HIGH SPEED IMAGING VIA ADVANCED MODELING

by

Soumendu Majee

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

School of Electrical and Computer Engineering

West Lafayette, Indiana

August 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL STATEMENT OF COMMITTEE APPROVAL

Dr. Charles A. Bouman, Co-Chair

School of Electrical and Computer Engineering

Dr. Gregery T. Buzzard, Co-chair

Department of Mathematics

Dr. Stanley Chan

School of Electrical and Computer Engineering

Dr. Mary Comer

School of Electrical and Computer Engineering

Approved by:

Dr. Dimitrios Peroulis

Dedicated to my family.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude towards my Ph.D. advisors Prof. Charles Bouman and Prof. Gregery Buzzard, for their guidance and support. Their guidance has transformed how I approach a problem and present ideas to others, significantly improving my research skills. Their passion and high standards for research have inspired me to push myself throughout my Ph.D. journey.

I would like to thank Craig Kemp at Eli Lilly and Company for sharing his expertise and providing many of the experimental data used in my Ph.D. research. I would like to thank Doga Gursoy from Argonne National Laboratory for his guidance during my Ph.D. I would like to thank my fellow graduate student, Thilo Balke, for the productive collaborations throughout my Ph.D. I would like to thank my committee members Prof. Stanley Chan and Prof. Mary Comer, for their valuable feedback and thoughtful discussions. I would also like to thank my former labmates Dilshan, Zeeshan, Aditya, Suhas, Shruthi, Xiao, Jing, Venkatesh, Camillo, and my current labmates Thilo, Maliha, Emma, Wenrui, Madhuri, Diyu for all the stimulating discussions and support throughout the years.

I am thankful to my partner and best friend, Mouli, for helping me overcome the hurdles of my Ph.D. journey and helping me see the light during the darkest of times. I am grateful to my parents, Dr. Swapan Majee and Dr. Niyati Majee, for nurturing my curiosity from a young age and their continuous support for my education. Their countless sacrifices, hard work, and support have enabled me to become the person I am today.

TABLE OF CONTENTS

LIST OF TABLES . . . 8

LIST OF FIGURES . . . 9

ABBREVIATIONS . . . 14

ABSTRACT . . . 15

1 A MODEL BASED NEURON DETECTION APPROACH USING SPARSE LOCATION PRIORS . . . 17

1.1 Introduction . . . 17

1.2 Forward and Prior Model . . . 19

1.3 Neuron Centers Estimation . . . 22

1.4 Shape Model Estimation . . . 23

1.5 Experimental Results . . . 26

1.5.1 Dataset Description . . . 26

1.5.2 Training . . . 27

1.5.3 Testing and Evaluation . . . 27

1.6 Conclusion . . . 31

2 MULTI-SLICE FUSION FOR SPARSE-VIEW AND LIMITED-ANGLE 4D CT RECONSTRUCTION . . . 34

2.1 Introduction . . . 34

2.2 Problem Formulation . . . 36

2.3 MACE Model Fusion 38

2.4 Multi-Slice Fusion using MACE 43

2.5 Training of CNN Denoisers 45

2.6 Distributed Reconstruction 46

2.7 Experimental Results 48

2.7.1 Simulated Data 360° 50

2.7.2 Simulated Data 90° 53

2.7.3 Real Data 360°: Vial Compression 54

2.7.4 Real Data 90°: Injector Pen 56

2.8 Conclusion 57

3 CODEX: A MODULAR FRAMEWORK FOR JOINT TEMPORAL DE-BLURRING AND TOMOGRAPHIC RECONSTRUCTION 64

3.1 Introduction 64

3.2 Problem Description 66

3.3 CodEx Formulation 68

3.3.1 Measurement Model 69

3.3.2 MAP Estimate 70

3.3.3 ADMM Formulation 71

3.3.4 Modular Implementation 73

3.4 Interlaced View Sampling 76

3.5 Results 80

3.5.1 Simulated data without noise 82

3.5.2 Simulated data with photon noise 83

3.5.3 Binned physical data 88

3.6 Conclusion 90

REFERENCES 91

A CORRECTION FOR FIXTURES OUTSIDE THE FIELD OF VIEW 99

B UNIQUENESS PROOF OF INTERLACED VIEW-ANGLES 101

VITA 102

LIST OF TABLES

2.1 Total reconstruction time of multi-slice fusion for each experimental case 50

2.2 Experimental specifications for Simulated Data 360° 50

2.3 Quantitative Evaluation for simulated data 360°. Multi-slice fusion has the highest PSNR and SSIM metric among all the methods. 52

2.4 Experimental specifications for Simulated Data 90° 53

2.5 Quantitative Evaluation for simulated data 90°. Multi-slice fusion has the highest PSNR and SSIM metric among all the methods. 53

2.6 Experimental specifications for Real Data 360°: Vial Compression 54

2.7 Experimental specifications for Real Data 90°: Injector Pen 56

3.1 Typical parameter choices for view-angles 78

3.2 Experimental setup for simulated data without noise 83

3.3 Experimental setup for simulated data with noise 84

3.4 Experimental setup for checking robustness with respect to the underlying object to be reconstructed. 87

3.5 Experimental setup for binned physical data 89

LIST OF FIGURES

1.1 Example of a GCaMP6 calcium fluorescence image used in this study. The bright circle is a neuron affected with GCaMP over-expression. The large black circles are blood vessels. The annular rings are normal neurons. . . . 19

1.2 Illustration of how $A^{(1)}X^{(1)} + A^{(2)}X^{(2)}$ model the neurons in the image. The ring shapes represent normal neurons and the disk shaped ones represent over-expressed neurons 20

1.3 Illustration of all the variables in the forward model and their sizes 21

1.4 Overview of neuron center estimation in MBND 24

1.5 Overview of the process of extracting training patches and estimating the shape models 25

1.6 The equatorial slices of the eigen-images for the (a) normal neuron data (b) neurons affected by GCaMP over-expression. Only the eigen-images corresponding to the highest eigenvalues are shown. 28

1.7 Eigenvalues of the (a) normal neuron data (b) neurons affected by GCaMP over-expression 29

1.8 Slice by slice comparison of detected neurons by our proposed method(MBND) with CellSegm Here (a), (e), (i), (m) show the original test image in slices 3,8,13,18 respectively. (b), (f), (j), (n) show the ground truth in slices 3,8,13,18 respectively. (c), (g), (k), (o) show the neurons detected by CellSegm in slices 3,8,13,18 respectively. (d), (h), (l), (p) show the neurons detected by our MBND in slices 3,8,13,18 respectively. To show the neurons, we draw a sphere around the neuron centers in 3D and we show the circular intersection of the surface of the sphere with the current slice in the figures. A dot in the center of the circle indicates that the neuron center is in the current slice. Ground truth and true detection are shown in green, false positives in red and missed detection in blue. 32

1.9 Precison recall plot of our proposed method(MBND) compared with CellSegm 33

2.1 Illustration of our multi-slice fusion approach. Each CNN denoiser operates along the time direction and two spatial directions. We fuse the CNN denoisers with the measurement model to produce a 4D regularized reconstruction. . . 35

2.2 Illustration of 4D cone-beam X-ray CT imaging. The dynamic object is rotated and several 2D projections (radiographs) of the object are measured for different angles. The projections are divided into N_t disjoint subsets for each of the N_t time-points. 37

2.3 Illustration of consensus equilibrium as analogous to a force balance equation: each agent pulls the solution toward its manifold and at equilibrium the forces balance each other. 40

2.4 Architecture of our 2.5D CNN denoiser. Different sizes of input and output necessitate a selection operator for the residual connection. Each green rectangle denotes a tensor, and each ellipse denotes an operation. Blue ellipses specify the shape of the convolution kernel. 44

2.5 Illustration of our training data generation. We extract 3D patches from a typical CT volume and add additive white Gaussian noise (AWGN) to generate training pairs. This makes the training process self-supervised. 46

2.6 Illustration of distributed computation of multi-slice fusion. We perform distributed computation of the F operator which is the main computational bottleneck in Algorithm 1. Each operator within F, namely $H_{\mathrm{xy,t}}$, $H_{\mathrm{yz,t}}$, $H_{\mathrm{zx,t}}$, and L can be executed in parallel. Furthermore, operators $H_{\mathrm{xy,t}}$, $H_{\mathrm{yz,t}}$, $H_{\mathrm{zx,t}}$, and L are 3D operators that can process the 4D volume "slice by slice" leading to a large number of concurrent operations that can be distributed among multiple compute nodes. 47

2.7 Comparison of different methods for simulated data 360°. Each image is a slice through the reconstructed object for one time-point along the spatial xy-plane. The reconstruction using FBP suffers from high noise and fails to recover the small hole in the bottom of the image. MBIR+TV and MBIR+4D-MRF suffer from jagged edges and fail to recover the small hole in the bottom of the image. MBIR+$H_{\mathrm{yz,t}}$ and MBIR+$H_{\mathrm{zx,t}}$ suffer from horizontal and vertical streaks, respectively, since the denoisers were applied in those planes. MBIR+$H_{\mathrm{xy,t}}$ cannot reconstruct the small hole in the bottom of the image since the xy-plane does not contain sufficient information. 48

2.8 Plot of cross-section through the phantom and reconstructions from simulated data 360°. Multi-slice fusion results in the most accurate reconstruction of the gap between materials. 51

2.9 Comparison of different methods for simulated data with 90° rotation of object per time-point. The FBP reconstruction has severe limited-angle artifacts. MBIR+TV improves the reconstruction in some regions but it suffers in areas affected by limited angular information. MBIR+4D-MRF reduces limited-angle artifacts, but allows severe artifacts to form that are not necessarily consistent with real 4D image sequences. In contrast, the multi-slice fusion result does not suffer from major limited-angle artifacts. 58

2.10 Illustration of the reconstruction quality obtained for extreme sparse-view data with different levels of limited angle per time-point. FBP results in strong artifacts due to sparse-views and limited angles. MBIR+TV and MBIR+4D-MRF mitigates most of the major sparse-view artifacts but suffers from limited angle artifacts in the 90° limited angle case. Multi-slice fusion results in fewer limited-angle and sparse-view artifacts and an improved PSNR metric. Moreover, multi-slice fusion results in reduced artifacts compared to MBIR+TV and MBIR+4D-MRF as the rotation per time point is decreased. 59

2.11 Comparison of different methods for Real Data 360°: vial. Each image is a slice through the reconstructed vial for one time-point along the spatial xy-plane. Both FBP and MBIR+4D-MRF suffer from obvious windmill artifacts, higher noise and blurred edges. In contrast to that, the multi-slice fusion reconstruction has smooth and uniform textures while preserving edge definition. MBIR+$H_{\mathrm{yz,t}}$ and MBIR+$H_{\mathrm{zx,t}}$ suffer from horizontal and vertical streaks. MBIR+$H_{\mathrm{xy,t}}$ cannot reconstruct the outer ring since the slice displayed is at the edge of the aluminum seal and the xy-plane does not contain sufficient information. Multi-slice fusion can resolve the edges of the rings better than either of MBIR+$H_{\mathrm{xy,t}}$, MBIR+$H_{\mathrm{yz,t}}$, and MBIR+$H_{\mathrm{zx,t}}$ since it has information from all the spatial coordinates. 60

2.12 Plot of cross-section through the vial at a time when the aluminum and glass have physically separated. Multi-slice fusion is able to resolve the junction between materials better while simultaneously producing a smoother reconstruction within materials compared to MBIR+4D-MRF and FBP. 61

2.13 Illustration of temporal resolution for real data 360° : vial. We plot a cross-section through the vial with time for each method: multi-slice fusion, MBIR+4D-MRF, FBP. Multi-slice fusion results in improved space-time resolution of the separation of aluminum and glass. 62

2.14 Volume rendering of the reconstructed spring and its cross-section for four time-points. A 90° limited set of views is used to reconstruct each time-point. The FBP reconstruction contains severe limited-angle artifacts. MBIR+4D-MRF mitigates some limited-angle artifacts but some artifacts remain. . . . 63

2.15 Experimental setup for Real Data 360°: Vial Compression. The vial is undergoing dynamic compression during the scan, to capture the mechanical response of the components. The glass vial (center) and the actuator (top) are held together by a frame constructed of tubes and plates. The tubes were placed outside the field of view of the CT scanner, thus causing artifacts in the reconstruction. We describe a correction for this in Appendix A. 63

3.1 Illustration of our method, CodEx. CodEx is a synergistic combination of a coded acquisition and a non-convex Bayesian reconstruction. During acquisition, CodEx flutters the exposure on and off rapidly in a known binary code during each view, resulting in a coded motion blur. CodEx subsequently uses the knowledge of the code to solve a non-convex iterative reconstruction problem to reconstruct the underlying object. 65

3.2 Illustration of motion blur due to rotation. In step-and-shoot acquisition, each measurement depends on the projection at a single angle (blue line). In fly-scanning, the detector integrates photons over a range of projection angles (red sector). This integration can be approximated as a discrete sum over micro-projection angles (red dots). 67

3.3 Illustration of our View angle sampling for coded exposure CT for $N_\theta = mK - n = 50$ ($K = 11$, $m = 5$, $n = 5$) and $M_\theta = 50$. Each new measurement angle shown by a blue dot is spaced by $K = 11$ discrete angular steps, but the measurement angles do not repeat until all $N_\theta = 50$ distinct micro-projection angles (blue dots) are sampled. 79

3.4 Examples of the binary codes of length 52 that are used for modulating the photon-flux in our experiments. 81

3.5 Comparison of reconstruction quality for simulated data without noise. Experimental details are in Table 3.2. For the Raskar and boxcar codes, the baseline reconstruction produces a blurred image. In contrast, CodEx is able to reconstruct the image without suffering from severe blurring. The CodEx-boxcar reconstruction has noticeable radial blur due to the non-invertible nature of the boxcar blur kernel. In contrast, the CodEx-Raskar and CodEx-snapshot reconstructions do not suffer from severe radial blur artifacts. 82

3.6 Comparison of reconstruction quality for simulated data with photon noise. Experimental details are in Table 3.3. For Raskar and boxcar codes, the baseline reconstruction leads to a blurred image. In contrast, CodEx-Raskar and CodEx-boxcar reconstructions do not suffer from severe blurring artifacts. The reconstructions with snapshot code suffer from high noise due to the limited photon counts in the measurements. The reconstructions with boxcar code suffer from radial blur artifacts and loss of fine features due to the non-invertible nature of the blur kernel. 84

3.7 Primal and dual residual convergence plots for simulated data with photon noise. Here t refers to the ADMM iteration number. The Raskar code leads to an improved convergence than the boxcar code. 85

3.8 Effect of the photon-flux (λ^0) and the code length on reconstruction quality for different codes. The remaining experimental parameters are kept the same as Table 3.3. When the photon-flux (λ^0) is low, the boxcar code and Raskar code produce better image quality than the snapshot code as a result of collecting more photons. When the photon-flux (λ^0) is high, the minor gains from increased photon count do not fully compensate for the loss of information by inverting the coded blur. Consequently at high photon-flux, the snapshot code produce better image quality than the boxcar code and Raskar codes. 86

3.9 Robustness with respect to underlying reconstructed object. Experimental details are summarized in Table 3.4. The CodEx reconstruction is able to recover the major features of the image despite the high angular blur. In contrast, the baseline results in overtly blurred reconstructions. 87

3.10 Comparison of reconstruction quality for binned physical data. Experimental details are summarized in Table 3.5. CodEx reconstructions preserve the major features in the image whereas the baseline leads to overtly blurred reconstructions. 88

3.11 Primal and dual residual convergence plots for binned physical data. Here t refers to the ADMM iteration number. The Raskar code leads to an improved convergence than the boxcar code. 89

A.1 Pipeline of the blind fixture correction in Algorithm 6. The vertical stripes in the yz-plane of the reconstruction and the ring at the edge of the field of view in the xy-plane of the reconstruction have been rectified after performing the correction. 100

List of Abbreviations is optional.

