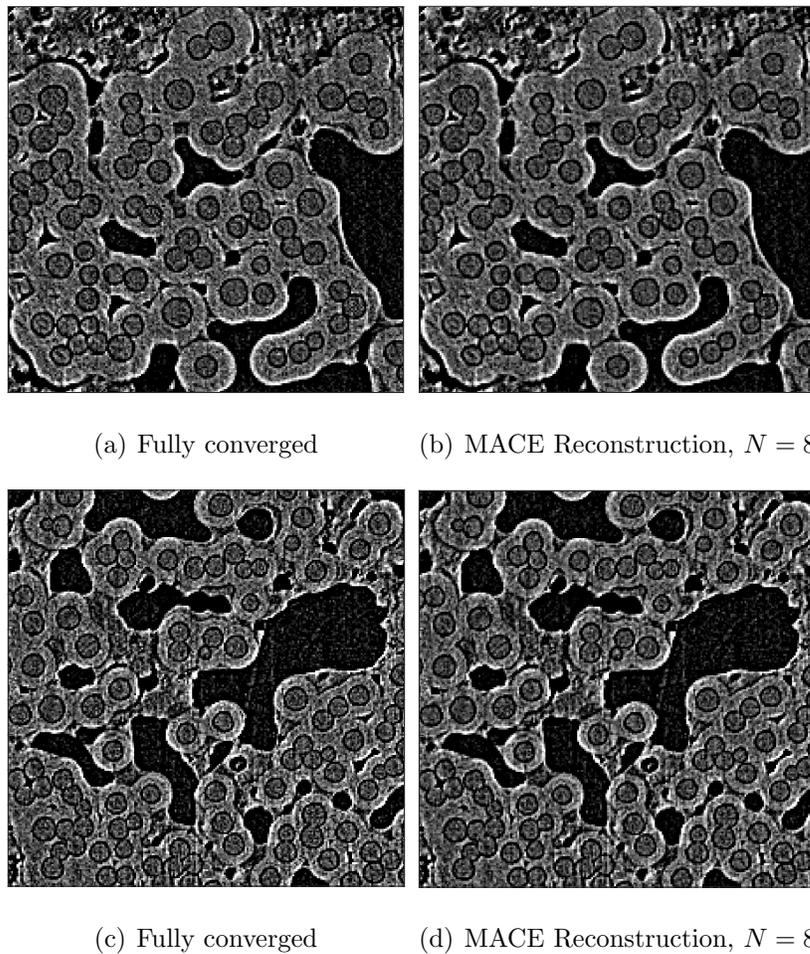


Fig. 2.7. Comparison of quality between (a),(c) fully converged result and (b),(d) the MACE reconstruction using 8 view-subsets on the NERSC supercomputer. Notice that both have equivalent image quality.

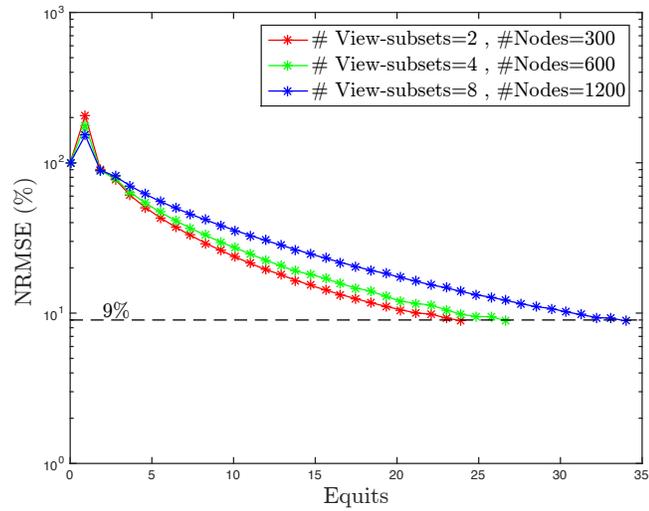


Fig. 2.8. MACE convergence for the High-Res. ceramic reconstruction on the NERSC supercomputer as a function of view-subsets (reconstruction size $1280 \times 1280 \times 1200$).

Table 2.4.

Computational performance for 3-D MACE reconstruction on the NERSC supercomputer as a function of #view-subsets

#View-subsets	1	2	4	8
#Nodes	150	300	600	1200
#Cores	10,200	20,400	40,800	81,600
Memory usage ² (GB)	-	25.12	12.56	6.28
#Equits	-	23.91	26.67	34.03
#Time (s)	-	275	154	121
MACE Algorithmic Speedup	-	2	3.58	5.62
Machine-time Speedup	-	2	3.57	4.55

² rough estimate when system-matrix is represented in conventional sparse matrix format, floating-point precision

2.4.3 MACE Reconstruction for large 3D datasets

In this section, we study the convergence and parallel efficiency of the 3D data sets of Table 2.1(b) using the MACE algorithm implemented on the NERSC supercomputer.

All cases use the SV-ICD algorithm for computation of the reconstructions at individual nodes with a Q-GGMRF as prior model and $\rho = 0.8$ as value of Mann parameter. As noted previously, for this case the system matrix is so large that it is not possible to compute the reconstruction with $N = 1$ view subset. So in order to produce our reference reconstruction, we ran 40 equits of MACE with $N = 2$ view subsets, which appeared to achieve full convergence.

Figure 2.7 compares zoomed in regions of the fully converged result with MACE reconstructions using $N = 8$ view subsets. Notice both have equivalent image quality.

Figure 2.8 shows the convergence of MACE as a function of number of equits. Notice that, as in the 2D case using the Q-GGMRF prior, the number of equits tends to increase as the number of view subsets, N , increases.

Table 2.4 summarizes the results of the experiment as a function of the number of view subsets, N . Notice that the memory requirements for reconstruction drop rapidly with the number of view subsets. This makes parallel reconstruction practical for large tomographic data sets. In fact, the results are not listed for the case of $N = 1$, since the system matrix is too large to store on the nodes of the NERSC supercomputer. Also, notice that parallel efficiency is good up until $N = 4$, but it drops off with $N = 8$.

2.5 Conclusion

In this chapter of the thesis, we proposed a novel MACE algorithm that distributes both the memory and computation of iterative CT reconstruction across a large number of parallel nodes. MACE integrates multiple sparse-view reconstructions across the compute cluster into a high-quality reconstruction. Further, we introduced a vari-

ant of MACE called MACE-PnP that supports the use of advanced PnP denoisers that improve reconstruction quality. We proved that MACE converges to the MAP estimate, while MACE-PnP converges to the solution of the serial PnP framework. In order to make MACE feasible for practical applications, we proposed a partial-update implementation that utilizes non-iterative proximal updates and proved convergence for the same. We analyzed the convergence and speedup of our method on a distributed memory system with real CT data. Finally, we demonstrated an implementation of our method on a massive supercomputer that enables large-scale reconstructions in real-time.

3. FAST ALGORITHMS FOR MODEL-BASED IMAGING THROUGH TURBULENCE

3.1 Introduction

Digital holography (DH) systems can acquire high-resolution images of far-away targets using a coherent laser source and an image-sensor such as a Focal plane array (FPA). The main advantage of DH systems is their ability to coherently detect weak signal fields that are modulations of the source optical field. The resulting complex images can be processed to remove severe distortions that would not otherwise be possible with conventional non-coherent detection. Consequently, DH imaging has huge potential for real-time remote-sensing and surveillance (ISR) applications.

Figure 3.1 illustrates remote-sensing using a DH system. Reflected light from the target is focused onto the FPA using a lens-array. This weak received field is then demodulated by mixing it with a strong reference field that is identical to the source but has a linear pixel-wise phase offset ¹. This demodulation technique is known as *optical heterodyning* [44, 45], and the resulting FPA measurement is the *hologram*. While most DH systems use simple Fourier-based methods to form the target image from the hologram, advanced image-formation methods such as [45, 46] can significantly improve quality.

The presence of deep atmospheric turbulence between the target and image sensor can pose a strong challenge to DH systems. As shown in Figure 3.1, turbulence distorts the point-wise phase of the pupil-plane optical field ², or equivalently, the Fresnel diffraction pattern of the target. In this case, DH systems must remove these phase-errors in order to recover a focused image of the target.

¹projecting the field at an oblique angle onto the FPA produces such a phase-offset

²We assume isoplanatic atmospheric conditions, which allows us to accumulate the phase-errors during wave propagation in the pupil-plane

Image sharpening (IS) [47–49] and model-based iterative reconstruction (MBIR) [44–46,50] are perhaps the two major classes of algorithms for estimating the unknown phase errors from DH images. The IS and DH-MBIR methods primarily differ in two ways. First, the IS method estimates the complex-valued *reflection coefficient*, which typically has abrupt spatial variations due to *speckle*. Alternatively, DH-MBIR estimates the real-valued *reflectance* [44,50], which is typically much smoother as seen in Figure 3.4. Second, IS and DH-MBIR use very different mathematical frameworks, and consequently, different iterative computations to estimate the phase-errors and the target image.

IS methods are based on maximizing the sharpness of the intensity image specified by the magnitude-squared of reflection coefficient, g . More specifically, IS uses simple Fourier-inversion methods to express g as a function of the detected pupil-plane field, y , and the unknown phase-errors, ϕ . Then, the phase-errors are estimated by maximizing a sharpness metric associated with the image $|g|^2$. However, g is affected by speckle noise,³ which limits the quality of both the sharpened image and the estimated phase-errors [44].

In contrast, DH-MBIR methods use a Bayesian framework to jointly estimate the target’s reflectance, r and the unknown phase-errors, ϕ from the detected pupil-plane field y . The key advantage of DH-MBIR is that DH-MBIR estimates the reflectance, r , which is much smoother than the reflection coefficient, $|g|^2$, estimated by IS. Therefore, since the unknown has fewer degrees of freedom, the estimation problem can be more accurately solved with with less data.

In this thesis chapter, we propose a method to significantly speedup MBIR for real-time DH applications. Our approach uses surrogate optimization to simplify the pixel-wise phase-error and reflectance updates that dominate the computation. We also show how fast parallel SIMD vector processing instructions together with cache prefetching can be used to speed these operations even on a single core of a modern

³ g is speckled since each pixel value can be modeled as a sum of many small scatterers, whose phase is a uniform random variable $\in (0, 2\pi)$

CPU. In our experiments with simulated datasets we verify the convergence of our method and show that we achieve dramatic speedup over the original DH-MBIR approach of [44]. More specifically, we show that on a single CPU core our method accelerates the reflectance and phase-error updates by a factor of 15.1x and 37.6x respectively as compared to the original approach, and consequently accelerates each DH-MBIR iteration by a factor of 23.7x.

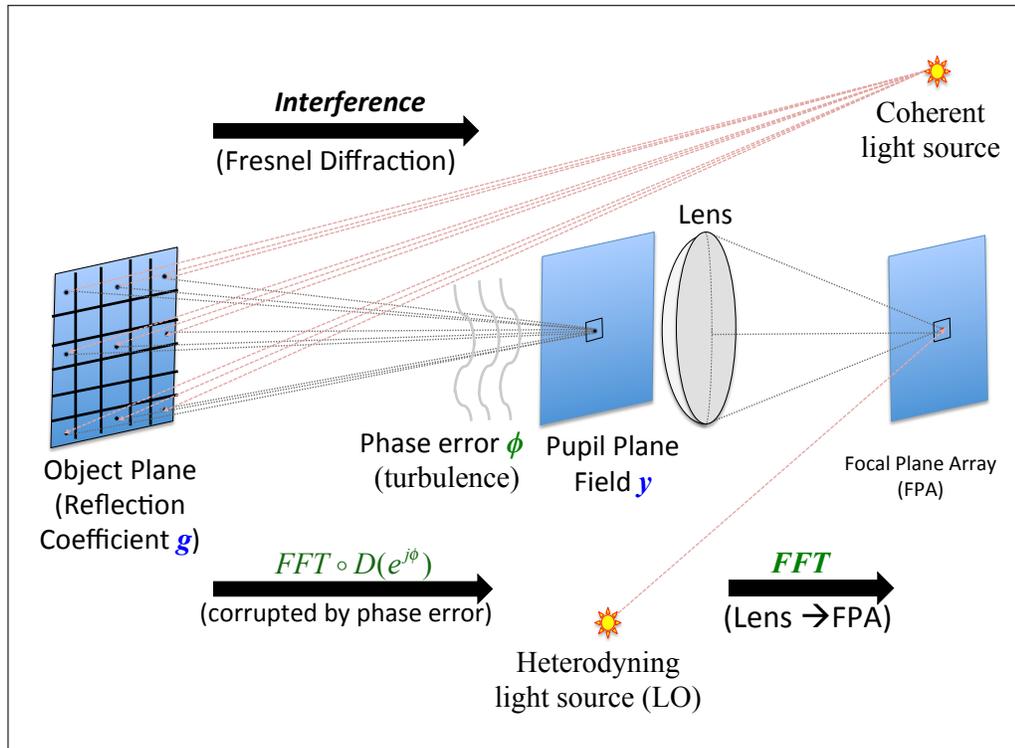


Fig. 3.1. Illustration of a Digital Holography system for remote-sensing. A laser source illuminates the target and a lens focuses the reflected light onto the FPA. The detected field is optically heterodyned with a strong reference field to form a hologram. However, atmospheric turbulence between the target and lens can corrupt the pupil-plane field by inducing phase-errors. These phase-errors must be removed prior to recovering the target image from the hologram.

3.2 Stastical Framework for DH-MBIR

In this section, we briefly describe the forward model of the DH system and summarize the MBIR reconstruction approach developed by Pellizari et al. in [44].

3.2.1 DH Forward Model

Following from Figure 3.1, the detected pupil-plane field $y \in \mathbb{C}^n$ is given by

$$y = \mathcal{D}(a)\mathcal{D}(e^{j\phi})Fg + w \quad (3.1)$$

where $g \in \mathbb{C}^n$ represents the object's reflection coefficient, $F \in \mathbb{C}^{n \times n}$ represents a 2-D Discrete Spatial Fourier transform (DSFT) that computes the Fresnel diffraction integral ⁴, $a \in \{0, 1\}^n$ denotes the aperture mask, $\phi \in (-\pi, \pi)^n$ represents the phase-errors in the pupil-plane caused by turbulence, and $w \sim \mathcal{N}(0, \sigma_w^2 I)$ represents white noise. So, we can specify (3.1) using the following conditional distribution

$$p(y|g, \phi) = \frac{1}{(2\pi\sigma_w^2)^{n/2}} \exp \left\{ -\frac{1}{2\sigma_w^2} \|y - A_\phi g\|^2 \right\}, \quad (3.2)$$

where $A_\phi = \mathcal{D}(a)\mathcal{D}(e^{j\phi})F$ is the system matrix ⁵.

Since we specify the speckle-free reflectance r as $E[|g|^2]$, we can model g as $\mathcal{N}(0, \mathcal{D}(r))$ shown below

$$p(g|r) = \frac{1}{(\pi^n |D(r)|)^{1/2}} \exp \{ -g^H D(r)^{-1} g \}. \quad (3.3)$$

3.2.2 MAP Estimation for DH Reconstruction

In [44], Pellizari et al. formulate the joint estimation of reflectance $r \in \mathbb{R}^{+n}$ and phase-errors ϕ as a Maximum-a-posteriori (MAP) estimation problem given by

$$(\hat{r}, \hat{\phi}) = \underset{r, \phi}{\operatorname{argmax}} \log p(y|r, \phi) + \log p(r) + \log p(\phi), \quad (3.4)$$

⁴we neglect the quadratic phase terms in the Fresnel integral

⁵As in [44], we approximate $D(a) = I$, so that A_ϕ is orthogonal

where $p(y|r, \phi)$ denotes the likelihood model, and, $p(r)$ and $p(\phi)$ denote the prior models for r and ϕ respectively. However, computing the likelihood model in (3.4) is not tractable without the knowledge of missing information, g . To overcome this issue, Pellizari et al. in [44] use the Expectation-Maximization (EM) algorithm to reformulate the MAP estimate of (3.4) as the following iterative update

$$(r^{(k)}, \phi^{(k)}) = \underset{r, \phi}{\operatorname{argmax}} Q(r, \phi; r^{(k-1)}, \phi^{(k-1)}), \quad (3.5)$$

where the function Q is specified by

$$Q(r, \phi; r', \phi') = E_g [\log p(y, g|r, \phi)|y, r', \phi'] + \log p(r) + \log p(\phi). \quad (3.6)$$

While the key ideas for deriving the Q function are briefly provided in the Appendix, the comprehensive derivation is available in [44].

Algorithm 5 shows the pseudo-code for implementing the EM algorithm specified by (3.5) and (3.6). Each of the M-steps represented by lines 11 and 14 is implemented using only one pass of Iterative Coordinate Descent (ICD) optimization [17], and, the prior models for phase and reflectance are specified by Markov Random Field (MRF) priors.

3.3 Accelerating MBIR

In this section, we propose a method that drastically speeds up the M-steps in Algorithm 5 that dominate the computation by using surrogate optimization and SIMD parallelization.

3.3.1 Fast Method for Reflectance Update

Each greedy pixel-wise r update is given by:

$$r_s \leftarrow \underset{u>0}{\operatorname{argmin}} \left\{ \frac{|\mu_s|^2 + C_s}{u} + \log u + \sum_{j \in \partial s} b_{s,j} \rho(u - r_j) \right\}, \quad (3.7)$$

where $\rho(\cdot)$ is a Gibbs-prior potential function, more specifically a Q-Generalized Gaussian Markov random field prior (Q-GGMRF) [35] in this case.

Algorithm 5 EM Algorithm for DH-MBIR

```

1: // Initial values:  $(r', \phi') \in (\mathbb{R}^{+n}, (-\pi, \pi)^n)$ 
2: // Assumption:  $\mathcal{D}(a) = I$ , so  $A_\phi^H A_\phi = nI$ 
3:
4: while not converged do
5:   // E-step: Find Posterior mean  $\mu$  and covariance  $C$ 
6:    $C = \left( \frac{1}{\sigma_w^2} A_\phi^H A_\phi + \mathcal{D}(r')^{-1} \right)^{-1} = \mathcal{D} \left( \frac{r' \sigma_w^2}{r' n + \sigma_w^2} \right)$ 
7:    $\mu = C A_\phi^H y$ 
8:    $z = A_0 \mu$ 
9:
10:  // M-step: Reflectance update
11:   $r \leftarrow \operatorname{argmin}_{r \in \mathbb{R}^{+n}} \left\{ \sum_{s=1}^n \left( \frac{|\mu_s|^2 + C_{s,s}}{r_s} + \log r_s \right) - \log p(r) \right\}$ 
12:
13:  // M-step: Phase-error update
14:   $\phi \leftarrow \operatorname{argmin}_\phi \left\{ -\frac{2}{\sigma_w^2} \operatorname{Real} (y^H \mathcal{D}(e^{j\phi}) z) - \log p(\phi) \right\}$ 
15:   $(r', \phi') \leftarrow (r, \phi)$ 
16: end while

```

In [45, 46, 50], Pellizari et al. compute the above update almost exactly using a method based on derivative-rooting. However, this approach requires finding the root(s) of a cubic polynomial which is computationally expensive.

In order to speed-up the r update, we replace the objective function in (3.7) with a suitable surrogate that is easy to minimize. However, designing a surrogate typically requires the objective function to be convex. So first, we substitute the $+\log(\cdot)$ function in (3.7) with its 1st order approximation, which yields a convex optimization of the form

$$\hat{r}_s \leftarrow \operatorname{argmin}_{u>0} \left\{ \frac{|\mu_s|^2 + C_s}{u} + \frac{u}{r_s} + \sum_{j \in \partial s} b_{s,j} \rho(u - r_j) \right\}. \quad (3.8)$$

We then design a quadratic surrogate function for (3.8) based on the Linear Interpolation of Derivative (LID) method [51]. We define the function $l(\cdot) : \mathbb{R}^+ \rightarrow \mathbb{R}$ as

$$l(u) = \frac{|\mu_s|^2 + C_s}{u} + \frac{u}{r_s}. \quad (3.9)$$

The first and second derivatives of $l(\cdot)$ are given by

$$l'(u) = -\frac{(|\mu_s|^2 + C_s)}{u^2} + \frac{1}{r_s}, \quad l''(u) = 2\frac{(|\mu_s|^2 + C_s)}{u^3}.$$

In accordance with the LID method, we specify a quadratic surrogate function that matches the derivative of $l(\cdot)$ defined by (3.9) at 2 different points, r_s and α_s , where r_s is the current value of the pixel being updated and α_s is another suitably chosen value in the search direction. In this case, our reflectance update is given by

$$\hat{r}_s \leftarrow \operatorname{argmin}_{u \in \mathcal{S}} \left\{ \frac{\theta_2}{2} u^2 + \theta_1 u + \sum_{j \in \partial s} b_{s,j} \rho(u - r_j) \right\}, \quad (3.10)$$

where \mathcal{S} denotes the search interval $(\min(r_s, \alpha_s), \max(r_s, \alpha_s))$, and, θ_2, θ_1 are given by

$$\begin{aligned} \theta_2 &= \frac{l'(r_s) - l'(\alpha_s)}{r_s - \alpha_s} = (|\mu_s|^2 + C_s) \left(\frac{r_s + \alpha_s}{r_s^2 \alpha_s^2} \right) \\ \theta_1 &= l'(r_s) - \theta_2 r_s = -\frac{(|\mu_s|^2 + C_s)}{r_s^2} + \frac{1}{r_s} - \theta_2 r_s. \end{aligned}$$

We select α_s that bounds our search interval as shown below

$$\alpha_s = \begin{cases} \frac{r_s}{1 + M} & \text{if } l'(r_s) + \sum_{j \in \partial s} b_{s,j} \rho'(r_s - r_j) > 0 \\ r_s(1 + M) & \text{e.w. ,} \end{cases}$$

where $M > 0$. In practice, we suitably vary M such that it decays slowly with the number of iterations.

We can replace the symmetric Gibbs-prior model in (3.10) with a quadratic surrogate function of its own. Consequently, (3.10) further simplifies to

$$\hat{r}_s \leftarrow \operatorname{argmin}_{u \in \mathcal{S}} \left\{ \frac{\theta_2}{2} u^2 + \theta_1 u + \sum_{j \in \partial s} b_{s,j} \frac{\tau_j}{2} (u - r_j)^2 \right\}, \quad (3.11)$$

where τ_j is given by

$$\tau_j = \begin{cases} \frac{\rho'(r_s - r_j)}{(r_s - r_j)} & r_s \neq r_j \\ \rho''(0) & r_s = r_j. \end{cases}$$

So, the surrogate-based reflectance update is given by

$$\hat{r}_s \leftarrow \text{clip} \left\{ - \left(\frac{\theta_1 - \sum_{j \in \partial s} b_{s,j} \tau_j r_j}{\theta_2 + \sum_{j \in \partial s} b_{s,j} \tau_j} \right), \mathcal{S} \right\}, \quad (3.12)$$

where the $\text{clip}\{x, (a, b)\}$ here denotes x when $x \in (a, b)$, a when $x < a$, and b otherwise.