ABBREVIATIONS

MBIR	Model based iterative reconstruction
CT	Computed tomography
MAP	Maximum a posteriori
DCT	Discrete cosine transform
CNN	Convolutional neural network
PnP	Plug and play
MACE	Multi agent consensus equilibrium
BM4D	Block matching 4D
FBP	Filtered back projection
MRF	Markov random field
TV	Total variation
ICD	Iterative coordinate descent
AWGN	Additive white Gaussian noise
RMSE	Root mean squared error
MSF	Multi-slice fusion
SNR	Signal to noise ratio
ADMM	Alternating direction method of multipliers

ABSTRACT

There is an increasing need to accurately image objects at a high temporal resolution for different applications in order to analyze the underlying physical, chemical, or biological processes. In this thesis, we use advanced models exploiting the image structure and the measurement process in order to achieve an improved temporal resolution. The thesis is divided into three chapters, each corresponding to a different imaging application.

In the first chapter, we propose a novel method to localize neurons in fluorescence microscopy images. Accurate localization of neurons enables us to scan only the neuron locations instead of the full brain volume and thus improve the temporal resolution of neuron activity monitoring. We formulate the neuron localization problem as an inverse problem where we reconstruct an image that encodes the location of the neuron centers. The sparsity of the neuron centers serves as a prior model, while the forward model comprises of shape models estimated from training data.

In the second chapter, we introduce multi-slice fusion, a novel framework to incorporate advanced prior models for inverse problems spanning many dimensions such as 4D computed tomography (CT) reconstruction. State of the art 4D reconstruction methods use model based iterative reconstruction (MBIR), but it depends critically on the quality of the prior modeling. Incorporating deep convolutional neural networks (CNNs) in the 4D reconstruction problem is difficult due to computational difficulties and lack of high-dimensional training data. Multi-Slice Fusion integrates the tomographic forward model with multiple low dimensional CNN denoisers along different planes to produce a 4D regularized reconstruction. The improved regularization in multi-slice fusion allows each time-frame to be reconstructed from fewer measurements, resulting in an improved temporal resolution in the reconstruction. Experimental results on sparse-view and limited-angle CT data demonstrate that Multi-Slice Fusion can substantially improve the quality of reconstructions relative to traditional methods, while also being practical to implement and train.

In the final chapter, we introduce CodEx, a synergistic combination of coded acquisition and a non-convex Bayesian reconstruction for improving acquisition speed in computed tomography (CT). In an ideal "step-and-shoot" tomographic acquisition, the object is rotated

to each desired angle, and the view is taken. However, step-and-shoot acquisition is slow and can waste photons, so in practice the object typically rotates continuously in time, leading to views that are blurry. This blur can then result in reconstructions with severe motion artifacts. CodEx works by encoding the acquisition with a known binary code that the reconstruction algorithm then inverts. The CodEx reconstruction method uses the alternating direction method of multipliers (ADMM) to split the inverse problem into iterative deblurring and reconstruction sub-problems, making reconstruction practical. CodEx allows for a fast data acquisition leading to a good temporal resolution in the reconstruction.

1. A MODEL BASED NEURON DETECTION APPROACH USING SPARSE LOCATION PRIORS

1.1 Introduction

Mapping the functional connectivity in the brain continues to be a major challenge in neuroscience. An important component in developing such a map is the ability to measure the activity of many neurons accurately at a high frame rate. Fluorescence microscope imaging using recently developed fluorescent calcium indicators is a good candidate for making these measurements due to a combination of high spatial and temporal resolution. The foundation of this form of neural imaging is the increase in calcium ion concentration whenever a neuron is activated [1]; this increased concentration leads to fluorescence from the calcium indicator, which is captured by a fluorescence microscope [2] [3]. Recently developed calcium indicators like GCaMP6 respond to changes in calcium ion concentration quickly and thus provide the possibility of measuring neural activity at a high frame rate [4].

In conventional raster scan fluorescence microscopy, in which all the voxels in the volume are scanned equally often, the frame rate suffers markedly as the size of the brain volume to be imaged increases [5] [6]. However, for purposes of capturing neural activity, it is necessary to scan only the volume containing the neurons, which is typically a small fraction of the total volume. Thus, replacing a full raster scan for each frame with a scan of the much smaller neural volume can dramatically increase the scan frame rate. Of course, this strategy relies on accurate knowledge of the neural volume, which we propose to obtain using a single GCaMP6 scan of the full volume of interest.

The central challenge of this research is to create a method for accurately detecting and determining the location of each neuron in a GCaMP6 image so that the volume can be sparsely scanned at a high frame rate. This goal presents a number of difficult challenges. First, there can be large illumination variations across the image, along with high noise. Furthermore, there is often fluorescence from the dense network of axons and dendrites, referred to as neuropil. There are also cylindrical blood vessels in the images that have similarities with neurons, making the detection of neurons more challenging. Also, some neurons are saturated with fluorescence; this is an artifact of GCaMP overexpression that leads to a significantly different type of calcium fluorescent image relative to other neurons [1] [3].

Before describing our method, we briefly review related work on cell segmentation and tracking methods, including frame to frame association and active contour models.

Several algorithms [7] [8] track cells in a sequence of 2D images by detecting cells in each frame and then establishing frame to frame association based on several criteria. However, there are problems in the frame to frame association phase if there is large variation in illumination from frame to frame, a high density of cells, and/or a large number of cells entering or leaving the frame [9] [10].

Active contour models have also been proposed for segmenting and tracking cells in microscopy images [11] [12] [13] [14] [15]. Active contour based methods take a deformable contour with a certain energy function and minimize that energy to segment a cell. However a good initialization is often required to avoid getting stuck on a local minima [16]. Active contours require a data attachment term to guide the contour. Many methods use a data attachment term that depends solely on the edge map computed from the image gradient [17] [18] [19]. However, in images with high noise, gradient information is unreliable. Other methods use statistical intensity information computed for the region inside the contour [20] [21].

Active contour models have been extended to include strong prior shape information in [22] [23] [24] [25]. However it is difficult to detect an unknown number of objects [26]. To mitigate this, variations of active contour with Marked Point process is proposed in [26] [27]. But these methods are formulated for 2D images and cannot be directly applied to 3D images.

In this paper, we solve the problem of detecting neuron centers in a 3D volume using a novel model-based neuron detection algorithm (MBND). We formulate this as an image reconstruction problem in which we construct a sparse location image that encodes the neuron centers. That is, a non-zero value in the location image corresponds to the center of a neuron. We model the observed neurons as a linear combination of shape models in the forward model and use separate shape models for normal neurons and abnormal over-expressed neurons; the coefficients in this linear combination are the nonzero entries in the

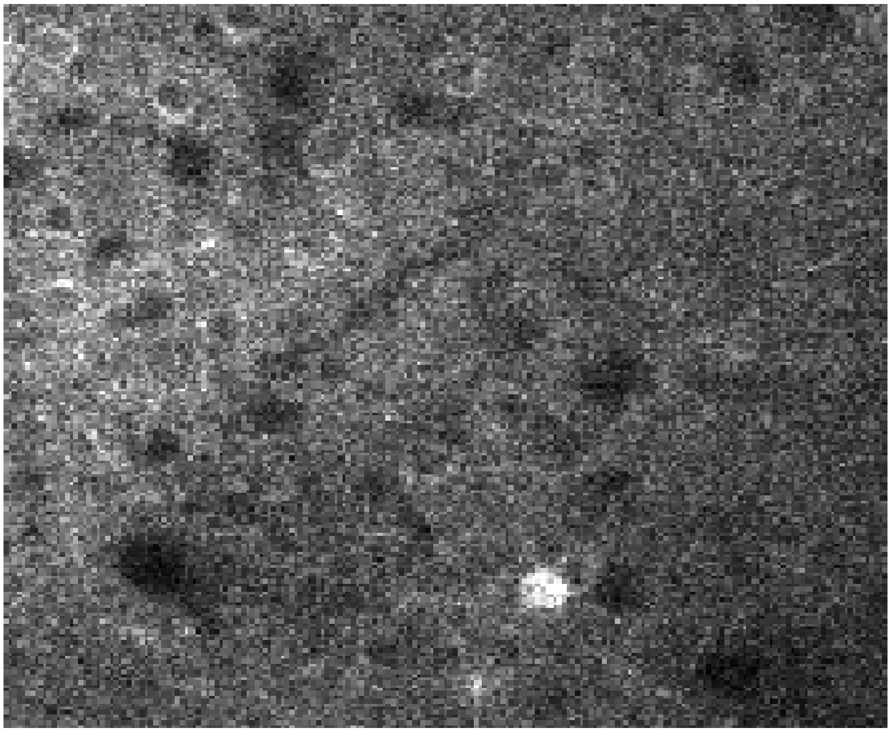


Figure 1.1. Example of a GCaMP6 calcium fluorescence image used in this study. The bright circle is a neuron affected with GCaMP over-expression. The large black circles are blood vessels. The annular rings are normal neurons.

location image. We use a sparsity based prior model for the construction of the location image since the neuron centers are sparsely distributed in the 3D volume. We model the background illumination variation in the image as a linear combination of low-frequency basis vectors of the Discrete Cosine Transform, and we model the presence of sparsely distributed bright dendrites as additive impulsive noise. We compute the MAP estimate of the location image by minimizing a convex cost function. Tests on GCaMP6 fluorescence neuron images show better accuracy than widely used methods.

1.2 Forward and Prior Model

To formulate the problem, let us denote the observed vectorized image as $Y \in \mathbb{R}^N$, where N is the total number of voxels in the 3D image. Let S denote the lattice in $\mathbb{R}^3$ representing the voxel locations in the image and let Ω map from the lattice points in S to an index i such that $1 \leq i \leq N$. Then $Y_{\Omega(s)}$ is the value of the image at the voxel location $s \in S$

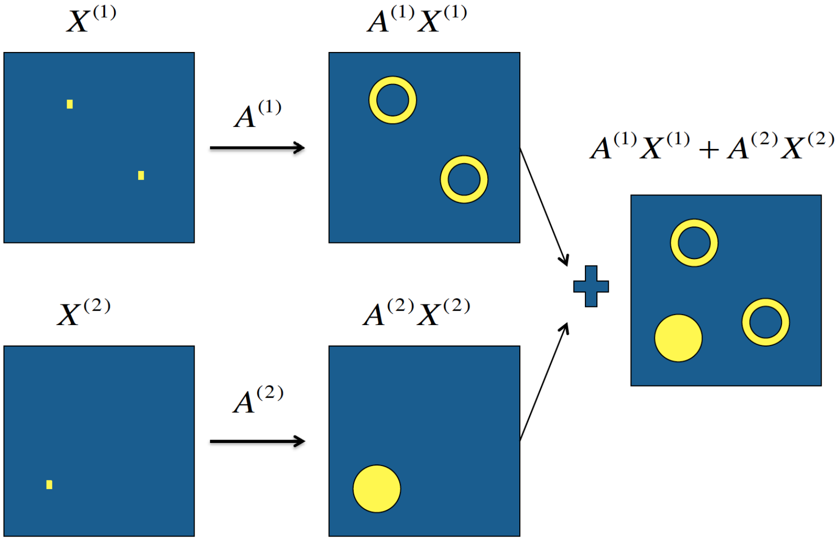


Figure 1.2. Illustration of how $A^{(1)}X^{(1)}+A^{(2)}X^{(2)}$ model the neurons in the image. The ring shapes represent normal neurons and the disk shaped ones represent over-expressed neurons

Let $A^{(1)}, A^{(2)} \in \mathbb{R}^{N\times N}$ be matrices corresponding to linear space invariant filters. The impulse responses of $A^{(1)}, A^{(2)}$ are the shape model of normal and over-expressed neurons. $A^{(1)}, A^{(2)}$ perform convolution with the shape model as their convolution kernel.

Let $X^{(1)}, X^{(2)} \in \mathbb{R}^{N}$ be sparse images which encode the location of the normal and over-expressed neuron centers respectively.We shall refer to them as location images. $X^{(1)}$ and $X^{(2)}$ represent the contribution of two neuron models to explain the data at each voxel location. $X^{(k)}$ is non-zero only at the center of neurons that are modeled by the shape model of $A^{(k)}$ and the non-zero value is proportional to the brightness of the neuron it models.

Therefore, $A^{(1)}X^{(1)}+A^{(2)}X^{(2)}$ models the neurons in the image Y. This is illustrated in figure 1.2.

Let the columns of the matrix $B \in \mathbb{R}^{N\times M}$ be the M basis vectors of the 3D Discrete Cosine Transform corresponding to the low-frequency components. Then given appropriate choice of the parameter vector $\theta \in \mathbb{R}^{M}$, $B\theta$ can be used to model the low-frequency background illumination variation in the image for an appropriate choice of $\theta \in \mathbb{R}^{M}$. We shall refer to

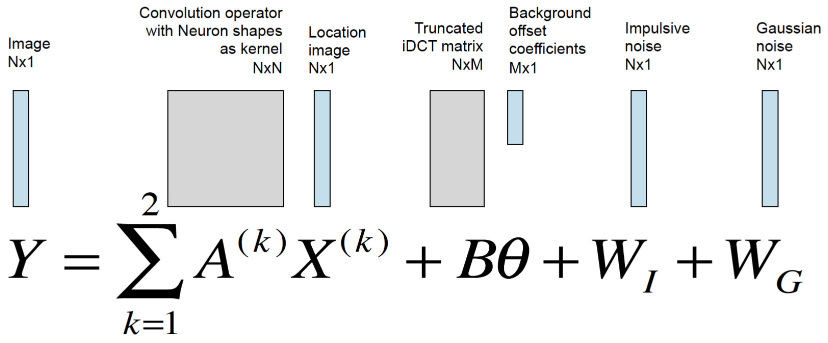


Figure 1.3. Illustration of all the variables in the forward model and their sizes

$B\theta$ as the background offset and θ as the background offset coefficients. Using these, we model the image Y as

$$Y = \sum_{k=1}^{2} A^{(k)} X^{(k)} + B\theta + W_I + W_G, \tag{1.1}$$

where W_I represents additive impulsive noise from dendrites, and W_G represents additive Gaussian noise. The variables in equation 1.1 and their sizes are illustrated in figure 1.3

We consider the Gaussian noise to be white so that the forward model (likelihood of the data, Y given unknown variables $X^{(1)}$, $X^{(2)}$, W_I and θ) can be written as

$$p(Y|X^{(1)}, X^{(2)}, \theta, W_I) = \frac{1}{z_F} exp\Big(- \frac{1}{2\sigma^2_{w_G}} \|Y - \sum_{k=1}^{2} A^{(k)} X^{(k)} - B\theta - W_I\|_2^2 \Big), \tag{1.2}$$

where z_F is the normalizing constant or the partition function of the forward model and $\sigma^2_{w_G}$ is the variance of the gaussian noise.

In the prior model formulation, we model the variables $X^{(1)}$, $X^{(2)}$, W_I as independent. Thus their joint prior probability $p(X^{(1)}, X^{(2)}, W_I)$ can be factored into the individual prior probabilities $p_1(X^{(1)})$, $p_2(X^{(2)})$ and $p_I(W_I)$ as:

$$p(X^{(1)}, X^{(2)}, W_I) = p_1(X^{(1)}) p_2(X^{(2)}) p_I(W_I) \tag{1.3}$$

Since, $X^{(1)}$, $X^{(2)}$ and W_I are sparse, we model their negative log prior to be proportional to their L1 norm. The L1 norm prior is known to induce sparsity in a convex fashion [28]. The individual prior probabilities are then given by:

$$p_k(X^{(k)}) = \frac{1}{z_k} exp\Big(- \frac{1}{\sigma_i} \|X^{(i)}\|_1 \Big) \quad , k = 1, 2 \tag{1.4}$$

and

$$p_I(W_I) = \frac{1}{z_{W_I}} exp\Big(- \frac{1}{\sigma_{W_I}} \|W_I\|_1 \Big), \tag{1.5}$$

where z_1, z_2 , z_{W_I} are the corresponding partition functions and σ_1, σ_2 , σ_{W_I} are scale parameters.