3.3.2 Fast Method for Phase-error Update

The exact pixel-wise phase-error update is given by

$$\phi_s \leftarrow \underset{u \in [-\pi, \pi)}{\text{argmin}} \left\{ -m_s \cos(u - \varphi_s) + \sum_{j \in \partial s} b_{s,j} \rho(u - \phi_j) \right\}, \quad (3.13)$$

where $m_s = 2y_s z_s^* / \sigma_w^2$ and $\rho(\cdot)$ is the phase-wrapped quadratic error with the form

$$\begin{aligned} \rho(\Delta) &= ([(\Delta + \pi) \bmod 2\pi] - \pi)^2 \\ &= \min_n (\Delta + 2\pi n)^2. \end{aligned} \quad (3.14)$$

In [44, 45] Pellizari et al. estimate the phase-errors on a lower-resolution grid and compute the above ICD update using the Golden-section search (GSS) method. However, the GSS method is computationally expensive since it involves nested evaluations of the local cost function.

We devise a faster update strategy for the ICD update. We first approximate the $\cos(\cdot)$ function in (3.13) with its Taylor's series expansion of $\cos(\Delta) = 1 - \frac{1}{2}\Delta^2$ to yield approximate ICD update of

$$\phi_s \leftarrow \underset{u \in [-\pi, \pi)}{\text{argmin}} \left\{ \frac{1}{2} m_s (u - \varphi_s)^2 + \sum_{j \in \partial s} b_{s,j} \rho(u - \phi_j) \right\}. \quad (3.15)$$

Substituting in the form of $\rho(\cdot)$ from (3.14), yields the update

$$\phi_s \leftarrow \operatorname{argmin}_{u \in [-\pi, \pi)} \left\{ \frac{1}{2} m_s (u - \varphi_s)^2 + \sum_{j \in \partial s} b_{s,j} \min_{n_j} (u - \phi_j + 2\pi n_j)^2 \right\}. \quad (3.16)$$

While exact solution of equation (3.16) is difficult, we can get a local minimum using alternating minimization. One iteration of alternating minimization is given by

$$n_j \leftarrow \operatorname{argmin}_{m \in \mathcal{Z}} (\phi_s - \phi_j + 2\pi m)^2, \quad j \in \partial s \quad (3.17)$$

$$\phi_s \leftarrow \operatorname{argmin}_{u \in [-\pi, \pi)} \left\{ \frac{m_s}{2} (u - \varphi_s)^2 + \sum_{j \in \partial s} b_{s,j} (u - \phi_j + 2\pi n_j)^2 \right\}. \quad (3.18)$$

Figure 3.2 illustrates how the above method approximates the wrapped-phase penalty function in (3.15). Note that since $\phi_j, \phi_s \in [-\pi, \pi)$, we can show that (3.17) specifies $n_j \in \{-1, 0, 1\}$, and further, $\phi_j - 2\pi n_j \in (-2\pi, 2\pi)$. Consequently, equation (3.18) can be simplified further as

$$\phi_s \leftarrow \tilde{\phi}_s + 2\pi k_s \quad (3.19)$$

where $\tilde{\phi}_s$ is defined as

$$\tilde{\phi}_s = \frac{m_s \varphi_s + 2 \sum_{j \in \partial s} b_{s,j} (\phi_j - 2\pi n_j)}{m_s + 2 \sum_{j \in \partial s} b_{s,j}},$$

and k_s is defined as

$$k_s = \begin{cases} 1 & \tilde{\phi}_s < -\pi \\ 0 & \tilde{\phi}_s \in [-\pi, \pi) \\ -1 & \tilde{\phi}_s > \pi. \end{cases}$$

3.3.3 SIMD Parallelization of ICD updates

ICD optimization is a serial pixel-wise greedy minimization method. However, from (3.7) and (3.13) we notice that the ICD updates for those pixels that do not form pair-wise MRF cliques with one another are independent and can be computed simultaneously. For example, Fig 3.3(a) shows that in the case of a MRF with symmetric 8-point neighborhood, we can specify a tiled pixel-grid of 4 different colors

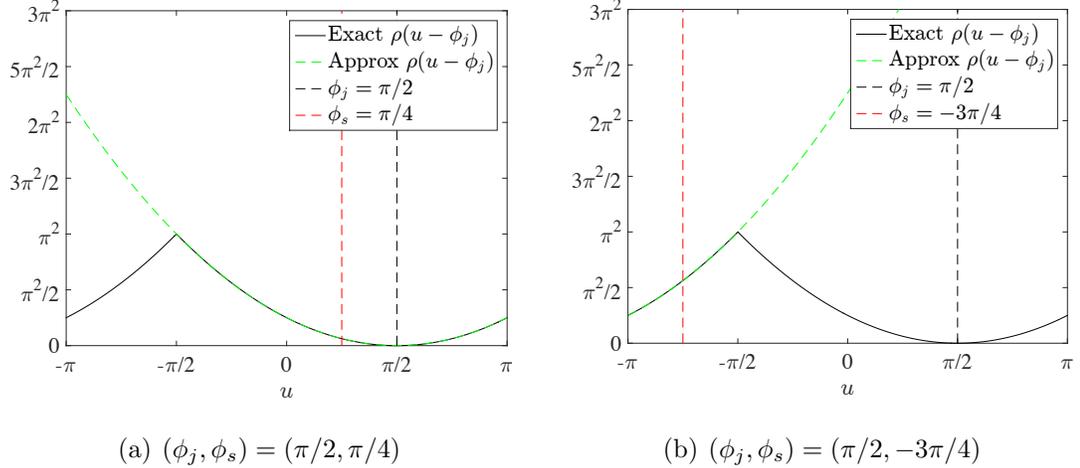


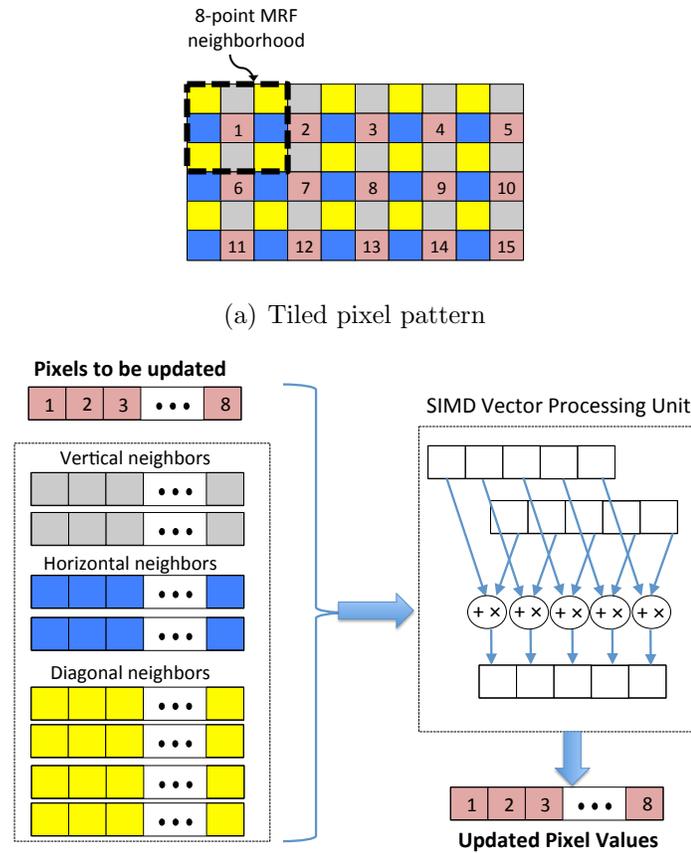
Fig. 3.2. Approximation of the wrapped-phase penalty function $\rho(u - \phi_j)$, $j \in \partial s$, for the s -th pixel ICD update. Sub-figures (a) and (b) show how the approximation varies with different values of (ϕ_j, ϕ_s) .

where ICD updates for pixels of the same color are fully disassociated. Furthermore, the simplified ICD updates represented by (3.12) and (3.19) can be fully implemented by basic addition and multiply operations⁶. So, we can utilize SIMD units that perform fast vector addition and multiplication on a single CPU core to update multiple disassociated pixels at a time. Figure 3.3(b) illustrates our idea of ICD parallelization using SIMD processing. Importantly, note that the required data must be packed into contiguous arrays prior to SIMD computation.

However, our above approach has one key limitation. Accessing the data in an interleaved manner adds significant overhead that reduces the benefit of the SIMD parallelism. To overcome this issue, we provide an alternate method that negates the need to sample data in a tiled fashion.

We alternatively propose a simple row-wise update strategy which extends the SIMD vector processing method of Figure 3.3(b) towards simultaneously computing the ICD updates for pixels in contiguous blocks within the same row. In order to

⁶We choose Q-GGMRF parameters [35] in (3.7) as $(q, p) = (2, 1)$ or $(2, 2)$ so that its surrogate coefficients τ_j in (3.11) can be computed easily



(b) Parallel updates for pixels of same color using SIMD operations

Fig. 3.3. SIMD Parallelization of ICD updates: (a) A tiled grid where pixels of the same color have independent ICD updates (b) SIMD processing units of a CPU can be used to update multiple pixels of the same color at a time using fast vector add and multiply operations.

explain why this method works in practice, let us assume that within the symmetric 8-point MRF distribution depicted in Figure 3.3(a), the contribution of the horizontal left neighbor is small compared to the other 7 neighbors put together. In this case, we can show that our row-wise SIMD method approximates serial pixel-wise minimization in raster order.

So, our row-wise update strategy provides dual benefits of fast memory access and SIMD parallelism. Our experiments demonstrate that this method significantly speeds-up ICD computation without affecting convergence.

3.4 Experimental Results

3.4.1 Method

In this section, we compare the computational performance, convergence and reconstruction quality of our fast DH-MBIR method against the exact DH-MBIR method of [44] using simulated datasets. We implement both MBIR methods on an Intel Xeon CPU (E5-2660 v3) in ANSI C with single-threading. The source code was compiled with the Intel *icc* compiler (ver. 17.0.1) which automatically converts the tight vectorizable loops into AVX2 instructions for SIMD processing. We implement all Fourier-based operations in MBIR using FFT routines from the Intel Math Kernel Library (MKL).

For our experiments, we simulate 6 different datasets, each based on the same object reflectance shown in Figure 3.4 but a different phase-error screen and reflection-coefficient.

We simulate the phase-errors in the pupil-plane based on a Power Spectral density (PSD) modeling technique [52,53]. For turbulent conditions, the 2-D spatial distribution of the phase-errors can be modeled as a *Kolmogorov* PSD in the Fourier domain (see [53], equation (1)). So, we generate the phase-errors by first scaling white noise in the Fourier domain with a Kolmogorov PSD and then applying an inverse FFT [44,53]. It is worth noting that the PSD incorporates a key parameter known as the *Fried* coherence length, r_0 , that determines the spatial correlation of the phase-errors. We specifically parameterize half of our simulated phase-error screens by $D_a/r_0 = 10$, and the other half by $D_a/r_0 = 20$, where D_a denotes aperture diameter. The top row of Figure 3.6(c) and (d) illustrates the simulated phase-screens.

We generate the reflection-coefficient based on the given object reflectance by sampling from the i.i.d complex Gaussian distribution specified by (3.3). Subsequently, we simulate the detected field in the pupil-plane based on the forward model specified by (3.1).

For most experiments, the grid-size for both the pupil-plane detection and the reconstruction is 256×256 . For CPU timing experiments alone we use a grid size of 128×128 .

We quantify the quality of our phase-error estimate based on the Strehl ratio metric specified by

$$\text{Strehl ratio} = \frac{\iint_{\text{pupil}} a(x, y) \exp \left\{ j(\phi(x, y) - \hat{\phi}(x, y)) \right\} dx dy}{\iint_{\text{pupil}} a(x, y) dx dy},$$

where a represents the aperture mask and $\phi - \hat{\phi}$ represents the estimation error.