1.3 Neuron Centers Estimation

Since the location of the neuron centers are encoded in the location images , $X^{(1)}$ and $X^{(2)}$, they need to be estimated first in order to calculate the neuron centers. The variables θ and W_I are not needed for calculating the neuron centers but they need to be estimated in order to estimate $X^{(1)}$ and $X^{(2)}$. For estimating the location images we use the maximum a posteriori probability (MAP) estimate. Since we also have the parameter vector θ , we find the joint MAP estimate of $X^{(1)}$,$X^{(2)}$, W_I and θ as:

$$\begin{aligned} \{X^{(1)^*}, X^{(2)^*}, \theta^*, W_I^*\} = \\ \underset{X^{(1)}, X^{(2)}, \theta, W_I}{\arg\min} \Big\{ - log(p(Y|X^{(1)}, X^{(2)}, \theta, W_I)) \\ - log(p(X^{(1)}, X^{(2)}, W_I)) \Big\}, \end{aligned} \tag{1.6}$$

where $p(Y|X^{(1)}, X^{(2)}, \theta, W_I)$ is the likelihood of the image Y given $X^{(1)}, X^{(2)}, \theta, W_I$ and $p(X^{(1)}, X^{(2)}, W_I)$ is the prior probability of $X^{(1)}$, $X^{(2)}$ and W_I. Using equations 1.4 and 1.5 we can find the MAP estimate as:

$$
\begin{aligned}
\{X^{(1)^*}, X^{(2)^*}, \theta^*, W_I^*\} = \\
\operatorname*{arg\,min}_{X^{(1)},X^{(2)},\theta,W_I} \Bigg\{ \frac{1}{2\sigma_{w_G}^2} \|Y - \sum_{k=1}^{2} A^{(k)}X^{(k)} - B\theta - W_I\|_2^2 \\
+ \sum_{k=1}^{2} \frac{1}{\sigma_k} \|X^{(k)}\|_1 + \frac{1}{\sigma_{W_I}} \|W_I\|_1 \ . \quad (1.7)
\end{aligned}
$$

The MAP estimate is computed by iteratively minimizing the cost function above. This is similar to the LASSO cost function [29]. We use the iterative co-ordinate descent method [30] to minimize the cost function in equation 1.7. We iteratively minimize the cost function in equation 1.7 until the change in forward model cost, $\frac{1}{2\sigma_{w_G}^2}\|Y - \sum_{k=1}^{2} A^{(k)}X^{(k)} - B\theta - W_I\|_2^2$, falls below a certain threshold ϵ.

After the location images $X^{(1)}$ and $X^{(2)}$ have been estimated, we can find the neuron center locations by finding the location of its non-zero voxels. However, due to imperfect matching between our shape model and the shape of the neurons in the data, we might have consecutive non-zero voxel values in the estimated location images $X^{(1)}$ and $X^{(2)}$ leading to duplicate neuron centers. To mitigate this, we compute the normal and over-expressed neuron centers as the local maxima of location images $X^{(1)}$ and $X^{(2)}$ respectively. In other words, we consider the location $s \in S$ to be a neuron center if $X^{(k)}_{\Omega(s)} > 0$ and $X^{(k)}_{\Omega(s)} > X^{(k)}_{\Omega(r)} \forall r \in \delta s$, $k = 1, 2$, where the neighborhood δs of s is given by:

$$\delta s = \{r \in S \mid \|W^{-1}(s - r)\|_2 < 1, r \neq s\}, \tag{1.8}$$

where $W = diag(w_x, w_y, w_z)$ is a weighting matrix providing the size of the neighborhood.

The overview of the full process of neuron center estimation is shown in figure 1.4.

1.4 Shape Model Estimation

The impulse responses of $A^{(1)}$ and $A^{(2)}$ serve as shape models and are estimated from training data. The impulse responses of $A^{(1)}$ and $A^{(2)}$ are image patches that represent the shape of the typical neuron of normal and over-expressed type respectively as shown in the

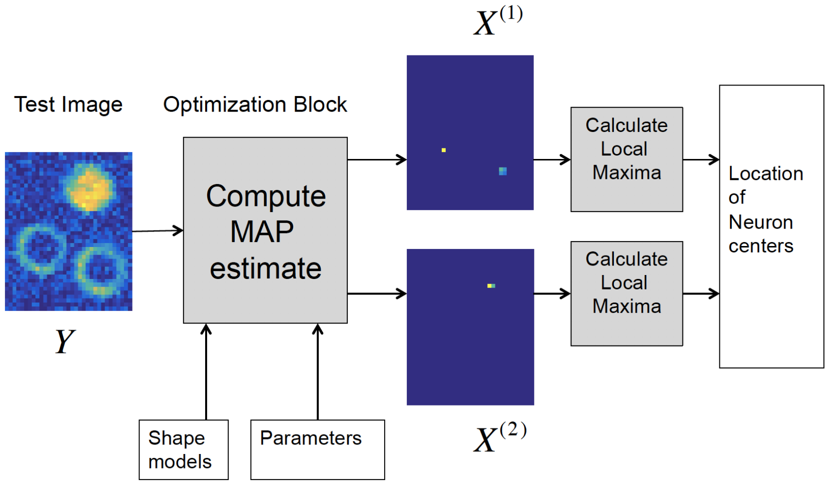


Figure 1.4. Overview of neuron center estimation in MBND

first images in figure 1.6(a)(b). Let the impulse response of $A^{(1)}$ and $A^{(2)}$ be $u^{(1)}$, $u^{(2)} \in \mathbb{R}^L$ in vectorized form. We extract neuron patches from the training volume to estimate $u^{(1)}$ and $u^{(2)}$. Let Z be the vectorized training volume. We manually select P_1 neuron centers of the normal type and P_2 neuron centers of the over-expressed type in the training volume and train the shape models using these. These neuron centers need not be exhaustive, i.e. there might be other neuron centers present in the training volume apart from these. To estimate the shape models that best fit the training volume, we minimize the minimum reconstruction error in equation 1.7 in the neuron region with respect to the shape models. To simplify the training, we find the background offset variable, θ by minimizing the reconstruction error with respect to θ to get $\theta = B^T Z$. We then compute the background offset subtracted training volume, $\widetilde{Z}$ as:

$$\widetilde{Z} = Z - BB^T Z \tag{1.9}$$

We then extract P_1 training patches of size L from $\widetilde{Z}$ centered around the P_1 neuron centers of the normal type and put them in the matrix $Y^{(1)} \in \mathbb{R}^{L\times P_1}$. Similarly, we extract P_2 training patches of size L from $\widetilde{Z}$ centered around the P_2 neuron centers of the over-expressed type and put them in the matrix $Y^{(2)} \in \mathbb{R}^{L\times P_2}$. The process of extracting training patches and estimating the shape models is illustrated in figure 1.5.

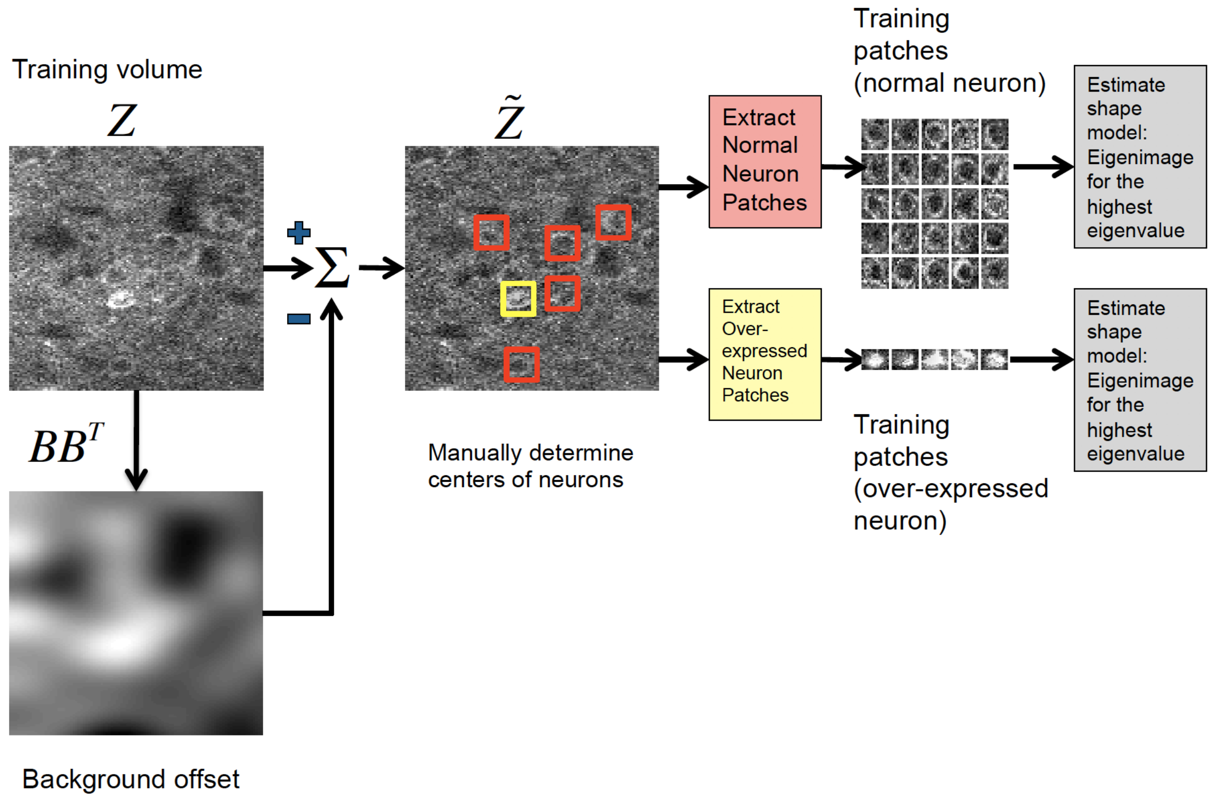

Training volume
Z
BB^T
Background offset
+
−
Σ
$\tilde{Z}$
Manually determine centers of neurons
Extract Normal Neuron Patches
Extract Over-expressed Neuron Patches
Training patches (normal neuron)
Training patches (over-expressed neuron)
Estimate shape model: Eigenimage for the highest eigenvalue
Estimate shape model: Eigenimage for the highest eigenvalue

Figure 1.5. Overview of the process of extracting training patches and estimating the shape models

Then, to estimate $u^{(k)}$ for $k = 1, 2$, we minimize the minimum reconstruction error with respect to $u^{(k)}$ as:

$$u^{(k)} = \arg\min_{u \in \mathbb{R}^L} \left\{ \sum_{i=1}^{P_k} \min_{\rho \in \mathbb{R}} \|Y_{*,i}^{(k)} - \rho u\|_2^2 \right\} \tag{1.10}$$

Define the sample correlation matrices $R_Y^{(k)} \in \mathbb{R}^{L \times L}$, for $k = 1, 2$ as

$$R_Y^{(k)} = \frac{1}{P_k} Y^{(k)} (Y^{(k)})^T \ . \tag{1.11}$$

Solving the least squares problem in equation 1.10 gives

$$u^{(k)} = \arg\max_{u \in \mathbb{R}^L} \left(\frac{u^T R_Y^{(k)} u}{\|u\|_2^2} \right) . \tag{1.12}$$

Let the eigenvalues and eigenvectors of $R_Y^{(k)}$ be $\Lambda^{(k)}$ and $E^{(k)}$ for $k = 1, 2$. So that

$$R_Y^{(k)} = E^{(k)} \Lambda^{(k)} (E^{(k)})^T \ . \tag{1.13}$$

The solution to equation 1.10 is well known: $u^{(k)}$ is the eigenvector of $R_Y^{(k)}$ corresponding to the highest eigenvalue. Since the solution to equation 1.12 is scale invariant, we set $\|u^{(k)}\|_2 = (\Lambda_{1,1}^{(k)})^{1/2}$ to capture the brightness scale of the neurons:

$$u^{(k)} = (\Lambda_{1,1}^{(k)})^{1/2} E_{*,1}^{(k)} \ . \tag{1.14}$$

1.5 Experimental Results

In this section we compare our Model Based Neuron Detection algorithm (MBND) with CellSegm [31]. CellSegm is a publicly available software for high throughput cell segmentation in fluorescence microscopy images. We use the segmct method within CellSegm toolbox which segments cells by iterative thresholding, hole filling and classification based on sizes of regions above threshold. We take the connected component centers of the final segmentation result to be the neuron centers.

1.5.1 Dataset Description

We use GCaMP6 labeled calcium imaging data of a mouse brain to evaluate our neuron detection algorithm. The data was provided by Prof. Meng Cui and Dr. Lingjie Kong at Purdue University. The full 3D data volume is of size 512x512x421 voxels along the x,y and z axes respectively. The inter-voxel distance is $1\mu m$. The neuron cells are approximately ellipsoids with diameter $10\mu m$ along each axis. Contrast stretching is done on the whole volume to improve the contrast and then the voxel values are normalized in the range 0-1. A section of the volume (after contrast stretching and normalization) is shown in figure 1.1 as an example.

We test our algorithm on a section of the full volume of size 104x101x21 that we will call test volume henceforth. We collect neuron patches for training from a separate section of size 351x201x51 from the whole volume. We will call this the training volume henceforth. We do not use the whole volume since collecting ground truth is time and labor intensive.

1.5.2 Training

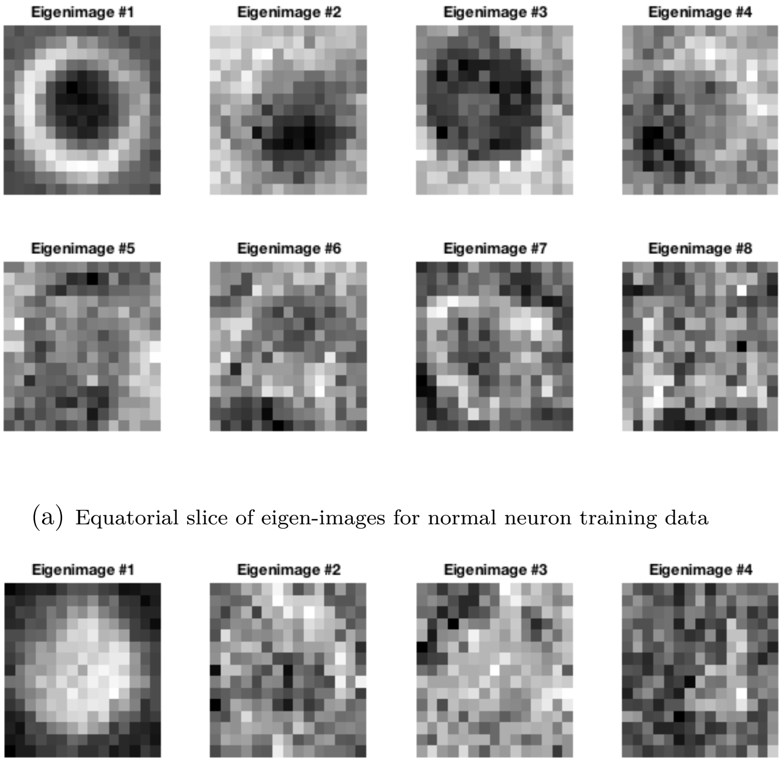
We subtract the low-frequency background from the training volume and then extract 84 centered neuron patches from it, 79 of them normal neuron patches and 5 over-expressed ones. For background subtraction we construct B by using the first 6 , 5 , 11 basis vectors of the 1D Discrete cosine transforms along the x,y,z directions respectively so that we have $M = 6 \times 5 \times 11 = 330$ basis vectors of the 3D Discrete cosine transform in B. The patches extracted were of size 15x15x7. After extracting the training patches, the shape models $u^{(1)}$ and $u^{(2)}$ were estimated using solution to equation 1.14. The equatorial slices of the eigen-images for the normal and over-expressed neuron training data are shown in figure 1.6. We only show the eigen-images corresponding to the highest eigenvalues. The eigenvalues are shown in figure 1.7.

As a first order approximation, we use only the eigen-image of the highest eigen-value for the shape models.