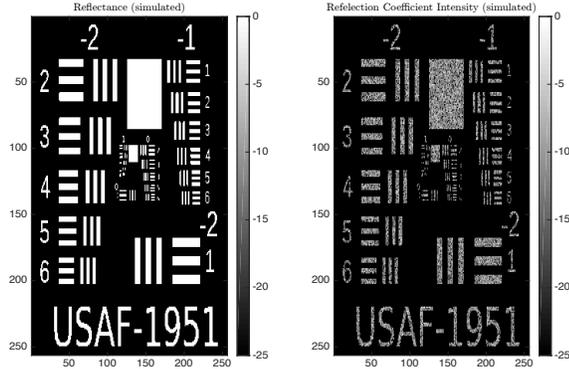


Fig. 3.4. Example depicting object reflectance, r and simulated reflection coefficient, g . Unlike $|g|^2$ (right), the reflectance r (left) is speckle-free and smooth.

3.4.2 Results

Table 3.1 shows the per-iteration CPU time for the exact and fast DH-MBIR methods. Notice that even without utilizing SIMD processing, the surrogate ICD updates

Table 3.1.
CPU Time per EM iteration

DH-MBIR Method	E-step	Reflectance update ¹		Phase-error update ²	Full Iteration ³
		$(q, p) = (2, 1)$	$(q, p) = (2, 2)$		
Exact	0.13 ms	1.25 ms	0.80 ms	9.02 ms	9.95 ms
Fast (No SIMD)	0.13 ms	0.53 ms	0.33 ms	0.38 ms	0.84 ms
Fast (with SIMD)	0.13 ms	0.10 ms	0.05 ms	0.24 ms	0.42 ms

¹ Prior model is a Q-GGMRF with 4-point neighborhood and power-parameters specified by (q, p)

² Prior model is a wrapped-phase GMRF with 8-point neighborhood

³ full iteration time calculated for case $(q, p) = (2, 2)$

Table 3.2.
Speedup of Fast DH-MBIR over Exact DH-MBIR

Reflectance update		Phase-error update	Full iteration
$(q, p) = (2, 1)$	$(q, p) = (2, 2)$		
12.50	15.09	37.58	23.7

are significantly faster than the exact ICD updates. In particular, we observe more than a 20x reduction in time for the phase-error estimation that dominates the exact DH-MBIR method. Furthermore, notice that SIMD parallelization provides additional acceleration of the surrogate ICD updates. Specifically, this speedup resulting

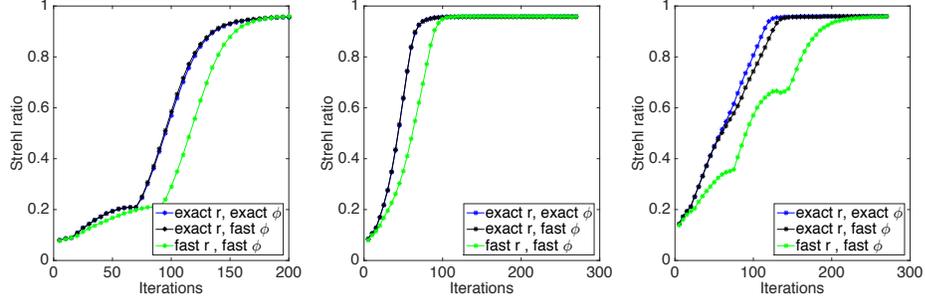
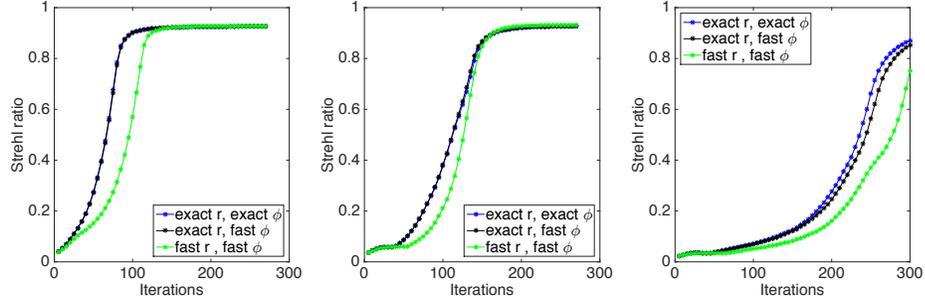
(a) 3 different datasets simulated with $D_a/r_0 = 10$ (b) 3 different datasets simulated with $D_a/r_0 = 20$

Fig. 3.5. Effect of fast updates for estimating reflectance, r , and phase-error, ϕ , on the convergence rate of DH-MBIR. The number of iterations for DH-MBIR convergence is unchanged when we use fast updates for ϕ estimation instead of exact updates, but increases when we use fast updates for r estimation in place of exact updates.

from SIMD is more prominent in the case of reflectance estimation as compared to phase-error estimation ⁷.

Table 3.2 shows the speedup of our fast DH-MBIR method over the exact DH-MBIR method based on the CPU time in Table 3.1. The combined effect of surrogate optimization and SIMD parallelization provides a speedup of 15.1x and 37.6x for the reflectance and phase-error updates respectively, and consequently accelerates each DH-MBIR iteration by a factor of 23.7x.

⁷the timing includes complex-arithmetic computation of parameters m_s and ϕ_s in (3.13) which is intensive and not SIMD-compatible. Further using SIMD processing to compute (3.17) does not provide any significant benefit.

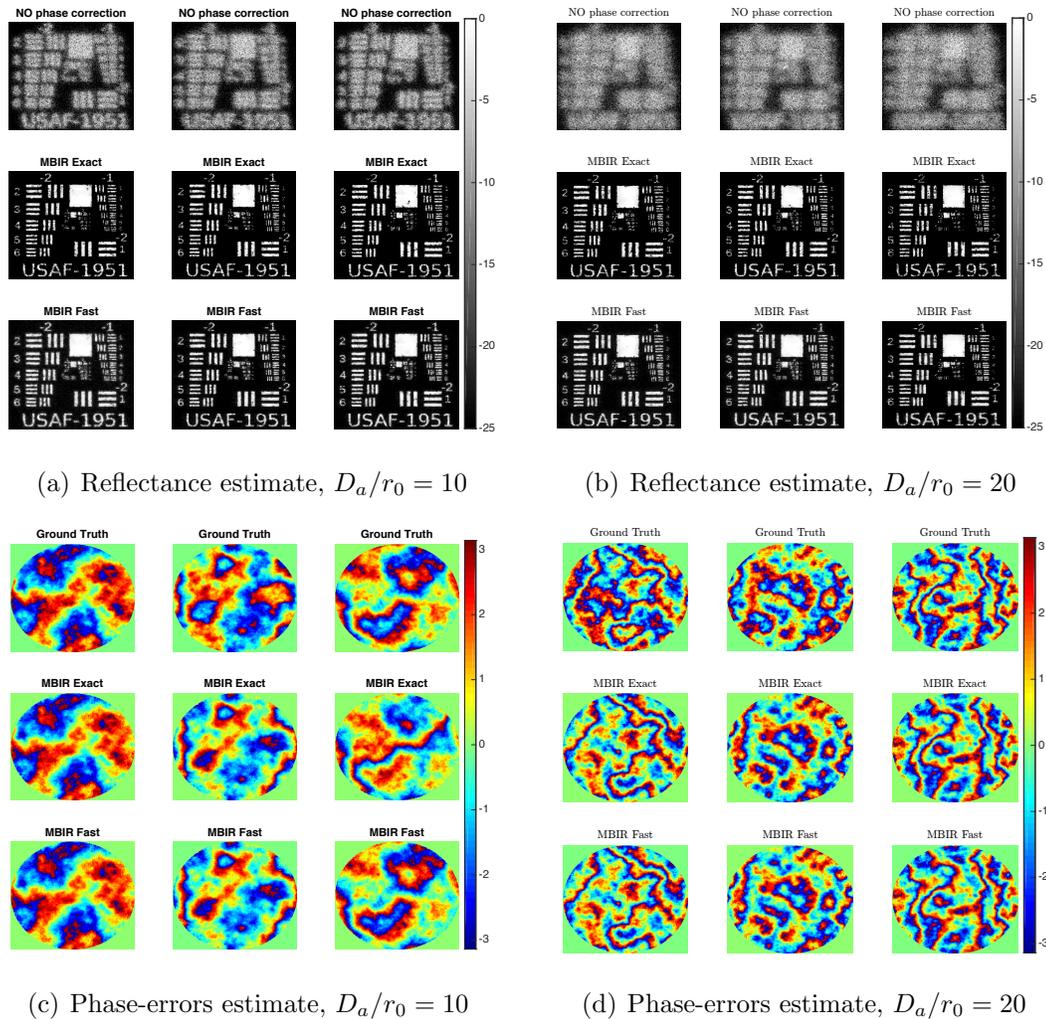


Fig. 3.6. Comparison of reconstruction quality for 6 different simulated datasets. For each dataset, the reflectance estimate without any phase recovery is shown in the top row of sub-figures (a) and (b) respectively, and the corresponding phase-errors in the pupil-plane are shown in the top row of sub-figures (c) and (d). In all sub-figures, the middle row shows reconstruction using the exact MBIR approach while the bottom row shows reconstruction using the fast MBIR approach. For most datasets, the reconstructions using the fast MBIR method are almost indistinguishable from the exact MBIR method.

Fig 3.5 shows the effect of fast updates on the convergence rate of DH-MBIR. Notice that replacing the exact updates for phase-error estimation with the fast updates

does not significantly alter the convergence rate. However, when we use fast updates for reflectance estimation in place of the exact updates, we require more iterations to converge.

Fig 3.6 compares the reconstruction quality of the fast and exact DH-MBIR approaches. We can see that for most datasets, the reflectance and phase-error estimates are almost indistinguishable. For the third dataset alone in Fig 3.6(c), the phase-error estimate from the fast DH-MBIR approach is more accurate as compared to the exact DH-MBIR method.

3.5 Conclusion

In this chapter of the thesis, we proposed a method that drastically reduces the computational cost DH-MBIR for real-time coherent imaging through turbulence. DH-MBIR jointly estimates the speckle-free object reflectance and atmospheric phase-errors from holographic sensor measurements. In order to speedup DH-MBIR, we first designed a surrogate function for the reflectance updates and a simple alternating minimization scheme for the wrapped phase-error updates. Further, we introduced a scheme that accelerates computation of the above surrogate updates using SIMD vector processing instructions. We demonstrated the effectiveness of our fast DH-MBIR method for real-time reconstruction with simulated datasets.