1.5.3 Testing and Evaluation

We apply both the CellSegm method and MBND on the test volume. To determine the accuracy of the detected centers we rely on a ground truth: a list of actual neuron center co-ordinates. The ground truth was manually determined by looking at the 3D stack and was verified by an expert. We consider a detected neuron center to be a correct one if it differs from the true center by a certain tolerance value. Namely, if $c_1 = [x_1, y_1, z_1]^T$ is the location of a true neuron center, and $c_2 = [x_2, y_2, z_2]^T$ is the location of a detected neuron center, then c_2 is considered a true detection of c_1 if:

$$\|D^{-1}(c_1 - c_2)\|_2 < 1 \tag{1.15}$$



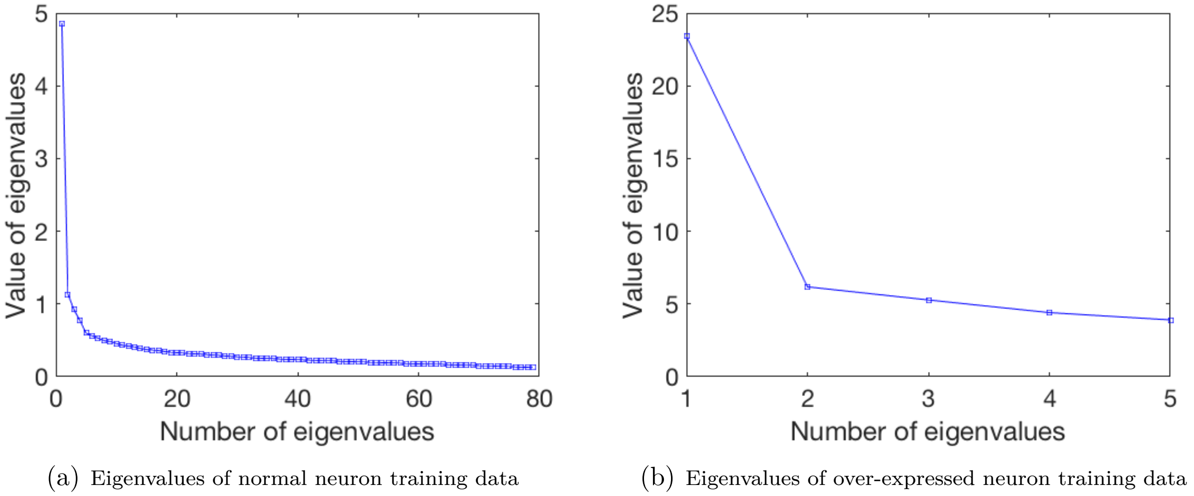
(a) Equatorial slice of eigen-images for normal neuron training data

(b) Equatorial slice of eigen-images for over-expressed neuron training data

Figure 1.6. The equatorial slices of the eigen-images for the (a) normal neuron data (b) neurons affected by GCaMP over-expression. Only the eigen-images corresponding to the highest eigenvalues are shown.

where, $D = diag(d_x, d_y, d_z)$ is a weighing matrix providing the tolerances along each of the x,y,z directions. The values of d_x, d_y, d_z should be small enough so that the ellipsoid with radii d_x, d_y, d_z is well inside the neuron and large enough that small deviations in center co-ordinates are still considered true detection. We take $d_x = 3$, $d_y = 3$, $d_z = 4$ since there are roughly half of the radii of the ellipsoidal neurons.

Each detected neuron center that is a true detection of some neuron in the ground truth is termed a true positive. If multiple neuron centers are detected within the tolerance limit of the actual neuron center, only one of them is considered a true positive and the rest are considered false positives. If a detected neuron center is not a true detection of any neurons in the ground truth then it is termed a false positive. All neuron centers in the ground truth that are not detected are termed false negatives. For a set of detected centers and ground



(a) Eigenvalues of normal neuron training data

(b) Eigenvalues of over-expressed neuron training data

Figure 1.7. Eigenvalues of the (a) normal neuron data (b) neurons affected by GCaMP over-expression

truth, let the total number of true positives, false positives and false negatives be denoted as: N_{TP}, N_{FP} and N_{FN} respectively. We evaluate the list of detected centers by the precision and recall metric defined as:

$$precision = \frac{N_{TP}}{N_{TP} + N_{FP}} \tag{1.16}$$

$$recall = \frac{N_{TP}}{N_{TP} + N_{FN}} \tag{1.17}$$

Precision and recall values lie in the range $[0, 1]$ and high values for both are desirable. Combining both precision and recall, we define a metric F-score as:

$$F = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{1.18}$$

For both MBND and CellSegm, we tune the parameters to get the best F-score on the test volume. We set the noise standard deviation $\sigma_{w_G} = 0.07$. For constructing B, use the first 6, 5, 11 basis vectors of the 1D Discrete cosine transforms along the x,y,z directions respectively so that we have $M = 6 \times 5 \times 11 = 330$ basis vectors of the 3D Discrete cosine transform in B. We set $w_x = w_y = w_z = 4$ for calculating local maxima of the location images. We set the parameters σ_{W_I}, σ_1, σ_2 so that a neuron of typical intensity and of the trained shape can be detected. Let there be a neuron $cu^{(k)}$, $(k = 1, 2)$ embedded in the test image, where c is a scalar constant and $u^{(k)}$ is estimated from equation 1.14. Then to detect it we need to have a non-zero solution to the equation:

$$\rho^* = \arg\min_{\rho \in \mathbb{R}} \frac{1}{2\sigma_{w_G}^2} \|cu^{(k)} - \rho u^{(k)}\|_2^2 + \frac{1}{\sigma_k}|\rho| \tag{1.19}$$

It can be shown that a non-zero solution corresponds to $\sigma_k \geq c\sigma_{w_G}^2 / \|u^{(k)}\|$. This is the motivation behind setting $\sigma_k = c_k \sigma_{w_G}^2 / \|u^{(k)}\|$, $(k = 1, 2)$ and $\sigma_{W_I} = c_I \sigma_{w_G}^2$, where c_1, c_2 and c_{W_I} are scalar constants. We set $c_1 = 3.3$, $c_2 = 2.7$ and $c_{W_I} = 2.5$.

For MBND, the best F-score obtained was 0.91 with corresponding $\textit{precision} = 0.95$ and $\textit{recall} = 0.87$. For CellSegm, the best F-score obtained was 0.13 with corresponding $\textit{precision} = 0.18$ and $\textit{recall} = 0.10$. The neuron detection results for MBND and CellSegm corresponding to their best F-score is displayed in figure 1.8. We also display the original image and ground truth to get a visual comparison. To show the neurons, we draw a sphere around the neuron centers in 3D and we show the circular intersection of the surface of the sphere with the current slice in the figures. A dot in the center of the circle indicates that the neuron center is in the current slice. Ground truth and true detection are shown in green, false positives in red and missed detection in blue.

Next, for MBND we vary the neuron regularizer σ_1 in equation 1.7 and keep the other parameters fixed to get a series of neuron detection results and the corresponding precision and recall values. For the CellSegm method, we vary the threshold and keeping the other parameters fixed to get a series of neuron detection results and the corresponding precision and recall values. We plot these precision recall values to generate a precision-recall plot. The precision recall plot of MBND is compared to that of CellSegm in figure 1.9.

From figures 1.8 and 1.9 we can see that MBND has minimal false positives and is able to detect many more neurons than CellSegm. From figures 1.8(o) and 1.8(g) we can see that CellSegm is able to detect only the very bright neurons. This is because iterative thresholding combined with other image-processing techniques are not good if there isn't a strong difference in illumination level between the object and the background. On the other hand, in MBND we train shape models for the neurons and thus MBND is able to detect neurons even when there isn't much illumination difference between the neuron and the background.

1.6 Conclusion

In this paper we presented a novel model based neuron detection scheme(MBND) to detect neuron centers in fluorescence images of neurons. MBND is able to detect neurons at very low signal to noise ratios. Results show that MBND performs better than one widely used method.

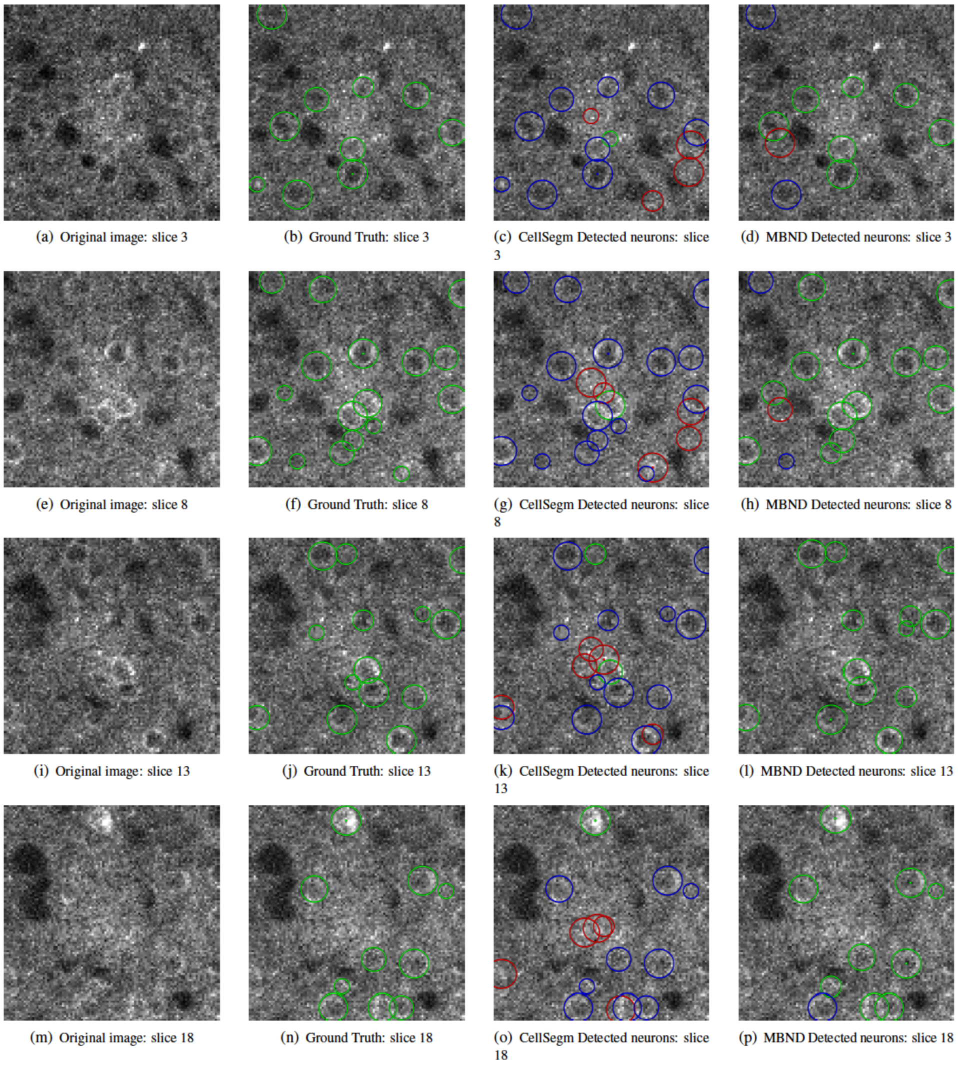

(a) Original image: slice 3
(b) Ground Truth: slice 3
(c) CellSegm Detected neurons: slice 3
(d) MBND Detected neurons: slice 3
(e) Original image: slice 8
(f) Ground Truth: slice 8
(g) CellSegm Detected neurons: slice 8
(h) MBND Detected neurons: slice 8
(i) Original image: slice 13
(j) Ground Truth: slice 13
(k) CellSegm Detected neurons: slice 13
(l) MBND Detected neurons: slice 13
(m) Original image: slice 18
(n) Ground Truth: slice 18
(o) CellSegm Detected neurons: slice 18
(p) MBND Detected neurons: slice 18

Figure 1.8. Slice by slice comparison of detected neurons by our proposed method(MBND) with CellSegm Here (a), (e), (i), (m) show the original test image in slices 3,8,13,18 respectively. (b), (f), (j), (n) show the ground truth in slices 3,8,13,18 respectively. (c), (g), (k), (o) show the neurons detected by CellSegm in slices 3,8,13,18 respectively. (d), (h), (l), (p) show the neurons detected by our MBND in slices 3,8,13,18 respectively. To show the neurons, we draw a sphere around the neuron centers in 3D and we show the circular intersection of the surface of the sphere with the current slice in the figures. A dot in the center of the circle indicates that the neuron center is in the current slice. Ground truth and true detection are shown in green, false positives in red and missed detection in blue.

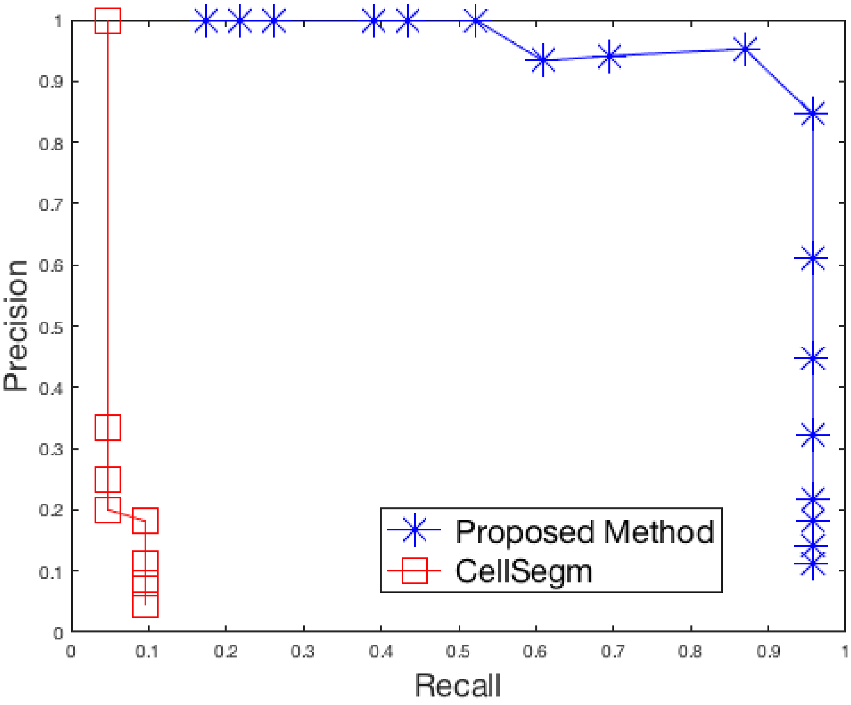


Figure 1.9. Precison recall plot of our proposed method(MBND) compared with CellSegm

2. MULTI-SLICE FUSION FOR SPARSE-VIEW AND LIMITED-ANGLE 4D CT RECONSTRUCTION

2.1 Introduction

Improvements in imaging sensors and computing power have made it possible to solve increasingly difficult reconstruction problems. In particular, the dimensionality of reconstruction problems has increased from the traditional 2D and 3D problems representing space to more difficult 4D or even 5D problems representing space-time and, for example, heart or respiratory phase [32]–[37].

These higher-dimensional reconstruction problems pose surprisingly difficult challenges computationally and perhaps more importantly, in terms of algorithmic design and training due to the curse of dimensionality [38]. However, the high dimensionality of the reconstruction also presents important opportunities to improve reconstruction quality by exploiting the regularity in the high-dimensional space. In particular, for time-resolved imaging, we can exploit the regularity of the image to reconstruct each frame with fewer measurements and thereby increase temporal resolution. In the case of 4D CT, the contributions of [33], [39], [40] have increased the temporal resolution by an order of magnitude by exploiting the space-time regularity of objects being imaged. These approaches use model-based iterative reconstruction (MBIR) [41], [42] to enforce regularity in 4D using simple space-time prior models. More recently, deep learning based post-processing for 4D reconstruction has been proposed as a method to improve reconstructed image quality [43].