REFERENCES

REFERENCES

- [1] U. van Stevendaal, J. P. Schlomka, A. Harding, and M. Grass, “A reconstruction algorithm for coherent scatter computed tomography based on filtered back-projection,” *Medical Physics*, vol. 30, no. 9, pp. 2465–2474, September 2003.
- [2] G. Harding, H. Strecker, D. Kosciesza, and J. Gordon, “Detector considerations relevant to x-ray diffraction imaging for security screening applications,” in *Proc. SPIE, Optics and Photonics in Global Homeland Security and Biometric Technology for Human Identification*, vol. 7306, no. 19, 2009, pp. 1–11.
- [3] G. Harding, J. Kosanetzky, and U. Neitzel, “X-ray diffraction computed tomography,” *Medical Physics*, vol. 14, no. 4, pp. 515–525, July 1987.
- [4] V. Sridhar, S. J. Kisner, S. Skatter, and C. A. Bouman, “Model-based reconstruction for x-ray diffraction imaging,” *Proceedings of SPIE*, vol. 9847, pp. 98 470k–98 470k–11, 2016.
- [5] C. A. Bouman and K. Sauer, “A unified approach to statistical tomography using coordinate descent optimization,” *IEEE Trans. on Image Processing*, vol. 5, no. 3, pp. 480–492, March 1996.
- [6] K. Chen and D. A. Castonon, “A regularized iterative reconstruction algorithm for X-Ray Diffraction Tomography,” in *Proc. SPIE, Next-Generation Spectroscopic Technologies*, vol. 8374, 2012, pp. D1–D11.
- [7] P. Jin, C. A. Bouman, and K. D. Sauer, “A model-based image reconstruction algorithm with simultaneous beam hardening correction for X-Ray CT,” *IEEE Transactions on Computational Imaging*, vol. 1, no. 3, pp. 200–216, September 2015.
- [8] G. T. Herman, “Correction for beam hardening in computed tomography,” *Physics in Medicine and Biology*, vol. 24, no. 1, pp. 81–106, 1979.
- [9] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, “Algorithms and applications for approximate nonnegative matrix factorization,” *Computational Statistics & Data Analysis*, vol. 52, no. 1, pp. 155–173, 2007.
- [10] J.-B. Thibault, K. Sauer, C. Bouman, and J. Hsieh, “A three-dimensional statistical approach to improved image quality for multi-slice helical CT,” *Medical Physics*, vol. 34, no. 11, pp. 4526–4544, 2007.
- [11] K. Sauer and C. Bouman, “A local update strategy for iterative reconstruction from projections,” *IEEE Trans. on Signal Processing*, vol. 41, no. 2, pp. 534–548, February 1993.
- [12] J. Fessler, “Fundamentals of ct reconstruction in 2d and 3d,” *Comprehensive Biomedical Physics*, vol. 2, pp. 263–295, 2014.

- [13] H. Turbell, "Cone-beam reconstruction using filtered backprojection," *PhD Thesis, Linkopins University*, 2001.
- [14] M. Beister and W. Kolditz, D. amd Kalender, "Iterative reconstruction methods in x-ray ct," *Physica Medica*, pp. 94–108, 2012.
- [15] S. J. Kisner, E. Haneda, C. A. Bouman, S. Skatter, M. Kourinny, and S. Bedford, "Model-based ct reconstruction from sparse views," *International Conference on Image Formation in X-Ray Computed Tomography*, pp. 444–447, 2012.
- [16] D. M. B., S. Basu, J. B. Thibault, H. J., J. A. Fessler, C. A. Bouman, and K. Sauer, "A Study of four minimization approaches for iterative reconstruction in X-ray CT," *IEEE Nuclear Science Symposium Conference Record*, 2005.
- [17] C. A. Bouman and K. D. Sauer, "A unified approach to statistical tomography using coordinate descent optimization," *IEEE Transactions on Image Processing*, vol. 5, no. 3, pp. 480–492, 1996.
- [18] S. Sreehari, S. Venkatakrishnan, B. Wohlberg, G. T. Buzzard, L. F. Drummy, J. P. Simmons, and C. A. Bouman, "Plug-and-Play Priors for bright field electron tomography and sparse interpolation," *IEEE Transactions on Computational Imaging*, vol. 2, no. 4, pp. 408–423, 2016.
- [19] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," *IEEE Global Conference on Signal and Information Processing*, 2013.
- [20] X. Wang, A. Sabne, S. J. Kisner, A. Raghunathan, S. Midkiff, and C. A. Bouman, "High performance model based image reconstruction," *ACM PPOPP Conference*, 2016.
- [21] A. Sabne, X. Wang, S. J. Kisner, A. Raghunathan, S. Midkiff, and C. A. Bouman, "Model-based iterative CT image reconstruction on GPUs," *ACM PPOPP Conference*, 2016.
- [22] D. Matenine, G. Cote, J. Mascolo-Fortin, Y. Goussard, and P. Despres, "System matrix computation vs storage on GPU: A comparative study in cone beam ct," *Medical Physics*, vol. 45, no. 2, pp. 579–588, 2018.
- [23] G. T. Buzzard, S. Chan, S. Sreehari, and C. A. Bouman, "Plug-and-play unplugged: Optimization free reconstruction using consensus equilibrium," *SIAM Journal on Imaging Science*, 2018.
- [24] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising," *SIAM Journal of Imaging Sciences*, vol. 10, no. 4, pp. 1804–1844, 2017.
- [25] J. Zheng, S. Saquib, K. Sauer, and C. A. Bouman, "Parallelizable bayesian tomography algorithm with rapid, guarenteed convergence," *IEEE Transactions on Medical Imaging*, 2000.
- [26] J. Fessler and D. Kim, "Axial block coordinate descent (abcd) algorithm for x-ray ct image reconstruction," *11th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, 2011.

- [27] X. Wang, A. Sabne, P. Sakdhnagool, S. J. Kisner, C. A. Bouman, and S. P. Midkiff, “Massively parallel 3d image reconstruction,” *ACM SC Conference*, 2017.
- [28] W. Linyuan, A. Cai, H. Zhang, B. Yan, L. Li, G. Hu, and S. Bao, “Distributed ct image reconstruction algorithm based on the alternating direction method,” *Journal of X-ray Science and Technology*, pp. 83–98, 2014.
- [29] J. Cui, G. Pratz, B. Meng, and C. S. Levin, “Distributed MLEM: An iterative tomographic image reconstruction algorithm for Distributed Memory Architectures,” *IEEE Transactions on Medical Imaging*, vol. 32, no. 5, pp. 957–967, 2013.
- [30] K. Sauer and C. Bouman, “Bayesian estimation of transmission tomograms using segmentation-based optimization,” *IEEE Transactions on Nuclear Science*, pp. 1144–1151, 1992.
- [31] A. Chambolle, V. Caselles, M. Novaga, D. Cremers, and T. Pock, “An introduction to total variation for image analysis,” *HAL archives*, 2009.
- [32] V. Sridhar, X. Wang, G. T. Buzzard, and C. A. Bouman, “Distributed Iterative CT Reconstruction,” *arXiv:1911.09278*, 2019.
- [33] V. Sridhar, G. T. Buzzard, and C. A. Bouman, “Distributed Framework for Fast Iterative CT reconstruction from View-subsets,” *Proc. of Conference on Computational Imaging, IS&T Electronic Imaging*, 2018.
- [34] X. Wang, V. Sridhar, Z. Ronaghi, R. Thomas, J. Deslippe, D. Parkinson, G. T. Buzzard, S. P. Midkiff, C. A. Bouman, and S. K. Warfield, “Consensus equilibrium framework for super-resolution and extreme-scale ct reconstruction,” *The International Conference for High Performance Computing, Networking, Storage, and Analysis*, 2019.
- [35] C. A. Bouman and K. Sauer, “A generalized gaussian image model for edge-preserving map estimation,” *IEEE Transactions on Image Processing*, vol. 2, no. 3, pp. 296–310, 1993.
- [36] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-D transform-domain collaborative filtering,” *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [37] C. Y. M. D. Zhang K., Zuo W. and Z. L., “Beyond a Gaussian Denoiser: Residual Learning of Deep Cnn for image denoising,” *IEEE Transactions on Image Processing*, 2017.
- [38] K. Sauer and C. A. Bouman, “A local update strategy for iterative reconstruction from projections,” *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 534–548, 1993.
- [39] N. Parikh and S. Boyd, “Proximal algorithms,” *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2013.
- [40] S. Majee, T. Balke, C. A. Kemp, G. T. Buzzard, and C. A. Bouman, “4d x-ray ct reconstruction using multi-slice fusion,” *to appear in the proceedings of the International Conference on Computational Photography*, 2019.

- [41] C. B. Buades A. and M. J.M., “A non-local algorithm for image denoising,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.
- [42] V. Sridhar, G. T. Buzzard, and C. A. Bouman, “MPI Implementation of Distributed Memory Approach to CT Reconstruction,” 2018. [Online]. Available: <https://github.rcac.purdue.edu/vsridha?tab=repositories>
- [43] Z. Yu, J. Thibault, C. Bouman, K. Sauer, and J. Hsieh, “Fast model-based X-Ray CT reconstruction using spatially non-homogeneous ICD optimization,” *IEEE Trans. on Image Processing*, vol. 20, no. 1, pp. 161–175, 2011.
- [44] C. J. Pellizari, M. F. Spencer, and C. A. Bouman, “Phase-error estimation and image reconstruction from digital-holography data using a Bayesian framework,” *Journal of the Optical Society of America A*, vol. 34, no. 9, pp. 1659–1669, 2017.
- [45] C. J. Pellizari, M. Spencer, and C. A. Bouman, “Optically coherent image reconstruction in the presence of phase errors using advanced-prior models,” *Proceedings of SPIE Defense + Security Conference*, vol. 10650, 2018.
- [46] C. J. Pellizari, R. Trahan III, H. Zhou, S. Williams, S. E. Williams, B. Nemati, M. Shao, and C. A. Bouman, “Optically coherent image formation and denoising using a plug and play inversion framework,” *Adaptive Optics*, vol. 56, no. 16, 2017.
- [47] S. T. Thurman and J. R. Fienup, “Phase-error correction in digital holography,” *Journal of the Optical Society of America A*, vol. 25, no. 4, pp. 983–994, 2008.
- [48] S. Sulaiman, S. Gibson, and M. Spencer, “Predictive dynamic digital holography and image sharpening,” *Journal of the Optical Society of America A*, vol. 35, no. 6, pp. 923–935, 2018.
- [49] H. A. Ilhan, M. Dogar, and M. Ozcan, “Digital holographic microscopy and focusing methods based on image sharpness,” *Journal of Microscopy*, vol. 255, no. 3, pp. 138–149, 2014.
- [50] C. J. Pellizari, R. Trahan III, H. Zhou, S. Williams, S. E. Williams, B. Nemati, M. Shao, and C. A. Bouman, “Synthetic aperture ladar: A model-based approach,” *IEEE Transactions on Computational Imaging*, vol. 3, no. 4, pp. 901–916, 2017.
- [51] Z. Yu, J.-B. Thibault, K. Sauer, C. A. Bouman, and J. Hsieh, “Accelerated line search for coordinate descent optimization,” *IEEE Nuclear Science Symposium*, vol. 5, pp. 2841–2844, 2006.
- [52] J. Schmidt, “Numerical simulation of optical wave propagation with examples in matlab,” *SPIE*, 2010.
- [53] S. Srinath, L. A. Poyneer, A. R. Rudy, and S. M. Ammons, “Computationally efficient autoregressive method for generating phase screens with frozen flow and turbulence in optical simulations,” *Optical Society of America Optics Express*, vol. 23, no. 26, 2015.
- [54] C. A. Bouman, “Model based image processing,” 2013. [Online]. Available: <https://engineering.purdue.edu/~bouman/publications/pdf/MBIP-book.pdf>

- [55] H. H. Bauschke and P. L. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*, ser. CMS Books in mathematics. New York, Dordrecht, Heidelberg: Springer, 2011.

APPENDICES

A. MACE FORMULATION AS A FIXED-POINT PROBLEM

We show that the MACE equations of (2.10) can be formulated as a fixed-point problem represented by (2.11). For a more detailed explanation see Corollary 3 of [23].