Recently, it has been demonstrated that plug-and-play (PnP) priors [44]–[47] can dramatically improve reconstruction quality by enabling the use of state-of-the-art denoisers as prior models in MBIR. So PnP has great potential to improve reconstruction quality in 4D CT imaging problems. However, state-of-the-art denoisers such as deep convolutional neural networks (CNN) and BM4D are primarily available for 2D and sometimes 3D images, and it is difficult to extend them to higher dimensions [38], [48], [49]. In particular, extending CNNs to 4D requires very computationally and memory intensive 4D convolution applied to 5D feature tensor structures. This problem is further compounded by the lack of GPU accelerated routines for 4D convolution from major Deep-Learning frameworks such as Ten-

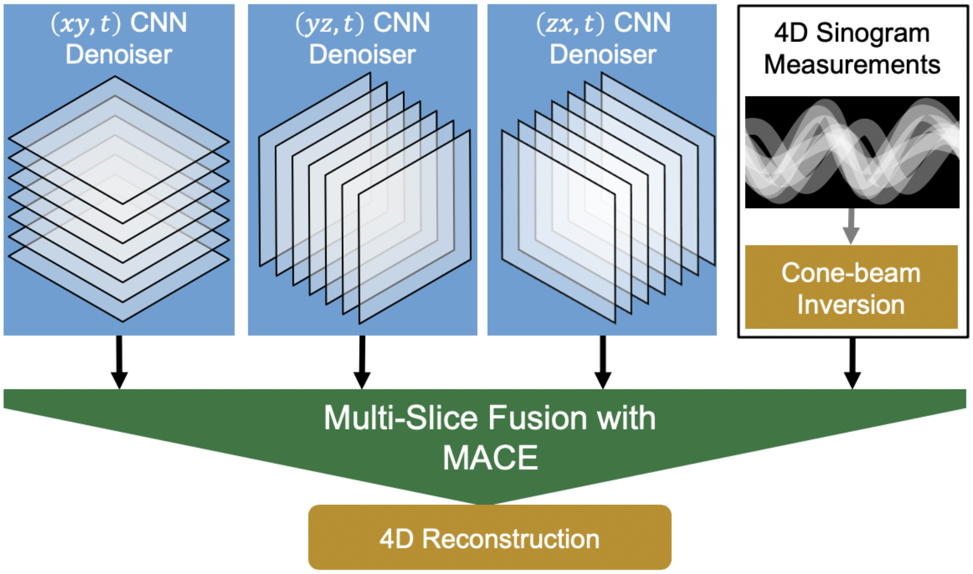


Figure 2.1. Illustration of our multi-slice fusion approach. Each CNN denoiser operates along the time direction and two spatial directions. We fuse the CNN denoisers with the measurement model to produce a 4D regularized reconstruction.

sorflow, Keras, PyTorch [1]. Furthermore, 4D CNNs require 4D ground truth data to train the PnP denoisers, which might be difficult or impossible to obtain.

In this paper, we present a novel 4D X-ray CT reconstruction algorithm that combines multiple low-dimensional CNN denoisers to implement a highly effective 4D prior model. Our approach, *multi-slice fusion*, integrates the multiple low-dimensional priors using multi-agent consensus equilibrium (MACE) [50]. MACE is an extension of the PnP framework that formulates the inversion problem using an equilibrium equation—as opposed to an optimization—and allows for the use of multiple prior models and agents.

Figure 2.1 illustrates the basic concept of our approach. Multi-slice fusion integrates together three distinct CNN denoisers each of which is trained to remove additive white Gaussian noise along lower dimensional slices (hyperplanes) of the 4D object. When MACE fuses the denoisers it *simultaneously* enforces the constraints of each denoising agent, so that the reconstructions are constrained to be smooth in all four dimensions. Consequently, multi-slice fusion results in high-quality reconstructions that are practical to train and compute

[1] ↑Currently only 1D, 2D, and 3D convolutions are supported with GPU acceleration

even when the dimensionality of the reconstruction is high. In our implementation, one MACE agent estimates the cone-beam tomographic inversion. The remaining 3 agents are CNN denoisers trained to remove additive white Gaussian noise along two spatial directions and the time direction. The CNN agents work along complimentary spatial directions and are designed to take as input a stack of five 2D slices from five neighboring time-points. We refer to this as 2.5D denoising [38], [51]. Further details are given in Section 2.4.

The MACE solution can be computed using a variety of algorithms, including variants of the plug-and-play algorithm based on ADMM or other approaches [44], [45], [52], [53]. We implement multi-slice fusion on distributed heterogeneous clusters in which different agent updates are distributed onto different cluster nodes. In particular, the cone-beam inversion computations are distributed onto multiple CPU nodes and concurrently, the CNN denoising computations are distributed onto multiple GPU nodes.

We present experiments using both simulated and real data of 4D NDE tomographic imaging from sparse-views, and we compare multi-slice fusion with MBIR using total variation (TV) and 4D Markov random field (MRF) priors. Our results indicate that multi-slice fusion can substantially reduce artifacts and increase resolution relative to these alternative reconstruction methods.

The rest of the paper is organized as follows. In section 2.2, we introduce the problem of 4D CT reconstruction. In section 2.3, we introduce the theory behind MACE model fusion. In section 2.4, we use the MACE framework to introduce multi-slice fusion. In section 2.5, we describe our training pipeline for training the CNN denoisers. In section 2.6, we describe our distributed implementation of multi-slice fusion on heterogeneous clusters. Finally, in section 2.7, we present results on sparse-view and limited-angle 4D CT using both simulated and real data.

2.2 Problem Formulation

In 4D X-ray CT imaging, a dynamic object is rotated and several 2D projections (radiographs) of the object are measured for different angles as illustrated in Figure 2.2. The problem is then to reconstruct the 4D array of X-ray attenuation coefficients from these measurements, where three dimensions correspond to the spatial dimensions and the fourth dimension corresponds to time.

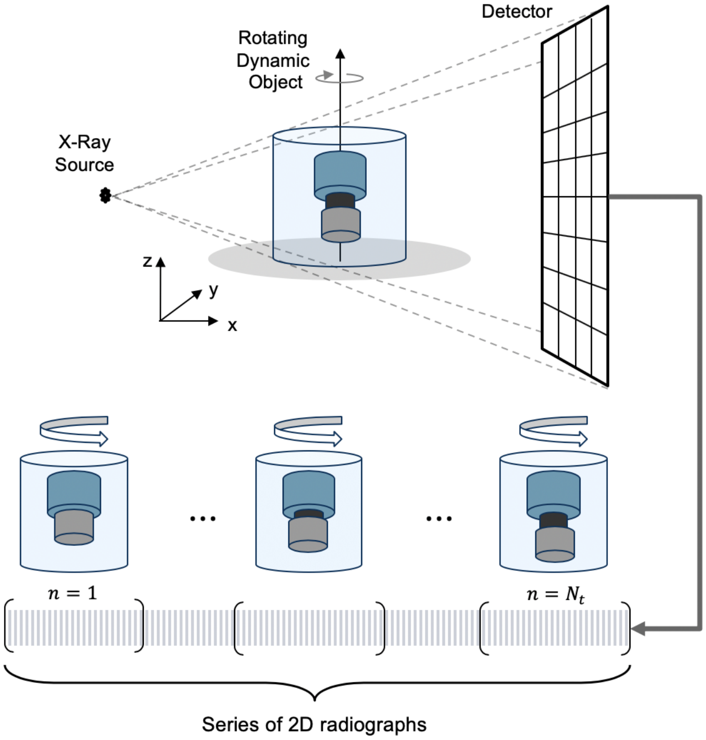


Figure 2.2. Illustration of 4D cone-beam X-ray CT imaging. The dynamic object is rotated and several 2D projections (radiographs) of the object are measured for different angles. The projections are divided into N_t disjoint subsets for each of the N_t time-points.

Let N_t be the number of time-points, M_n be the number of measurements at each time-point, and N_s be the number of voxels at each time-point of the 4D volume. For each time-point $n \in \{1, \ldots, N_t\}$, define $y_n \in \mathbb{R}^{M_n}$ to be the vector of sinogram measurements at time n, and $x_n \in \mathbb{R}^{N_s}$ to be the vectorized 3D volume of X-ray attenuation coefficients for that time-point. Let us stack all the measurements to form a measurement vector $y = [y_1^\top, .., y_{N_t}^\top]^\top \in \mathbb{R}^M$ where $M = \sum_{n=1}^{N_t} M_n$ is the total number of measurements. Similarly, let us stack the 3D volumes at each time-point to form a vectorized 4D volume $x = [x_1^\top, \ldots, x_{N_t}^\top]^\top \in \mathbb{R}^N$, where $N = N_t N_s$ is the total number voxels in the 4D volume. The 4D reconstruction problem then becomes the task of recovering the 4D volume of attenuation coefficients, x, from the series of sinogram measurements, y.

In the traditional maximum a posteriori (MAP) approach, the reconstruction is given by

$$x^* = \arg\min_x \left\{ l(x) + \beta h(x) \right\} \ , \tag{2.1}$$

where $l(x)$ is the data-fidelity or log-likelihood term, $h(x)$ is the 4D regularizer or prior model, and the unit-less parameter β controls the level of regularization in the reconstruction. The data-fidelity term, $l(x)$, can be written in a separable fashion as

$$l(x) = \frac{1}{2} \sum_{n=1}^{N_t} \| y_n - A_n x_n \|^2_{\Lambda_n} \ , \tag{2.2}$$

where A_n is the system matrix, and Λ_n is the weight matrix for time-point n. The weight matrix accounts for the non-uniform noise variance due to a Gaussian approximation [30] of the underlying Poisson noise. The weight matrix is computed as $\Lambda_n = \text{diag}\{c \exp\{-y_n\}$ where the scalar c is empirically chosen [33].

If the prior model, $h(x)$, can be expressed analytically like a 4D Markov random field (MRF) as in [33], [35], then the expression in equation (2.1) can be minimized iteratively to reconstruct the image. However, in practice, it can be difficult to represent an advanced prior model in the form of a tractable cost function $h(x)$ that can be minimized. Consequently, PnP algorithms have been created as a method for representing prior models as denoising operations[44], [45]. More recently, PnP methods have been generalized to the multi-agent consensus equilibrium (MACE) framework as a way to integrate multiple models in a principled manner [35], [50], [54].

2.3 MACE Model Fusion

In this section, we use the multi-agent consensus equilibrium (MACE) framework to fuse the data-fidelity term and multiple denoisers; these multiple denoisers form a single prior model for reconstruction. This allows us to construct a 4D prior model using low-dimensional CNN denoisers (described in Section 2.4).

To introduce the concept of consensus equilibrium, let us first consider a variation of the optimization problem in equation (2.1) with K regularizers $h_k(x)$, $k = 1, \ldots, K$. The modified optimization problem can thus be written as

$$x^* = \arg\min_x \left\{ l(x) + \frac{\beta}{K} \sum_{k=1}^{K} h_k(x) \right\} \ , \tag{2.3}$$

where the normalization by K is done to make the regularization strength independent of the number of regularizers.

Now we transform the optimization problem of equation (2.3) to an equivalent consensus equilibrium formulation. However, in order to do this, we must introduce additional notation. First, we define the proximal maps of each term in equation (2.3). We define $L(x) : \mathbb{R}^N \rightarrow \mathbb{R}^N$ to be the proximal map of $l(x)$ as

$$L(x) = \arg\min_{z \in \mathbb{R}^N} \left\{ l(z) + \frac{1}{2\sigma^2} \|x - z\|_2^2 \right\} \ , \tag{2.4}$$

for some $\sigma > 0$. Similarly, we define $H_k(x) : \mathbb{R}^N \rightarrow \mathbb{R}^N$ to be the the proximal map of each $h_k(x)$, $k = 1, \ldots, K$ as

$$H_k(x) = \arg\min_{z \in \mathbb{R}^N} \left\{ \frac{1}{2\sigma^2} \|x - z\|_2^2 + h_k(z) \right\} \ . \tag{2.5}$$

Each of these proximal maps serve as agents in the MACE framework. We stack the agents together to form a stacked operator $F : \mathbb{R}^{(K+1)N} \rightarrow \mathbb{R}^{(K+1)N}$ as

$$F(W) = \begin{bmatrix} L(W_0) \\ H_1(W_1) \\ \vdots \\ H_K(W_K) \end{bmatrix} \ , \tag{2.6}$$

where $W \in \mathbb{R}^{(K+1)N}$ is stacked representative variable. The consensus equilibrium is the vector $W^* \in \mathbb{R}^{(K+1)N}$ that satisfies

$$F(W^*) = G(W^*) \ , \tag{2.7}$$

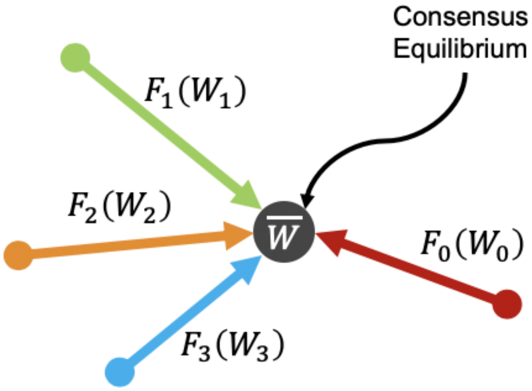


Figure 2.3. Illustration of consensus equilibrium as analogous to a force balance equation: each agent pulls the solution toward its manifold and at equilibrium the forces balance each other.

where G is an averaging operator given as

$$G(W) = \begin{bmatrix} \overline{W} \\ \vdots \\ \overline{W} \end{bmatrix} , \tag{2.8}$$

and the weighted average is defined as

$$\overline{W} = \frac{1}{1+\beta} W_0 + \frac{\beta}{1+\beta} \left(\frac{1}{K} \sum_{k=1}^{K} W_k \right) . \tag{2.9}$$

Notice the weighting scheme is chosen to balance the forward and prior models. The unitless parameter β is used to tune the weights given to the prior model and thus the regularization of the reconstruction. Equal weighing of the forward and prior models can be achieved using $\beta = 1$.

If W^* satisfies the consensus equilibrium condition of equation (2.7), then it can be shown [50] that $\overline{W^*}$ is the solution to the optimization problem in equation (2.3). Thus if the agents in MACE are true proximal maps then the consensus equilibrium solves an equivalent optimization problem.

However, if the MACE agents are not true proximal maps, then there is no inherent optimization problem to be solved, but the MACE solution still exists. In this case, the

MACE solution can be interpreted as the balance point between the forces of each agent as illustrated in Figure 2.3. Each agent pulls the solution toward its manifold and the consensus equilibrium solution represents a balance point between the forces of each agent. Thus MACE provides a way to incorporate non-optimization based models such as deep neural networks for solving inverse problems.

To see how we can incorporate deep neural network based prior models, first notice that equation (2.5) can be interpreted as the MAP estimate for a Gaussian denoising problem with prior model h_k and noise standard deviation σ. Thus we can replace each MACE operator, H_k, for each $k = 1, \ldots, K$ in equation (2.5) with a deep neural network trained to remove additive white Gaussian noise of standard deviation σ.

It is interesting to note that when H_k is implemented with a deep neural network denoiser, then the agent H_k is not, in general, a proximal map and there is no corresponding cost function h_k. We know this because for H_k to be a proximal map, it must satisfy the condition that $\nabla H_k(x) = [\nabla H_k(x)]^\top$ (see [44], [55]), which is equivalent to H_k being a conservative vector function (see for example [56, Theorem 2.6, p. 527]). For a CNN, ∇H_k is a function of the trained weights, and in the general case, the condition will not be met unless the CNN architecture is specifically designed to enforce such a condition.

The consensus equilibrium equation 2.7 states the condition that the equilibrium solution must satisfy. However, the question remains of how to compute this equilibrium solution. Our approach to solving the consensus equilibrium equations is to first find an operator that has the equilibrium solution as a fixed point, and then use standard fixed point solvers. To do this, we first notice that the averaging operator has the property that $G(G(W)) = G(W)$. Intuitively, this is true because applying averaging twice is the same as applying it once. Using this fact, we see that

$$(2G - I)(2G - I) = 4GG - 4G + I = I \ , \tag{2.10}$$

where I is the identity mapping. We then rewrite equation (2.7) as

$$\begin{aligned} FW^* &= GW^* \\ (2F - I)W^* &= (2G - I)W^* \\ (2G - I)(2F - I)W^* &= W^* \ . \end{aligned}$$

So from this we see that the following fixed point relationship must hold for the consensus equilibrium solution.

$$(2G - I)(2F - I)W^* = W^* \ , \tag{2.11}$$

and the consensus equilibrium solution W^* is a fixed point of the mapping $T = (2G-I)(2F-I)$.