Proof A simple calculation shows that for any $\mathbf{v} \in \mathbb{R}^{nN}$, operator \mathbf{G} defined in (2.9) follows

$$\mathbf{G}\mathbf{G}\mathbf{v} = \mathbf{G}\mathbf{v}, \text{ and so } (2\mathbf{G} - \mathbf{I})(2\mathbf{G} - \mathbf{I})\mathbf{v} = \mathbf{v}.$$

Thus $2\mathbf{G} - \mathbf{I}$ is self-inverse. We define \mathbf{w}^* as

$$\mathbf{w}^* = (2\mathbf{G} - \mathbf{I})\mathbf{v}^*,$$

in which case $\mathbf{v}^* = (2\mathbf{G} - \mathbf{I})\mathbf{w}^*$ due to the above self-inverse property. Additionally, (2.10) gives

$$(2\mathbf{F} - \mathbf{I})\mathbf{v}^* = (2\mathbf{G} - \mathbf{I})\mathbf{v}^*.$$

Note that the RHS of the above is merely \mathbf{w}^* . So, plugging \mathbf{v}^* in terms of \mathbf{w}^* on the LHS, we get

$$(2\mathbf{F} - \mathbf{I})(2\mathbf{G} - \mathbf{I})\mathbf{w}^* = \mathbf{w}^*.$$

Hence \mathbf{w}^* fixed-point of a specific map $\mathbf{T} : \mathbb{R}^{nN} \rightarrow \mathbb{R}^{nN}$, where $\mathbf{T} = (2\mathbf{F} - \mathbf{I})(2\mathbf{G} - \mathbf{I})$. Finding \mathbf{w}^* gives us \mathbf{v}^* , since $\mathbf{v}^* = (2\mathbf{G} - \mathbf{I})\mathbf{w}^*$. ■

B. CONVERGENCE OF PARTIAL-UPDATE MACE APPROACH

Lemma B.0.1 *Let $F_i : \mathbb{R}^n \rightarrow \mathbb{R}^n$, denote the proximal map of a strictly convex and continuously differentiable function $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$. Let $\varepsilon_s \in \mathbb{R}^n$ be defined as $\varepsilon_s = [0, \dots, 1, \dots, 0]^t$ where entry 1 is at s -th index. Let $\tilde{F}_i(v; x)$ denote the partial-update for $F_i(v)$ from an initial state x as shown in Algorithm 3. Then $F_i(v) = x$ if and only if $\tilde{F}_i(v; x) = x$.*

Proof We first assume $\tilde{F}_i(v; x) = x$. Since f_i is strictly convex and continuously differentiable, line 4 of Algorithm 3 can be re-written as

$$\alpha_s = \left\{ \alpha \mid \varepsilon_s^t \left[\nabla f_i(z + \alpha \varepsilon_s) + \frac{1}{\sigma^2}(z + \alpha \varepsilon_s - v) \right] = 0 \right\}. \quad (\text{B.1})$$

Since $\tilde{F}_i(v; x) = x$, from line 5 of Algorithm 3 and the fact that $\varepsilon_s, s = 1, \dots, n$ are independent, it follows that $\alpha_s = 0, s = 1, \dots, n$. Applying $\alpha_s = 0$ repeatedly to lines 4-5 of Algorithm 3 and using (B.1), we get

$$\frac{\partial f_i(x)}{\partial x_s} + \frac{1}{\sigma^2}(x_s - v_s) = 0, \quad s = 1, \dots, n.$$

Stacking the above result vertically, we get

$$\nabla f_i(x) + \frac{1}{\sigma^2}(x - v) = 0.$$

Since f_i is strictly convex and continuously differentiable the above gives

$$x = \underset{z}{\operatorname{argmin}} \left\{ f_i(z) + \frac{\|z - v\|^2}{2\sigma^2} \right\}$$

and so, $x = F_i(v)$. Therefore, $\tilde{F}_i(v; x) = x$ gives $F_i(v) = x$.

The converse can be proved by reversing the above steps. ■

Proof Proof of Theorem 2.2.1

Assume Partial-update MACE algorithm has a fixed-point $(\mathbf{w}^*, \mathbf{X}^*)$. Then from (2.13) we get,

$$\mathbf{X}^* = \tilde{\mathbf{F}}(\mathbf{v}^*; \mathbf{X}^*, \sigma) \text{ and} \quad (\text{B.2})$$

$$\mathbf{w}^* = 2\mathbf{X}^* - \mathbf{v}^*, \quad (\text{B.3})$$

where $\mathbf{v}^* = (2\mathbf{G} - \mathbf{I})\mathbf{w}^*$. So, (B.3) can be re-written as

$$\mathbf{w}^* = 2\mathbf{X}^* - (2\mathbf{G} - \mathbf{I})\mathbf{w}^*, \text{ which gives}$$

$$\mathbf{X}^* = \mathbf{G}\mathbf{w}^*.$$

So, $X_i^* = \bar{\mathbf{w}}^*$ for $i = 1, \dots, N$, and consequently, (B.2) can be expressed as

$$\bar{\mathbf{w}}^* = \tilde{F}_i(v_i^*; \bar{\mathbf{w}}^*, \sigma), \quad i = 1, \dots, N.$$

Applying Lemma B.0.1 to the above we get

$$\bar{\mathbf{w}}^* = F_i(v_i^*; \sigma), \quad i = 1, \dots, N.$$

By stacking the above result vertically, we get

$$\mathbf{G}\mathbf{w}^* = \mathbf{F}(\mathbf{v}^*).$$

Based on definition of \mathbf{v}^* , the above gives

$$\mathbf{G}\mathbf{w}^* = \mathbf{F}(2\mathbf{G} - \mathbf{I})\mathbf{w}^*.$$

Multiplying both LHS and RHS by 2 and further subtracting \mathbf{w}^* from both sides, we get

$$(2\mathbf{F} - \mathbf{I})(2\mathbf{G} - \mathbf{I})\mathbf{w}^* = \mathbf{w}^*.$$

Therefore, any fixed-point of the Partial-update MACE algorithm, $(\mathbf{w}^*, \mathbf{X}^*)$, is a solution to the exact MACE approach specified by (2.11). ■

Proof Proof of Theorem 2.2.2 (convergence of the Partial-update MACE algorithm)

We can express the function f_i defined in Theorem 2.2.2 more compactly. For this purpose, let subset J_i be represented as $J_i = \{k_1, k_2, \dots, k_M\}$. Then, f_i can be compactly written as

$$f_i(x) = \frac{1}{2} \|\tilde{y}_i - \tilde{A}_i x\|_{\tilde{\Lambda}_i}^2 + \frac{\beta}{N} x^T B x. \quad (\text{B.4})$$

where \tilde{y}_i , \tilde{A}_i and $\tilde{\Lambda}_i$ are defined as

$$\tilde{y}_i = \begin{bmatrix} y_{k_1} \\ \vdots \\ y_{k_M} \end{bmatrix}, \quad \tilde{A}_i = \begin{bmatrix} A_{k_1} \\ \vdots \\ A_{k_M} \end{bmatrix} \quad \text{and} \quad \tilde{\Lambda}_i = \begin{bmatrix} \Lambda_{k_1} & & \\ & \ddots & \\ & & \Lambda_{k_M} \end{bmatrix}.$$

From (B.4), we can express F_i , the proximal map of f_i , as

$$\begin{aligned} F_i(v) &= \operatorname{argmin}_{z \in \mathbb{R}^n} \left\{ f_i(z) + \frac{\|z - v\|^2}{2\sigma^2} \right\} \\ &= \operatorname{argmin}_{z \in \mathbb{R}^n} \left\{ \frac{1}{2} z^t \left(H_i + \frac{I}{\sigma^2} \right) z - z^t \left(b_i + \frac{v}{\sigma^2} \right) \right\}, \end{aligned} \quad (\text{B.5})$$

where $H_i \in \mathbb{R}^{n \times n}$ and $b_i \in \mathbb{R}^n$ are defined as

$$H_i = \tilde{A}_i^t \tilde{\Lambda}_i \tilde{A}_i + (\beta/N) B \quad \text{and} \quad b_i = \tilde{A}_i^t \tilde{y}_i.$$

We can obtain $\tilde{F}_i(v; x)$, the partial-update for $F_i(v)$ defined in (B.5), by using the convergence analysis of [38, 54] for ICD optimization. This gives

$$\tilde{F}_i(v; x) = -(L_i + D_i + \sigma^{-2} I)^{-1} (L_i^t x - b_i - \sigma^{-2} v), \quad (\text{B.6})$$

where matrices $L_i, D_i \in \mathbb{R}^{n \times n}$, are defined as

$L_i =$ Lower triangular sub-matrix of H_i (excluding diag.)

$D_i =$ Diagonal sub-matrix of H_i .

Further we define $\tilde{L}_i \in \mathbb{R}^{n \times n}$ by $\tilde{L}_i = L_i + D_i$. We can re-write equation (B.6) as

$$\begin{aligned} \tilde{F}_i(v; x) &= -(\tilde{L}_i + \sigma^{-2} I)^{-1} (L_i^t x - b_i - \sigma^{-2} v) \\ &= -\sigma^2 (I + \sigma^2 \tilde{L}_i)^{-1} (L_i^t x - b_i - \sigma^{-2} v) \end{aligned} \quad (\text{B.7})$$

Let $\varepsilon = \sigma^2$. For sufficiently small ε^2 , we can approximate $(I + \varepsilon\tilde{L}_i)^{-1}$ in (B.7) as

$$(I + \varepsilon\tilde{L}_i)^{-1} = I - \varepsilon\tilde{L}_i + \mathcal{O}(\varepsilon^2).$$

Plugging the above approximation into equation (B.7), simplifying, and dropping the $\mathcal{O}(\varepsilon^2)$ term, we get

$$\tilde{F}_i(v; x) = (I - \varepsilon\tilde{L}_i)v - \varepsilon L_i^t x + \varepsilon b_i \quad (\text{B.8})$$

and hence

$$2\tilde{F}_i(v; x) - v = (I - 2\varepsilon\tilde{L}_i)v - 2\varepsilon L_i^t x + 2\varepsilon b_i. \quad (\text{B.9})$$

Define block matrices $\mathbf{L} \in \mathbb{R}^{Nn \times Nn}$ and $\tilde{\mathbf{L}} \in \mathbb{R}^{Nn \times Nn}$ with L_i and \tilde{L}_i along the diagonal, respectively.

Using (B.8), (B.9), \mathbf{L} , and $\tilde{\mathbf{L}}$ in (2.13), we can express the Partial-update MACE update up to terms involving $\mathcal{O}(\varepsilon^2)$ as

$$\begin{aligned} \begin{bmatrix} \mathbf{w}^{(k+1)} \\ X^{(k+1)} \end{bmatrix} &= \begin{bmatrix} \rho & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} - 2\varepsilon\tilde{\mathbf{L}} & -2\varepsilon\mathbf{L}^t \\ \mathbf{I} - \varepsilon\tilde{\mathbf{L}} & -\varepsilon\mathbf{L}^t \end{bmatrix} \begin{bmatrix} \mathbf{v}^{(k)} \\ X^{(k)} \end{bmatrix} + \begin{bmatrix} 1 - \rho & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w}^{(k)} \\ X^{(k)} \end{bmatrix} + \mathbf{c} \\ &= \begin{bmatrix} \rho & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} - 2\varepsilon\tilde{\mathbf{L}} & -2\varepsilon\mathbf{L}^t \\ \mathbf{I} - \varepsilon\tilde{\mathbf{L}} & -\varepsilon\mathbf{L}^t \end{bmatrix} \begin{bmatrix} 2\mathbf{G} - \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{w}^{(k)} \\ X^{(k)} \end{bmatrix} + \begin{bmatrix} 1 - \rho & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w}^{(k)} \\ X^{(k)} \end{bmatrix} + \mathbf{c} \\ &= \begin{bmatrix} \rho(\mathbf{I} - 2\varepsilon\tilde{\mathbf{L}})(2\mathbf{G} - \mathbf{I}) + (1 - \rho)\mathbf{I} & -2\rho\varepsilon\mathbf{L}^t \\ (\mathbf{I} - \varepsilon\tilde{\mathbf{L}})(2\mathbf{G} - \mathbf{I}) & -\varepsilon\mathbf{L}^t \end{bmatrix} \begin{bmatrix} \mathbf{w}^{(k)} \\ X^{(k)} \end{bmatrix} + \mathbf{c}, \end{aligned} \quad (\text{B.10})$$

where $\mathbf{c} \in \mathbb{R}^{2nN}$ is a constant term based on variables ρ and b_i . Define $\mathbf{M}_\varepsilon \in \mathbb{R}^{2nN \times 2nN}$ and $\mathbf{z} \in \mathbb{R}^{2nN}$ as follows

$$\mathbf{M}_\varepsilon = \begin{bmatrix} \rho(\mathbf{I} - 2\varepsilon\tilde{\mathbf{L}})(2\mathbf{G} - \mathbf{I}) + (1 - \rho)\mathbf{I} & -2\rho\varepsilon\mathbf{L}^t \\ (\mathbf{I} - \varepsilon\tilde{\mathbf{L}})(2\mathbf{G} - \mathbf{I}) & -\varepsilon\mathbf{L}^t \end{bmatrix} \quad (\text{B.11})$$

$$\mathbf{z} = \begin{bmatrix} \mathbf{w} \\ X \end{bmatrix} \quad (\text{B.12})$$

Then we can re-write (B.10) as follows

$$\mathbf{z}^{(k+1)} = \mathbf{M}_\varepsilon \mathbf{z}^{(k)} + \mathbf{c} \quad (\text{B.13})$$

For $\mathbf{z}^{(k)}$ to converge in the limit $k \rightarrow \infty$, the absolute value of any eigenvalue of \mathbf{M}_ε must be in the range $(0, 1)$. We first determine the eigenvalues of \mathbf{M}_0 , where $\mathbf{M}_0 = \lim_{\varepsilon \rightarrow 0} \mathbf{M}_\varepsilon$, and then apply a 1st order approximation in ε to obtain the eigenvalues of \mathbf{M}_ε . We can express \mathbf{M}_0 as

$$\mathbf{M}_0 = \begin{bmatrix} 2\rho\mathbf{G} + (1 - 2\rho)\mathbf{I} & 0 \\ 2\mathbf{G} - \mathbf{I} & 0 \end{bmatrix}$$

Let $\lambda_0 \in \mathbb{R}$ and $\mathbf{z}_0 = [\mathbf{w}_0^t X_0^t]^t \in \mathbb{R}^{2nN}$ represent eigenvalue and eigenvector of \mathbf{M}_0 . Then $\mathbf{M}_0\mathbf{z}_0 = \lambda_0\mathbf{z}_0$, and so

$$\mathbf{G}\mathbf{w}_0 = \frac{1}{2\rho}(\lambda_0 + 2\rho - 1)\mathbf{w}_0 \quad (\text{B.14})$$

$$(2\mathbf{G} - \mathbf{I})\mathbf{w}_0 = \lambda_0 X_0 \quad (\text{B.15})$$

Since \mathbf{G} is an orthogonal projection onto a subspace, all of its eigenvalues are 0 or 1. This with (B.14) implies that $\lambda_0 + 2\rho - 1$ is 0 or 2ρ . In the first case, $\lambda_0 = 1 - 2\rho$, which lies in the open interval $(-1, 1)$ for ρ in $(0, 1)$. In the second case, $\lambda_0 = 1$. Applying this in (B.14) and (B.15), we get $X_0 = \mathbf{w}_0 = \mathbf{G}\mathbf{w}_0$, so that each eigenvector for $\lambda_0 = 1$ has $X_0 = \mathbf{w}_0$ with all subvectors identical.

Let $\lambda_\varepsilon \in \mathbb{R}$ and $\mathbf{z}_\varepsilon = [\mathbf{w}_\varepsilon^t X_\varepsilon^t]^t \in \mathbb{R}^{2nN}$ represent eigenvalue and eigenvector of \mathbf{M}_ε respectively. Let a be the derivative of λ_ε with respect to ε , and let $\mathbf{u}_1 = \nabla_\varepsilon \mathbf{w}_\varepsilon$ and $\mathbf{u}_2 = \nabla_\varepsilon X_\varepsilon$. Applying a 1st order approximation in ε , λ_ε and \mathbf{z}_ε are given by

$$\lambda_\varepsilon = \lambda_0 + a(\varepsilon - 0) = 1 + a\varepsilon$$

$$\mathbf{z}_\varepsilon = \begin{bmatrix} \mathbf{w}_0 + (\varepsilon - 0)\mathbf{u}_1 \\ X_0 + (\varepsilon - 0)\mathbf{u}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{w}_0 + \varepsilon\mathbf{u}_1 \\ \mathbf{w}_0 + \varepsilon\mathbf{u}_2 \end{bmatrix}.$$

If we can prove that a is negative when ε is infinitely small positive, then consequently $|\lambda_\varepsilon| < 1$, and so, the system of equations specified by equation (B.10) converges (note that the case of $\lambda_0 = 1 - 2\rho$ gives $|\lambda_\varepsilon| < 1$ by continuity for small ε). Since $\mathbf{M}_\varepsilon\mathbf{z}_\varepsilon = \lambda_\varepsilon\mathbf{z}_\varepsilon$, the first component of equation (B.11) gives

$$\begin{aligned} & \left[\rho(\mathbf{I} - 2\varepsilon\tilde{\mathbf{L}})(2\mathbf{G} - \mathbf{I}) + (1 - \rho)\mathbf{I} \right] (\mathbf{w}_0 + \varepsilon\mathbf{u}_1) \\ & - 2\rho\varepsilon\mathbf{L}^t(\mathbf{w}_0 + \varepsilon\mathbf{u}_2) = (1 + a\varepsilon)(\mathbf{w}_0 + \varepsilon\mathbf{u}_1). \end{aligned}$$

Neglecting terms $\mathcal{O}(\varepsilon^2)$, expanding, and using $(2\mathbf{G} - \mathbf{I})\mathbf{w}_0 = \mathbf{w}_0$, the above simplifies for $\varepsilon > 0$ to

$$2\rho(\mathbf{G} - \mathbf{I})\mathbf{u}_1 = \left[2\rho(\tilde{\mathbf{L}} + \mathbf{L}^t) + a\mathbf{I} \right] \mathbf{w}_0.$$

Applying \mathbf{G} to both sides, using $\mathbf{G}(\mathbf{G} - \mathbf{I}) = \mathbf{G}^2 - \mathbf{G} = 0$ and $\mathbf{G}\mathbf{w}_0 = \mathbf{w}_0$, we get

$$\mathbf{0} = 2\rho\mathbf{G}(\tilde{\mathbf{L}} + \mathbf{L}^t)\mathbf{w}_0 + a\mathbf{w}_0$$

and so,

$$\mathbf{G}(\tilde{\mathbf{L}} + \mathbf{L}^t)\mathbf{w}_0 = -\frac{a}{2\rho}\mathbf{w}_0. \quad (\text{B.16})$$

Since $\tilde{L}_i + L_i^t, i = 1, \dots, N$ is positive definite for each i , so is $\mathbf{H} = \tilde{\mathbf{L}} + \mathbf{L}^t$. Further, $\mathbf{G} \in \mathbb{R}^{nN \times nN}$ is an orthogonal projection matrix with n -dimensional range. Hence \mathbf{G} can be expressed as $\mathbf{G} = PP^t$, where $P \in \mathbb{R}^{nN \times n}$ is orthogonal basis of the range of \mathbf{G} (i.e $P^tP = I$). Since $\mathbf{w}_0 = \mathbf{G}\mathbf{w}_0$, equation (B.16) can be written as

$$PP^t\mathbf{H}PP^t\mathbf{w}_0 = -\frac{a}{2\rho}PP^t\mathbf{w}_0$$

Multiply both LHS and RHS by P^t , and define $\tilde{\mathbf{w}}_0 = P^t\mathbf{w}_0$. Since $P^tP = I$, we get

$$P^t\mathbf{H}P\tilde{\mathbf{w}}_0 = -\frac{a}{2\rho}\tilde{\mathbf{w}}_0$$

This implies that $-a/(2\rho)$ is an eigenvalue of $P^t\mathbf{H}P$. Since $\rho > 0$ and $P^t\mathbf{H}P$ is positive definite, we have $a < 0$, and consequently, $|\lambda_\varepsilon| < 1$.

Since all eigenvalues of \mathbf{M}_ε have absolute value less than 1, the system of equations specified by (2.13) converges to a point $\mathbf{z}^* = (\mathbf{w}^*, \mathbf{X}^*)$ in the limit $k \rightarrow \infty$. From Theorem 2.2.1, \mathbf{w}^* is a solution to the exact MACE approach specified by (2.11). ■

C. DERIVING THE MACE-PNP APPROACH

Theorem C.0.1 *Let $F_i, i = 1, \dots, N$, be the proximal map of a closed, convex, differentiable function, f_i , let $\sum_{i=1}^N f_i = f$, let F be the proximal map for f , and let H be any denoiser. Then, the MACE framework specified by equilibrium conditions (2.16) – (2.18) is exactly equivalent to the standard PnP framework specified by (2.14) and (2.15).*

Proof Assume (2.16) – (2.18) hold. Then, as per (2.16),

$$x^* = F_i(x^* + u_i^*; \sigma), i = 1, \dots, N.$$

Since f_i is convex, differentiable, and F_i is defined as

$$F_i(x; \sigma) = \operatorname{argmin}_{v \in \mathbb{R}^n} \left\{ f_i(v) + \frac{\|v - x\|^2}{2\sigma^2} \right\},$$

it follows from the above stated equilibrium condition that

$$\begin{aligned} \nabla f_i(x^*) + \frac{x^* - (x^* + u_i^*)}{\sigma^2} &= 0, \text{ or,} \\ \nabla f_i(x^*) - \frac{u_i^*}{\sigma^2} &= 0. \end{aligned} \tag{C.1}$$

Summing the above equation over $i = 1, \dots, N$ we get

$$\sum_{i=1}^N \nabla f_i(x^*) - \frac{N\bar{\mathbf{u}}^*}{\sigma^2} = 0,$$

where $\bar{\mathbf{u}} = \sum_{i=1}^N u_i/N$. Since $f = \sum_{i=1}^N f_i$, the above can be re-written as

$$\nabla f(x^*) + \frac{x^* - (x^* + N\bar{\mathbf{u}}^*)}{\sigma^2} = 0.$$

Since f is convex, the above equation implies that

$$x^* = \operatorname{argmin}_{v \in \mathbb{R}^n} \left\{ f(v) + \frac{\|v - (x^* + N\bar{\mathbf{u}}^*)\|^2}{2\sigma^2} \right\}, \text{ and so,}$$

$$x^* = F(x^* + N\bar{\mathbf{u}}^*; \sigma).$$

From (2.17) and (2.18), we additionally get $x^* = H(x^* - N\bar{\mathbf{u}}^*)$. Therefore, we get (2.14) and (2.15), where $\alpha^* = -N\bar{\mathbf{u}}^*$.

For the converse, assume (2.14) and (2.15) hold. Then, as per (2.14), $F(x^* - \alpha^*; \sigma) = x^*$. So, we get

$$\nabla f(x^*) + \frac{x^* - (x^* - \alpha^*)}{\sigma^2} = 0.$$

Since $f = \sum_{i=1}^N f_i$, we can re-write the above as

$$\alpha^* = - \sum_{i=1}^N \sigma^2 \nabla f_i(x^*).$$

We define u_i^* as $u_i^* = \sigma^2 \nabla f_i(x^*)$. So, from the above equation, we get $\alpha^* + \sum_{i=1}^N u_i^* = 0$, which gives (2.18). Further from the definition of u_i^* we get

$$\nabla f_i(x^*) + \frac{x^* - (x^* + u_i^*)}{\sigma^2} = 0, \text{ and so,}$$

$$F_i(x^* + u_i^*; \sigma) = x^*,$$

which gives (2.16). Also, as per (2.15), $H(x^* + \alpha^*) = x^*$, which gives (2.17). Therefore, we obtain (2.16) – (2.18), where $u_i^* = \sigma^2 \nabla f_i(x^*)$, $i = 1, \dots, N$. ■

Remark: As in [23], the theorem statement and proof can be modified to allow for nondifferentiable, but still convex functions, f_i .