We can apply a variety of iterative fixed point algorithms to equation (2.11) to compute the equilibrium solution. These algorithms have varying convergence guarantees and convergence speeds [50]. One such algorithm is Mann iteration [50], [54], [57]. Mann iteration performs the following pseudo-code steps until convergence where ← indicates assignment of a psuedo-code variable.

$$W \leftarrow (1 - \rho)W + \rho TW \ , \tag{2.12}$$

where weighing parameter $\rho \in (0, 1)$ is used to control the speed of convergence. In particular, when $\rho = 0.5$, the Mann-iteration solver is equivalent to the consensus-ADMM algorithm [50], [58]. It can be shown that the Mann iteration converges to a fixed point of $T = (2G-I)(2F-I)$ if T is a non-expansive mapping [50].

Note that each Mann iteration update in equation (2.12) involves performing the minimization in equation (2.4). This nested iteration is computationally expensive and leads to slow convergence. Instead of minimizing equation (2.4) till convergence, we initialize with the result of the previous Mann iteration and perform only three iterations of iterative coordinate descent (ICD). We denote this partial update operator as $\tilde{L}(W_0, X_0)$ where X_0 is the initial condition to the iterative update. The corresponding new F operator approximation is then given by

$$\tilde{F}(W;X) = \begin{bmatrix} \tilde{L}(W_0;X_0) \\ H_1(W_1) \\ \vdots \\ H_k(W_K) \end{bmatrix} . \tag{2.13}$$

Algorithm 1 shows a simplified Mann iteration using partial updates. We perform algebraic manipulation of the traditional Mann iterations[54], [57] in order to obtain the simplified but equivalent Algorithm 1. It can be shown that partial update Mann iteration also converges [54], [57] to the fixed point in equation (2.11). We used a zero initialization, $x^{(0)} = 0$, in all our experiments and continue the partial update Mann iteration until the differences between state vectors X_k become smaller than a fixed threshold.

Algorithm 1: Partial update Mann iteration for computing the MACE solution

```
Input: Initial Reconstruction: $x^{(0)} \in \mathbb{R}^N$
Output: Final Reconstruction: $x^*$
$X \leftarrow W \leftarrow \begin{bmatrix} x^{(0)} \\ \vdots \\ x^{(0)} \end{bmatrix}$
while not converged do
    $X \leftarrow \tilde{F}(W;X)$
    $Z \leftarrow G(2X - W)$
    $W \leftarrow W + 2\rho(Z - X)$
$x^* \leftarrow X_0$
```

2.4 Multi-Slice Fusion using MACE

We use four MACE agents to implement multi-slice fusion. We set $K = 3$ and use the names $H_{\text{xy,t}}$, $H_{\text{yz,t}}$, $H_{\text{zx,t}}$ to denote the denoising agents H_1, H_2, H_3 in equation (2.6). The agent L enforces fidelity to the measurement while each of the denoisers $H_{\text{xy,t}}$, $H_{\text{yz,t}}$, $H_{\text{zx,t}}$ enforces regularity of the image in orthogonal image planes. MACE imposes a consensus between the operators L, $H_{\text{xy,t}}$, $H_{\text{yz,t}}$, $H_{\text{zx,t}}$ to achieve a balanced reconstruction that lies

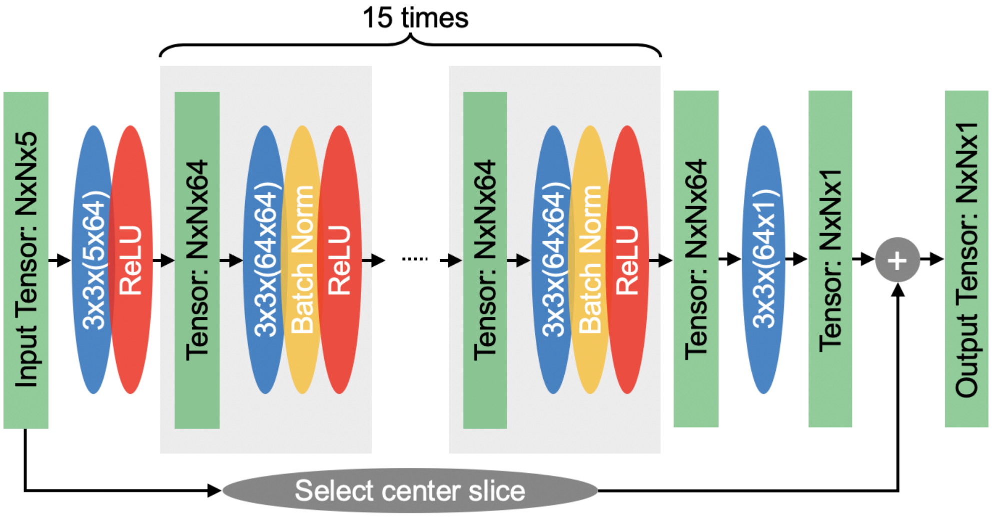


Figure 2.4. Architecture of our 2.5D CNN denoiser. Different sizes of input and output necessitate a selection operator for the residual connection. Each green rectangle denotes a tensor, and each ellipse denotes an operation. Blue ellipses specify the shape of the convolution kernel.

at the intersection of the solution space of the measurement model and each of the prior models. The MACE stacked operator F encompassing all four agents can be written as

$$F(W) = \begin{bmatrix} L(W_0) \\ H_{\mathrm{xy,t}}(W_1) \\ H_{\mathrm{yz,t}}(W_2) \\ H_{\mathrm{zx,t}}(W_3) \end{bmatrix} . \tag{2.14}$$

Here the representative variable $W \in \mathbb{R}^{4N}$ is formed by stacking four vectorized 4D volumes.

The three denoisers $H_{\mathrm{xy,t}}$, $H_{\mathrm{yz,t}}$, and $H_{\mathrm{zx,t}}$ share the same architecture and trained model but are applied along different planes of the 4D space. The CNN architecture is shown in Figure 2.4. We have modified a typical CNN architecture [59] to input information from a third dimension. The channel dimension of a convolution layer is typically used to input multiple color channels for denoising 2D color images using CNNs. We re-purpose the channel dimension to input five adjacent 2D slices of the noisy image to the network and output the denoised center slice. The other slices are being denoised by shifting the 5-slice moving window. We call this 2.5D since the receptive field along the convolution dimensions is large but in the channel dimension is small. It has been shown that this type of 2.5D processing is a computationally efficient way of performing effective 3D denoising with CNNs [38], [51]. We use the notation $H_{\mathrm{xy,t}}$ to denote a CNN space-time denoiser that performs convolution in the xy-plane and uses the convolution channels to input slices from neighboring time-points. The denoisers $H_{\mathrm{yz,t}}$ and $H_{\mathrm{zx,t}}$ are analogous to $H_{\mathrm{xy,t}}$ but are applied along the yz and zx-plane, respectively. This orientation of the three denoisers ensures that

1. The spatial dimensions x, y, z are treated equivalently. This ensures the regularization to be uniform across all spatial dimensions;

2. Each dimension in x, y, z, and t is considered at least once. This ensures that model fusion using MACE incorporates information along all four dimensions.

Since the three denoising operators $H_{\mathrm{xy,t}}$, $H_{\mathrm{yz,t}}$, and $H_{\mathrm{zx,t}}$ process the 4D volume “slice by slice”, they can be implemented in parallel on large scale parallel computers. Details on distributed implementation are described in section 2.6.

2.5 Training of CNN Denoisers

All three prior model agents $H_{\mathrm{xy,t}}$, $H_{\mathrm{yz,t}}$, and $H_{\mathrm{zx,t}}$ in multi-slice fusion share the same 2.5D model shown in Figure 2.4 but are oriented along different planes. Consequently we train a single 2.5D CNN model using 3D data. Even though the CNN needs to denoise 3D time-space data, we train it using 3D spatial data since 3D volumes are widely available unlike time-space data.

Figure 2.5 outlines our training data generation. We start with a low-noise 3D CT volume that is representative of the objects to be reconstructed. We extract 3D patches from the CT volume and add pseudo-random additive white Gaussian noise (AWGN) to the patches to generate the training pairs. We then train the CNN to remove the noise. The use of AWGN is due to the mathematical form of the quadratic norm term in the proximal map in equation 2.5 and follows from the theory of Plug-and-play [44], [45].

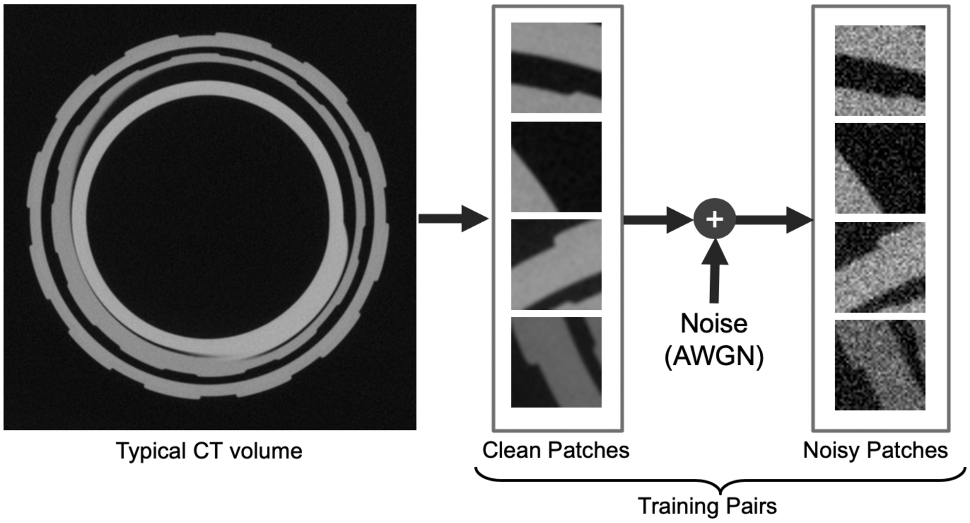


Figure 2.5. Illustration of our training data generation. We extract 3D patches from a typical CT volume and add additive white Gaussian noise (AWGN) to generate training pairs. This makes the training process self-supervised.

2.6 Distributed Reconstruction

The computational structure of multi-slice fusion is well-suited to a highly distributed implementation. The main computational bottleneck in Algorithm 1 is the F operator. Fortunately, F is a parallel operator and thus its individual components L, $H_{\mathrm{xy,t}}$, $H_{\mathrm{yz,t}}$, and $H_{\mathrm{zx,t}}$ can be executed in parallel. The operators L, $H_{\mathrm{xy,t}}$, $H_{\mathrm{yz,t}}$, and $H_{\mathrm{zx,t}}$ can themselves be parallelized internally as well. The distributed implementation of multi-slice fusion is illustrated in Figure 2.6.

The CNN denoisers $H_{\mathrm{xy,t}}$, $H_{\mathrm{yz,t}}$, and $H_{\mathrm{zx,t}}$ are 2.5D denoisers that denoise the 4D volume by processing it slice by slice and thus can be trivially parallelized leading to a large number of concurrent operations. The concurrent operations for all three denoisers are distributed among multiple GPUs due to the availability of optimized GPU routines in Tensorflow. In our experiments we used a GPU cluster with three Nvidia Tesla P100 GPUs to compute the CNN denoising operators.

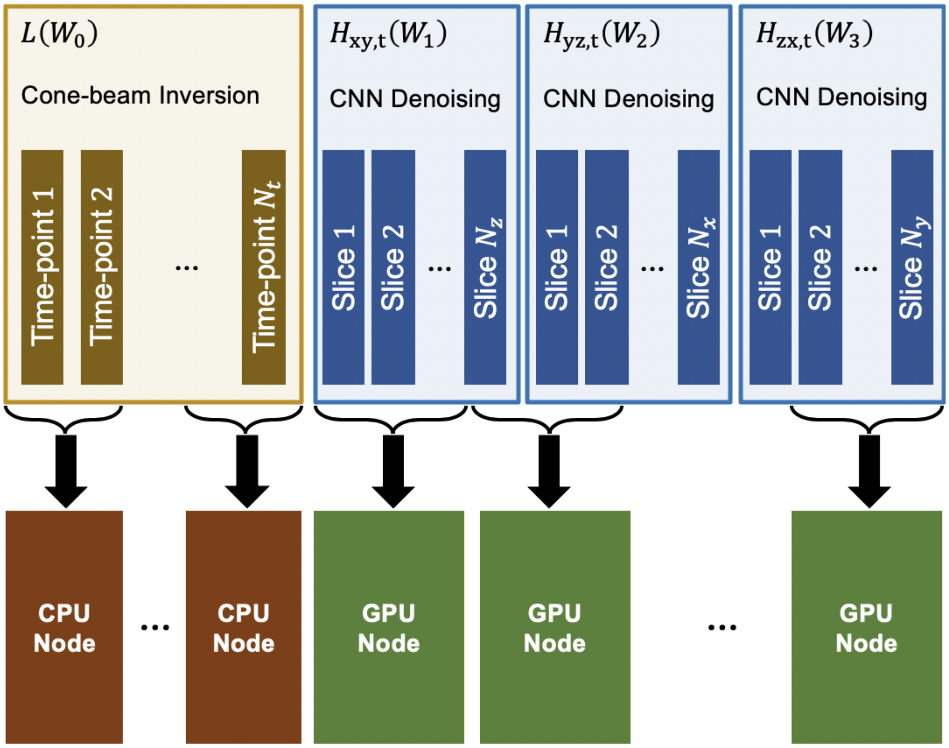
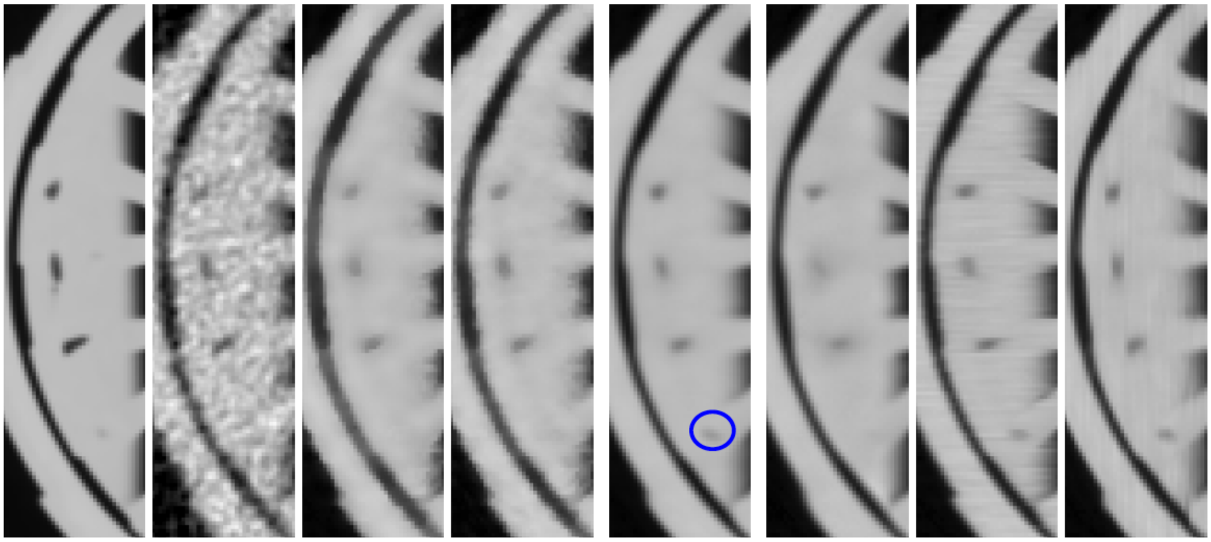


Figure 2.6. Illustration of distributed computation of multi-slice fusion. We perform distributed computation of the F operator which is the main computational bottleneck in Algorithm 1. Each operator within F, namely $H_{\text{xy,t}}$, $H_{\text{yz,t}}$, $H_{\text{zx,t}}$, and L can be executed in parallel. Furthermore, operators $H_{\text{xy,t}}$, $H_{\text{yz,t}}$, $H_{\text{zx,t}}$, and L are 3D operators that can process the 4D volume "slice by slice" leading to a large number of concurrent operations that can be distributed among multiple compute nodes.