Proof Proof of Theorem 2.3.1

Assume (2.16) – (2.18) hold. We define \mathbf{t}^* as $\mathbf{t}^* = N\bar{\mathbf{u}}^*$. So, (2.18) gives $\alpha^* + (\sum_{i=1}^N t_i^*)/N = 0$, or, $\alpha^* = -\bar{\mathbf{t}}^*$. Consequently, we can express (2.17) as

$$H(x^* - \bar{\mathbf{t}}^*) = x^*. \tag{C.2}$$

Further, (2.16) specifies $F_i(x^* + u_i^*; \sigma) = x^*$. We showed earlier in (C.1) that this gives $\nabla f_i(x) - u_i/\sigma^2 = 0$. So, we get

$$\begin{aligned} \nabla f_i(x^*) - \frac{Nu_i^*}{N\sigma^2} &= 0, \quad \text{or,} \\ \nabla f_i(x^*) + \frac{x^* - (x^* + t_i^*)}{(\sqrt{N}\sigma)^2} &= 0, \quad \text{or,} \\ F_i(x^* + t_i^*; \sqrt{N}\sigma) &= x^*, \quad i = 1, \dots, N. \end{aligned} \quad (\text{C.3})$$

Define \mathbf{w}^* as $\mathbf{w}^* = \hat{x}^* - \mathbf{t}^*$, where \hat{x}^* is N vertical copies of x^* . We write (C.2) as $\mathbf{G}_H \mathbf{w}^* = \hat{x}^*$. So, based on definition of \mathbf{w}^* , we have $\mathbf{t}^* = \hat{x}^* - \mathbf{w}^* = \mathbf{G}_H \mathbf{w}^* - \mathbf{w}^* = (\mathbf{G}_H - \mathbf{I})\mathbf{w}^*$. We can write (C.3) as $\mathbf{F}(\hat{x}^* + \mathbf{t}^*) = \hat{x}^*$ according to (2.19), and so, by plugging in \hat{x}^* and \mathbf{t}^* in terms of \mathbf{w}^* we get

$$\begin{aligned} \mathbf{F}(\mathbf{G}_H \mathbf{w}^* + (\mathbf{G}_H - \mathbf{I})\mathbf{w}^*) &= \mathbf{G}_H \mathbf{w}^*, \quad \text{or,} \\ \mathbf{F}(2\mathbf{G}_H - \mathbf{I})\mathbf{w}^* &= \mathbf{G}_H \mathbf{w}^*. \end{aligned}$$

Multiplying by 2 and adding \mathbf{w}^* on both sides we get

$$(2\mathbf{F} - \mathbf{I})(2\mathbf{G}_H - \mathbf{I})\mathbf{w}^* = \mathbf{w}^*.$$

Therefore, \mathbf{w}^* is a fixed-point of $\mathbf{T}_H = (2\mathbf{F} - \mathbf{I})(2\mathbf{G}_H - \mathbf{I})$, and, $\mathbf{G}_H(\mathbf{w}^*) = \hat{x}^*$, where \mathbf{w}^* is given by $\mathbf{w}^* = \hat{x}^* - N\mathbf{u}^*$.

For the converse, assume $(2\mathbf{F} - \mathbf{I})(2\mathbf{G}_H - \mathbf{I})\mathbf{w}^* = \mathbf{w}^*$ and $\mathbf{G}_H \mathbf{w}^* = \hat{x}^*$ hold. The former gives $\mathbf{F}(2\mathbf{G}_H - \mathbf{I})\mathbf{w}^* = \mathbf{G}_H \mathbf{w}^*$. Applying the latter, we get $\mathbf{F}(2\hat{x}^* - \mathbf{w}^*) = \hat{x}^*$. Define \mathbf{t}^* as $\mathbf{t}^* = \hat{x}^* - \mathbf{w}^*$. So, we have $\mathbf{F}(\hat{x}^* + \mathbf{t}^*) = \hat{x}^*$, or,

$$F_i(x^* + t_i^*; \sqrt{N}\sigma) = x^*, \quad i = 1, \dots, N.$$

A calculation with the definition of F_i shows that the above gives

$$F_i(x^* + t_i^*/N; \sigma) = x^*, \quad i = 1, \dots, N.$$

Define $\mathbf{u}^* = \mathbf{t}^*/N$. So, from the above, $F_i(x^* + u_i^*; \sigma) = x^*$, which gives (2.16). Since $\mathbf{G}_H \mathbf{w}^* = \hat{x}^*$, we have $x^* = H\bar{\mathbf{w}}^*$. Combining this with $\mathbf{w}^* = \hat{x}^* - \mathbf{t}^*$, we

get $x^* = H(x^* - \bar{\mathbf{t}}^*)$. Define $\alpha^* = -\bar{\mathbf{t}}^*$. So we have, $x^* = H(x^* + \alpha^*)$, which gives (2.17). Also from definition of α^* and \mathbf{u}^* , we get $\alpha^* + \sum_{i=1}^N u_i^* = 0$, which gives (2.18). Therefore, we obtain (2.16) to (2.18), where \mathbf{u}^* is given by $N\mathbf{u}^* = \hat{x}^* - \mathbf{w}^*$. ■

Proof Proof of Lemma 2.3.2

First we show that $2\mathbf{G}_H - \mathbf{I}$ is non-expansive when H is firmly non-expansive. This proof also applies to the case where H is a proximal map, since proximal maps are firmly non-expansive [39]. Consider any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{nN}$. Then

$$\begin{aligned} & \|(2\mathbf{G}_H - \mathbf{I})\mathbf{x} - (2\mathbf{G}_H - \mathbf{I})\mathbf{y}\|^2 \\ &= 4\|\mathbf{G}_H\mathbf{x} - \mathbf{G}_H\mathbf{y}\|^2 + \|\mathbf{x} - \mathbf{y}\|^2 - 4\langle \mathbf{G}_H\mathbf{x} - \mathbf{G}_H\mathbf{y}, \mathbf{x} - \mathbf{y} \rangle \end{aligned} \quad (\text{C.4})$$

By writing \mathbf{G}_H in terms of H , we simplify the last term as

$$\begin{aligned} & \langle \mathbf{G}_H\mathbf{x} - \mathbf{G}_H\mathbf{y}, \mathbf{x} - \mathbf{y} \rangle \\ &= \sum_{i=1}^N \langle H\bar{\mathbf{x}} - H\bar{\mathbf{y}}, x_i - y_i \rangle \\ &= \sum_{i=1}^N \langle H\bar{\mathbf{x}} - H\bar{\mathbf{y}}, \bar{\mathbf{x}} - \bar{\mathbf{y}} + (x_i - \bar{x}) - (y_i - \bar{y}) \rangle \\ &= N\langle H\bar{\mathbf{x}} - H\bar{\mathbf{y}}, \bar{\mathbf{x}} - \bar{\mathbf{y}} \rangle. \end{aligned}$$

Since H is *firmly non-expansive*, [55, Prop. 4.2] implies that $\langle H\bar{\mathbf{x}} - H\bar{\mathbf{y}}, \bar{\mathbf{x}} - \bar{\mathbf{y}} \rangle \geq \|H\bar{\mathbf{x}} - H\bar{\mathbf{y}}\|^2$. This gives

$$\langle \mathbf{G}_H\mathbf{x} - \mathbf{G}_H\mathbf{y}, \mathbf{x} - \mathbf{y} \rangle \geq N\|H\bar{\mathbf{x}} - H\bar{\mathbf{y}}\|^2 = \|\mathbf{G}_H\mathbf{x} - \mathbf{G}_H\mathbf{y}\|^2.$$

Plugging the above into the first equation of this proof, we get

$$\|(2\mathbf{G}_H - \mathbf{I})\mathbf{x} - (2\mathbf{G}_H - \mathbf{I})\mathbf{y}\|^2 \leq \|\mathbf{x} - \mathbf{y}\|^2.$$

Therefore, $(2\mathbf{G}_H - \mathbf{I})$ is a non-expansive map. Also, since $F_i, i = 1, \dots, N$ is the proximal map of a convex function, F_i is a resolvent operator, so $2F_i - I$ is a reflected resolvent operator, hence non-expansive. This means $2\mathbf{F} - \mathbf{I}$ is non-expansive, so $(2\mathbf{F} - \mathbf{I})(2\mathbf{G}_H - \mathbf{I})$ is non-expansive, since it is the composition of two non-expansive maps. ■

D. EM SURROGATE FUNCTION

To compute the Q-function in (3.6), we first need to compute $\log p(y, g|r, \phi)$ as well as the posterior distribution $p(g|y, r', \phi')$. The former is specified by

$$\begin{aligned} \log p(y, g|r, \phi) &= \log p(y|g, \phi) + \log p(g|r) \\ &= -\frac{1}{2\sigma_w^2} \|y - A_\phi g\|^2 - g^H \mathcal{D}(r)^{-1} g - \log |\mathcal{D}(r)| + \text{const.} \end{aligned} \quad (\text{D.1})$$

Similarly, we can obtain the posterior distribution as

$$\begin{aligned} p(g|y, r', \phi') &= p(y, g|r', \phi') / p(y|r', \phi') \\ &= \frac{1}{z} \exp \left\{ -\frac{1}{2\sigma_w^2} \|y - A_{\phi'} g\|^2 - g^H \mathcal{D}(r')^{-1} g \right\} \end{aligned}$$

where z is a normalizing constant. The above can be more compactly expressed as a complex Gaussian distribution

$$p(g|y, r', \phi') = \frac{1}{z} \exp \{ -(g - \mu)^H C^{-1} (g - \mu) \},$$

where mean μ and covariance matrix C are given by

$$\begin{aligned} C &= \left(\frac{1}{\sigma_w^2} A_{\phi'}^H A_{\phi'} + \mathcal{D}(r')^{-1} \right)^{-1} \\ \mu &= C A_{\phi'}^H g \end{aligned}$$

Consequently the posterior mean and variance are given by

$$E[g|r', \phi'] = \mu \quad \text{and} \quad E[gg^H|r', \phi'] = \mu\mu^H + C. \quad (\text{D.2})$$

From (D.1) and (D.2), we can show that

$$E[\log(y, g|r, \phi)|r', \phi'] = -\sum_{s=1}^n \left(\frac{|\mu_s|^2 + C_{s,s}}{r_s} + \log r_s \right) + \frac{2}{\sigma_w^2} \text{Real} (y^H \mathcal{D}(e^{j\phi}) A_0 \mu) + \text{const.}$$

The full derivation is available in [44], Appendix B.

VITA

VITA

Venkatesh Sridhar completed his Bachelors degree in Electronics & Instrumentation Engineering from the Birla Institute of Technology and Science, Pilani, India, in 2012 and his Masters degree in Electrical & Computer Engineering from Purdue University, West Lafayette, USA, in 2014. He worked at Intel Corporation, Oregon, USA as an Analog Engineer from 2014-2015. From August 2015 on-wards, he has been pursuing his PhD degree at Purdue University, under the supervision of Prof Charles Bouman and Prof Gregory Buzzard. His research interests include computational imaging, tomography, radar, inverse problems, parallel computing, and hardware acceleration of signal processing algorithms.