The cone-beam inversion operator, L, can also be computed for each time-point independently due to the separable structure in equations (2.4) and (2.2). This leads to a large number of concurrent operations which are distributed among multiple CPU nodes. The cone-beam inversion for each time-point is computed using a coordinate-descent minimization with multi-threaded parallelism. Further details about the cone-beam inversion can be found in [34].



(a) Phantom (b) FBP (3D) (c) MBIR+TV (3D) (d) MBIR+4D-MRF (e) **Multi-slice fusion (proposed)** (f) MBIR+$H_{\mathrm{xy,t}}$ (g) MBIR+$H_{\mathrm{yz,t}}$ (h) MBIR+$H_{\mathrm{zx,t}}$

Figure 2.7. Comparison of different methods for simulated data 360°. Each image is a slice through the reconstructed object for one time-point along the spatial xy-plane. The reconstruction using FBP suffers from high noise and fails to recover the small hole in the bottom of the image. MBIR+TV and MBIR+4D-MRF suffer from jagged edges and fail to recover the small hole in the bottom of the image. MBIR+$H_{\mathrm{yz,t}}$ and MBIR+$H_{\mathrm{zx,t}}$ suffer from horizontal and vertical streaks, respectively, since the denoisers were applied in those planes. MBIR+$H_{\mathrm{xy,t}}$ cannot reconstruct the small hole in the bottom of the image since the xy-plane does not contain sufficient information.

2.7 Experimental Results

We present experimental results on two simulated and two real 4D X-ray CT data for Non-Destructive Evaluation (NDE) applications to demonstrate the improved reconstruction quality of our method. The four experimental cases are outlined below

1. **Simulated Data 360°:** Sparse-view results on simulated data with a set of sparse views ranging over 360° at each reconstructed time-point;

2. **Simulated Data 90°:** Sparse-view limited-angle results on simulated data with a set of sparse views ranging over 90° at each reconstructed time-point;

3. **Real Data 360°:** Sparse-view results on real data with a set of sparse views ranging over 360° at each reconstructed time-point;

4. **Real Data 90°:** Sparse-view limited-angle results on real data with a set of sparse views ranging over 90° at each reconstructed time-point.

The selection of the rotation range per time-point is arbitrary and can be chosen after the measurements have been taken. For example, a full rotation with 400 views can be used as a single time-point or as four time-points with 100 views each. The four time-points per rotation can provide extra temporal resolution, however, they require a more difficult reconstruction with incomplete information.

We compare multi-slice fusion with several other methods outlined below

- FBP: Conventional 3D filtered back projection reconstruction;
- MBIR+TV: MBIR reconstruction using a total variation (TV) prior [60] in the spatial dimensions;
- MBIR+4D-MRF: MBIR reconstruction using 4D Markov random field prior [33] with $q = 2.2$, $p = 1.1$, 26 spatial neighbors and 2 temporal neighbors;
- MBIR+$H_{\mathrm{xy,t}}$: MBIR using the CNN $H_{\mathrm{xy,t}}$ as a PnP prior;
- MBIR+$H_{\mathrm{yz,t}}$: MBIR using the CNN $H_{\mathrm{yz,t}}$ as a PnP prior;
- MBIR+$H_{\mathrm{zx,t}}$: MBIR using the CNN $H_{\mathrm{zx,t}}$ as a PnP prior.

We used two CPU cluster nodes, each with 20 Kaby Lake CPU cores and 96 GB system memory to compute the cone-beam inversion. We used three GPU nodes, each with a Nvidia Tesla P100 GPU (16 GB GPU-memory) and 192 GB system memory to compute the CNN denoisers. To compute the multi-slice fusion reconstruction, we run Algorithm 1 for 10 Mann iterations, with 3 iterations of cone-beam inversion per Mann iteration. The total reconstruction time of multi-slice fusion for each experimental case are given in Table 2.1.

The 2.5D CNN denoiser model used in the reconstructions was trained using a low-noise 3D CT reconstruction of a bottle and screw cap made from different plastics. The object

Table 2.1. Total reconstruction time of multi-slice fusion for each experimental case

Experimental Case	Reconstruction size (x,y,z,t)	Reconstruction Time
Simulated Data 360°	$240 \times 240 \times 28 \times 8$	8 mins
Simulated Data 90°	$240 \times 240 \times 28 \times 8$	8 mins
Real Data 360°	$731 \times 731 \times 91 \times 16$	133 mins
Real Data 90°	$263 \times 263 \times 778 \times 12$	46 mins

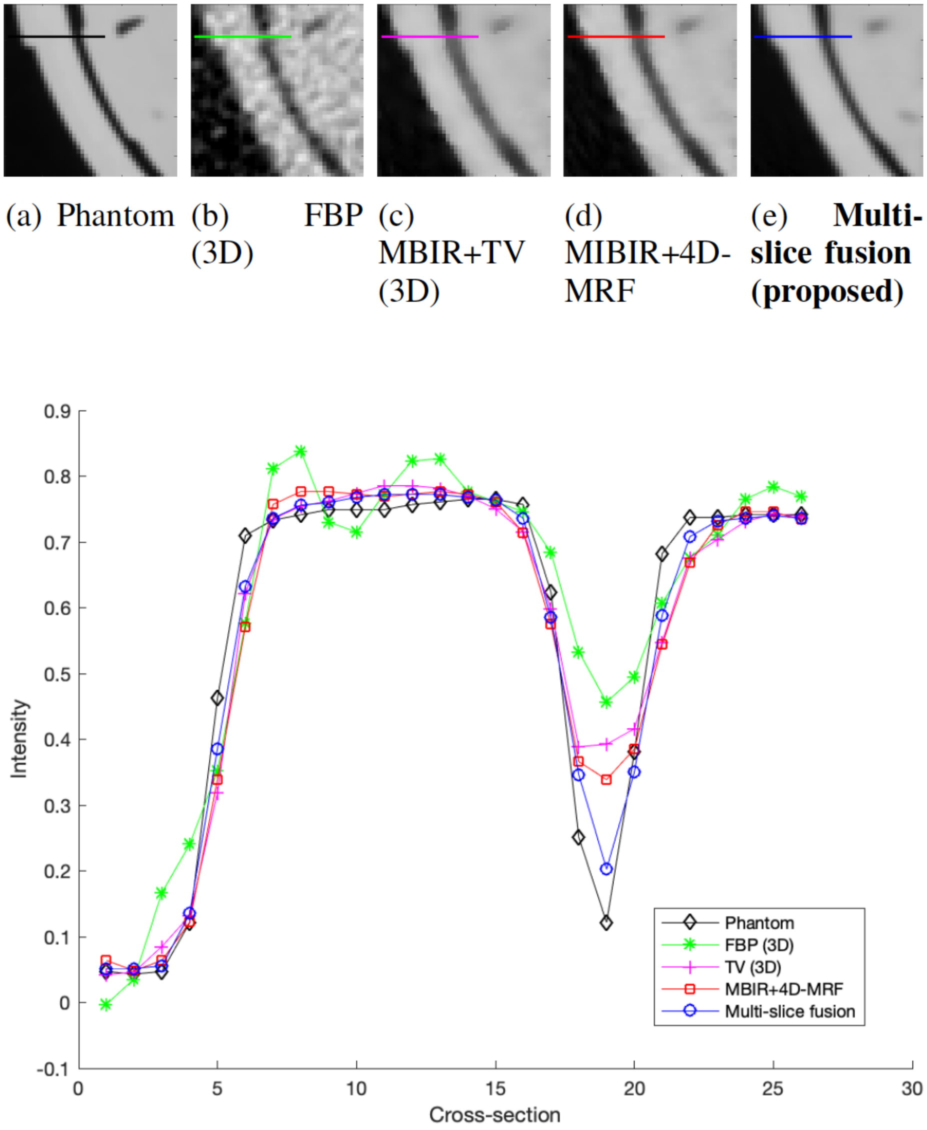
is representative of a variety of Non-Destructive Evaluation (NDE) problems in which the objects to be imaged are constructed from a relatively small number of distinct materials. The extracted patches were normalized to $[0, 1]$ and random rotation, mirroring, intensity shift were applied. The standard deviation of the additive white Gaussian noise added during training was 0.1.

2.7.1 Simulated Data 360°

Table 2.2. Experimental specifications for Simulated Data 360°

Magnification	5.57
Number of Views per Time-point	75
Rotation per Time-point	360°
Cropped Detector Array	240×28 , $(0.95\text{ mm})^2$
Voxel Size	$(0.17\text{ mm})^3$
Reconstruction Size (x,y,z,t)	$240 \times 240 \times 28 \times 8$

In this section we present results on simulated data to evaluate our method in a sparse-view setting. Each time-point is reconstructed from a sparse set of views spanning 360°. We take a low-noise CT reconstruction of a bottle and screw cap and denoise it further using BM4D [49] to generate a clean 3D volume to be used as a 3D phantom. We then vertically translate the 3D phantom by one pixel per time-point to generate a 4D phantom x^0. We generate simulated sinogram measurements as $\mathcal{N}(Ax^0, \Lambda^{-1})$ where A is the projection matrix and the inverse covariance matrix $\Lambda = \text{diag}\{c \exp\{-Ax^0\}$ accounts for the non-uniform noise variance due to a Gaussian approximation [30] of the underlying Poisson noise. We then



(a) Phantom (b) FBP (3D) (c) MBIR+TV (3D) (d) MIBIR+4D-MRF (e) **Multi-slice fusion (proposed)**

(f) Plot of cross-section

Figure 2.8. Plot of cross-section through the phantom and reconstructions from simulated data 360°. Multi-slice fusion results in the most accurate reconstruction of the gap between materials.

perform a 4D reconstruction from the simulated sinogram data and compare with the 4D phantom. The experimental specifications are summarized in Table 2.2.

Figure 2.7 compares reconstructions using multi-slice fusion with several other methods. Each image is a slice through the reconstructed object for one time-point along the spatial xy-plane. The reconstruction using FBP suffers from high noise and fails to recover the small hole in the bottom of the image. The reconstructions using MBIR+TV and MBIR+4D-MRF suffer from jagged edges and fail to recover the small hole in the bottom of the image. MBIR+$H_{\text{yz,t}}$ and MBIR+$H_{\text{zx,t}}$ suffer from horizontal and vertical streaks, respectively, since the denoisers were applied in those planes. MBIR+$H_{\text{xy,t}}$ does not suffer from streaks in the figure since we are viewing a slice along the xy-plane, but it suffers from other artifacts. MBIR+$H_{\text{xy,t}}$ cannot reconstruct the small hole in the bottom of the image since the xy-plane does not contain sufficient information. It is to be noted that multi-slice fusion enhances the size and contrast of the small hole highlighted by the blue circle relative to the phantom. This can cause deviations when measuring the size of small features in the reconstruction.

Next we plot a cross-section through the object for multi-slice fusion, MBIR+4D-MRF, MBIR+TV, FBP, and the phantom in Figure 2.8. Multi-slice fusion results in the most accurate reconstruction of the gap between materials.

Finally we report the peak signal to noise ratio (PSNR) and the structural similarity index measure (SSIM) [61] with respect to the phantom for each method in Table 2.3 to objectively measure image quality. We define the PSNR for a given 4D reconstruction x with a phantom x^0 as

$$\text{PSNR}(x) = 20 \log_{10} \left(\frac{\text{Range}(x^0)}{\text{RMSE}(x, x^0)} \right), \tag{2.15}$$

where range is computed from the 0.1st and 99.9th percentiles of the phantom. As can be seen from Table 2.3, multi-slice fusion results in the highest PSNR and SSIM scores.

Table 2.3. Quantitative Evaluation for simulated data 360°. Multi-slice fusion has the highest PSNR and SSIM metric among all the methods.

Method	PSNR(dB)	SSIM
FBP	19.69	0.609
MBIR+TV	26.63	0.860
MBIR+4D-MRF	25.84	0.787
Multi-slice fusion	**29.07**	**0.943**
MBIR+$H_{\text{xy,t}}$	29.03	0.922
MBIR+$H_{\text{yz,t}}$	28.04	0.932
MBIR+$H_{\text{zx,t}}$	28.31	0.926

2.7.2 Simulated Data 90°

Table 2.4. Experimental specifications for Simulated Data 90°

Magnification	5.57
Number of Views per Time-point	36
Rotation per Time-point	90°
Cropped Detector Array	240×28 , $(0.95\text{ mm})^2$
Voxel Size	$(0.17\text{ mm})^3$
Reconstruction Size (x,y,z,t)	$240 \times 240 \times 28 \times 8$

In this section we present results on simulated data to evaluate our method in a sparse-view and limited-angle setting. Each time-point is reconstructed from a sparse set of views spanning 90°. The simulated measurement data is generated in a similar fashion as Section 2.7.1 using the experimental specifications summarized in Table 2.4.

Figure 2.9 shows a comparison of different methods for simulated data with 90° rotation of object per time-point. The FBP reconstruction has severe limited-angle artifacts. MBIR+TV improves the reconstruction in some regions but it suffers in areas affected by limited angular information. MBIR+4D-MRF reduces limited-angle artifacts, but allows severe artifacts to form that are not necessarily consistent with real 4D image sequences. In contrast, the multi-slice fusion result does not suffer from major limited-angle artifacts.

Table 2.5 shows peak signal to noise ratio (PSNR) and structural similarity index measure (SSIM) with respect to the phantom for each method. Multi-slice fusion results in the highest PSNR and SSIM scores.

Table 2.5. Quantitative Evaluation for simulated data 90°. Multi-slice fusion has the highest PSNR and SSIM metric among all the methods.

Method	PSNR(dB)	SSIM
FBP	10.86	0.467
MBIR+TV	15.35	0.801
MBIR+4D-MRF	14.25	0.742
Multi-slice fusion	**19.44**	**0.875**

In order to determine the effectiveness of our method for more challenging data, we generate extreme sparse-view simulated data with different angle of rotation per time-point while keeping the rest of the experimental setup the same as Table 2.4. Figure 2.10 illustrates the reconstruction quality obtained for the extreme sparse-view data with different levels of limited angle. FBP results in strong artifacts due to sparse-views and limited angles. MBIR+TV and MBIR+4D-MRF mitigates most of the major sparse-view artifacts but suffers from limited angle artifacts in the 90° limited angle case. Multi-slice fusion results in fewer limited-angle and sparse-view artifacts and an improved PSNR metric. Moreover, multi-slice fusion results in a reduced motion and sparse view artifacts as compared to MBIR+TV and MBIR+4D-MRF as the rotation per time point is decreased.

2.7.3 Real Data 360°: Vial Compression

Table 2.6. Experimental specifications for Real Data 360°: Vial Compression

Scanner Model	North Star Imaging X50
Voltage	140 kV
Current	500 μA
Exposure	20 ms
Source-Detector Distance	839 mm
Magnification	5.57
Number of Views per Time-point	150
Rotation per Time-point	360°
Cropped Detector Array	731×91, $(0.25\text{ mm})^2$
Voxel Size	$(0.0456\text{ mm})^3$
Reconstruction Size (x,y,z,t)	$731 \times 731 \times 91 \times 16$

In this section we present results on real data to evaluate our method in a sparse-view setting. The data is from a dynamic cone-beam X-ray scan of a glass vial, with elastomeric stopper and aluminum crimp-seal, using a North Star Imaging X50 X-ray CT system. The experimental specifications are summarized in Table 2.6.

The vial is undergoing dynamic compression during the scan, to capture the mechanical response of the components as shown in Figure 2.15. Of particular interest is the moment when the aluminum seal is no longer in contact with the underside of the glass neck finish. This indicates the moment when the force applied exceeds that exerted by the rubber on the glass; this is known as the “residual seal force” [62].

During the scan, the vial was held in place by fixtures that were placed out of the field of view as shown in Figure 2.15. As the object rotated, the fixtures periodically intercepted the path of the X-rays resulting in corrupted measurements and consequently artifacts in the reconstruction. To mitigate this problem, we incorporate additional corrections that are described in Appendix A.

Figure 2.11 compares multi-slice fusion with several other methods. Each image is a slice through the reconstructed vial for one time-point along the spatial xy-plane. Both FBP and MBIR+4D-MRF suffer from obvious artifacts, higher noise and blurred edges. In contrast to that, the multi-slice fusion reconstruction has smooth and uniform textures while preserving edge definition. Figure 2.11 also illustrates the effect of model fusion by comparing multi-slice fusion with MBIR+$H_{\mathrm{xy,t}}$, MBIR+$H_{\mathrm{yz,t}}$, and MBIR+$H_{\mathrm{zx,t}}$. MBIR+$H_{\mathrm{yz,t}}$ and MBIR+$H_{\mathrm{zx,t}}$ suffer from horizontal and vertical streaks respectively since the denoisers were applied in those planes. MBIR+$H_{\mathrm{xy,t}}$ does not suffer from streaks in the figure since we are viewing a slice along the xy-plane, but it suffers from other artifacts. MBIR+$H_{\mathrm{xy,t}}$ cannot reconstruct the outer ring since the slice displayed is at the edge of the aluminum seal and the xy-plane does not contain sufficient information. In contrast, multi-slice fusion can resolve the edges of the rings better than either of MBIR+$H_{\mathrm{xy,t}}$, MBIR+$H_{\mathrm{yz,t}}$, and MBIR+$H_{\mathrm{zx,t}}$ since it uses information from all the spatial coordinates.

Next, we plot a cross-section through the object for multi-slice fusion, MBIR+4D-MRF and FBP in Figure 2.12. For this, we choose a time-point where we know the aluminum and glass have separated spatially, thus creating an air-gap. Multi-slice fusion results in a deeper and more defined reconstruction of the gap between materials. This supports that multi-slice fusion is able to preserve fine details in spite of producing a smooth regularized image.

Finally in Figure 2.13 we plot a cross-section through the object with respect to time to show the improved space-time resolution of our method. We do this for FBP, MBIR+4D-MRF and multi-slice fusion. Multi-slice fusion results in improved space-time resolution of the separation of aluminum and glass.

2.7.4 Real Data 90°: Injector Pen

Table 2.7. Experimental specifications for Real Data 90°: Injector Pen

Scanner Model	North Star Imaging X50
Voltage	165 kV
Current	550 μA
Exposure	12.5 ms
Source-Detector Distance	694 mm
Magnification	2.83
Number of Views per Time-point	36
Rotation per Time-point	90°
Cropped Detector Array	263 × 768, $(0.254\text{ mm})^2$
Voxel Size	$(0.089\text{ mm})^3$
Reconstruction Size (x,y,z,t)	263 × 263 × 778 × 12

In this section we present results on real data to evaluate our method in a sparse-view and limited-angle setting. The data is from a dynamic cone-beam X-ray scan of an injector pen using a North Star Imaging X50 X-ray CT system. The experimental specifications are summarized in Table 2.7.

The injection device is initiated before the dynamic scan starts and completes a full injection during the duration of the scan. We are interested in observing the motion of a particular spring within the injector pen in order to determine whether it is working as expected. The spring in question is a non-helical wave-spring [63] that is constructed out of circular rings that are joined together. The spring exhibits a fast motion and as a result we need a high temporal resolution to observe the motion of the spring. To have sufficient temporal resolution we reconstruct one frame for every 90° rotation of the object instead of the conventional 360° rotation.

Figure 2.14 shows a volume rendering of the reconstructed spring and a cross-section through it for four time-points and reconstruction methods FBP, MBIR+4D-MRF, and multi-slice fusion. The FBP reconstruction contains severe limited-angle artifacts. MBIR+4D-MRF mitigates some limited-angle artifacts but some artifacts remain. In contrast, multi-slice fusion mitigates most limited-angle artifacts. The cross-sections of the spring in the multi-slice fusion reconstruction are more circular than the other methods, which align with our prior knowledge about the spring. The fast compression of the spring causes the rings within the spring to move significantly within a time-point, resulting in the observed blur in the multi-slice fusion reconstruction. Strong limited-angle artifacts in the other reconstructions mask this effect.

2.8 Conclusion

In this paper, we proposed a novel 4D X-ray CT reconstruction algorithm, multi-slice fusion, that combines multiple low-dimensional denoisers to form a 4D prior. Our method allows the formation of an advanced 4D prior using state-of-the-art CNN denoisers without needing to train on 4D data. Furthermore, it allows for multiple levels of parallelism, thus enabling reconstruction of large volumes in a reasonable time. Although we focused on 4D X-ray CT reconstruction for NDE applications, our method can be used for any reconstruction problem involving multiple dimensions.

Acknowledgment

The authors would like to acknowledge support from Eli Lilly and Company under research project funding agreement 17099289. Charles A. Bouman and Gregery T. Buzzard were supported in part by NSF grant CCF-1763896. We also thank M. Cory Victor and Dr. Coralie Richard from Eli Lilly and Company for their assistance and guidance in setting up the residual seal force test experiment.

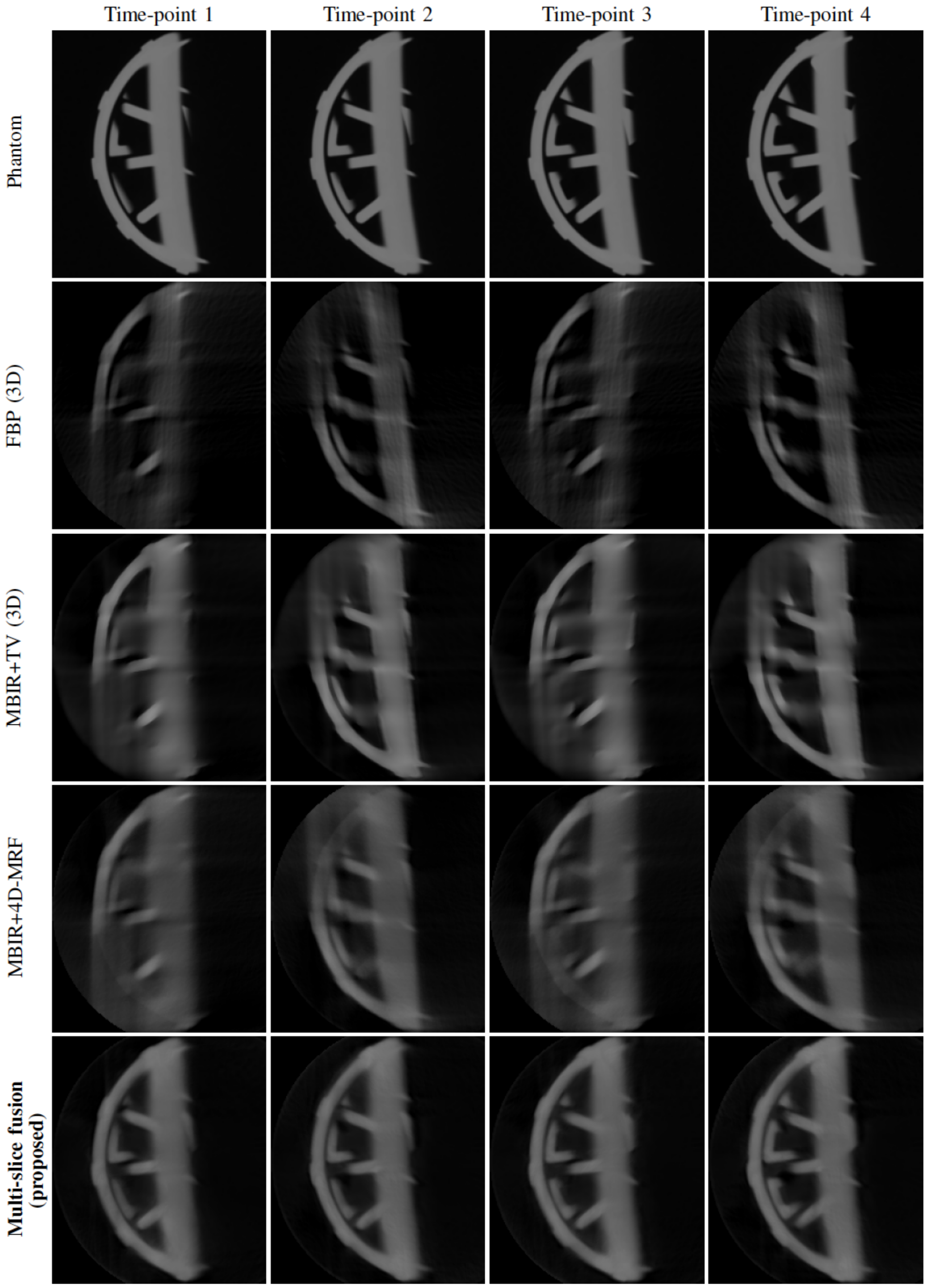


Figure 2.9. Comparison of different methods for simulated data with 90° rotation of object per time-point. The FBP reconstruction has severe limited-angle artifacts. MBIR+TV improves the reconstruction in some regions but it suffers in areas affected by limited angular information. MBIR+4D-MRF reduces limited-angle artifacts, but allows severe artifacts to form that are not necessarily consistent with real 4D image sequences. In contrast, the multi-slice fusion result does not suffer from major limited-angle artifacts.

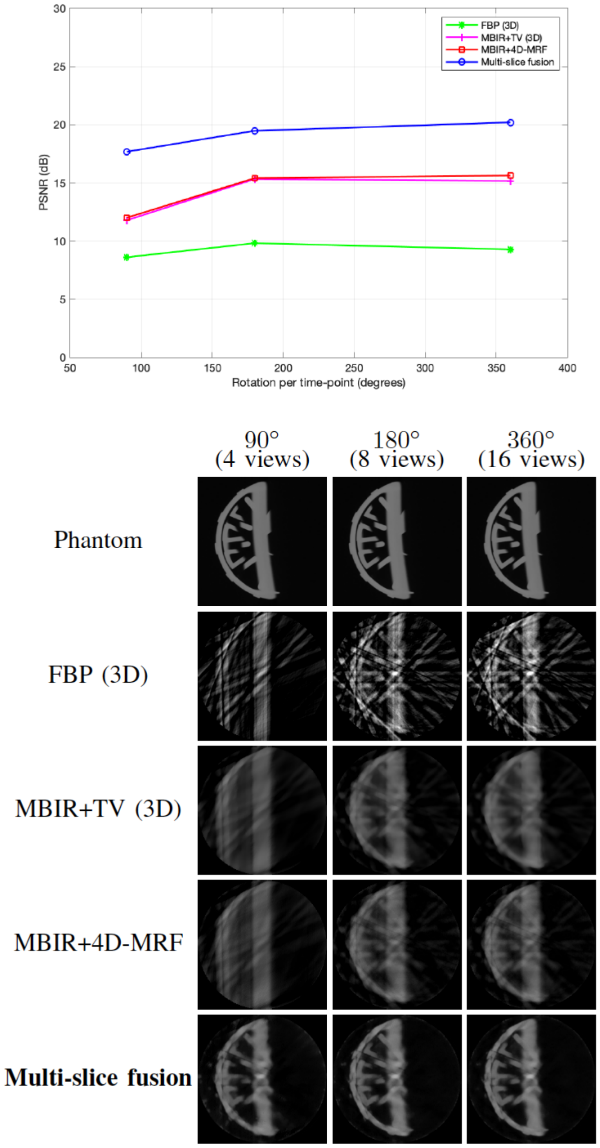
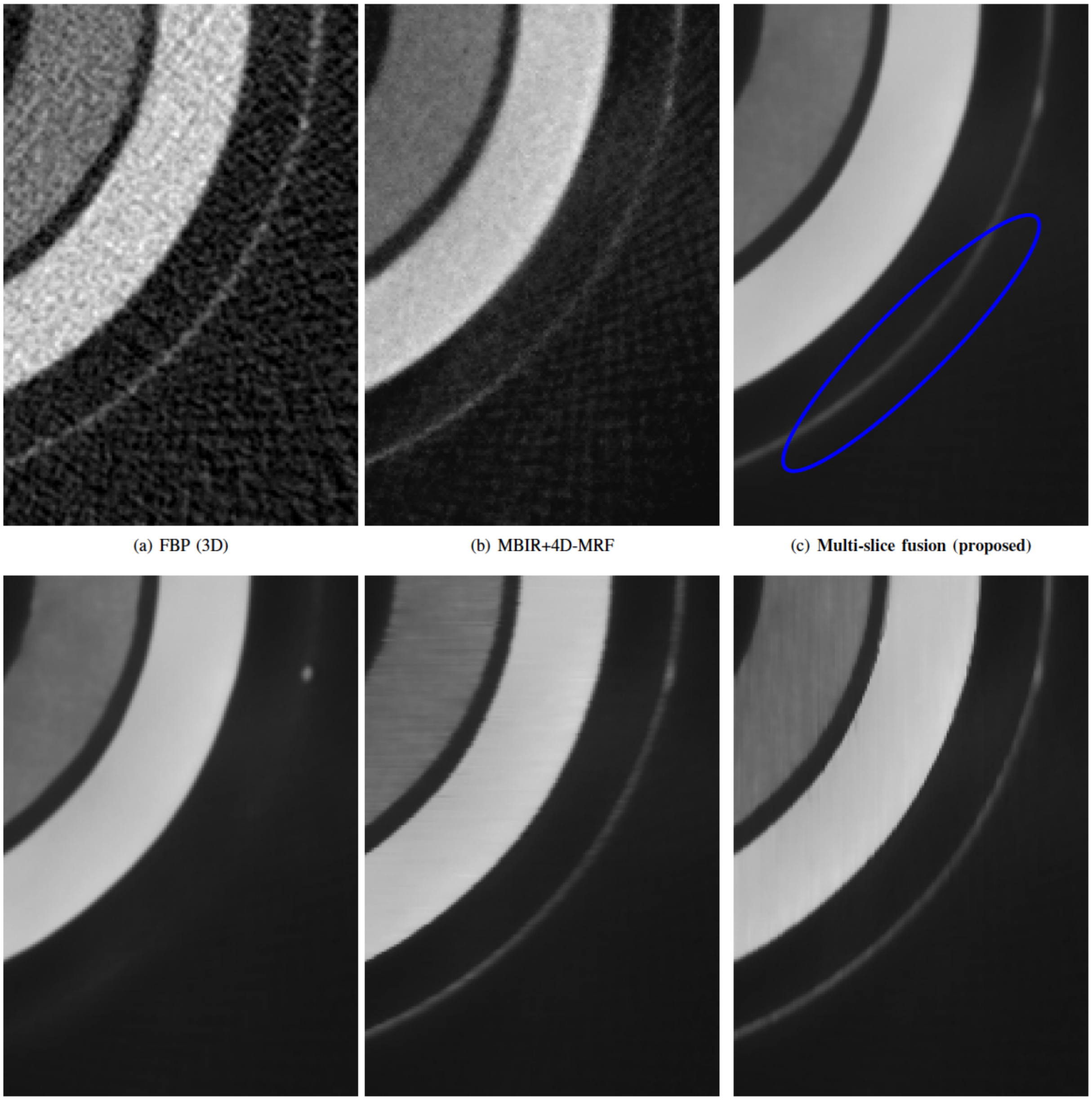


Figure 2.10. Illustration of the reconstruction quality obtained for extreme sparse-view data with different levels of limited angle per time-point. FBP results in strong artifacts due to sparse-views and limited angles. MBIR+TV and MBIR+4D-MRF mitigates most of the major sparse-view artifacts but suffers from limited angle artifacts in the 90° limited angle case. Multi-slice fusion results in fewer limited-angle and sparse-view artifacts and an improved PSNR metric. Moreover, multi-slice fusion results in reduced artifacts compared to MBIR+TV and MBIR+4D-MRF as the rotation per time point is decreased.



(a) FBP (3D)

(b) MBIR+4D-MRF

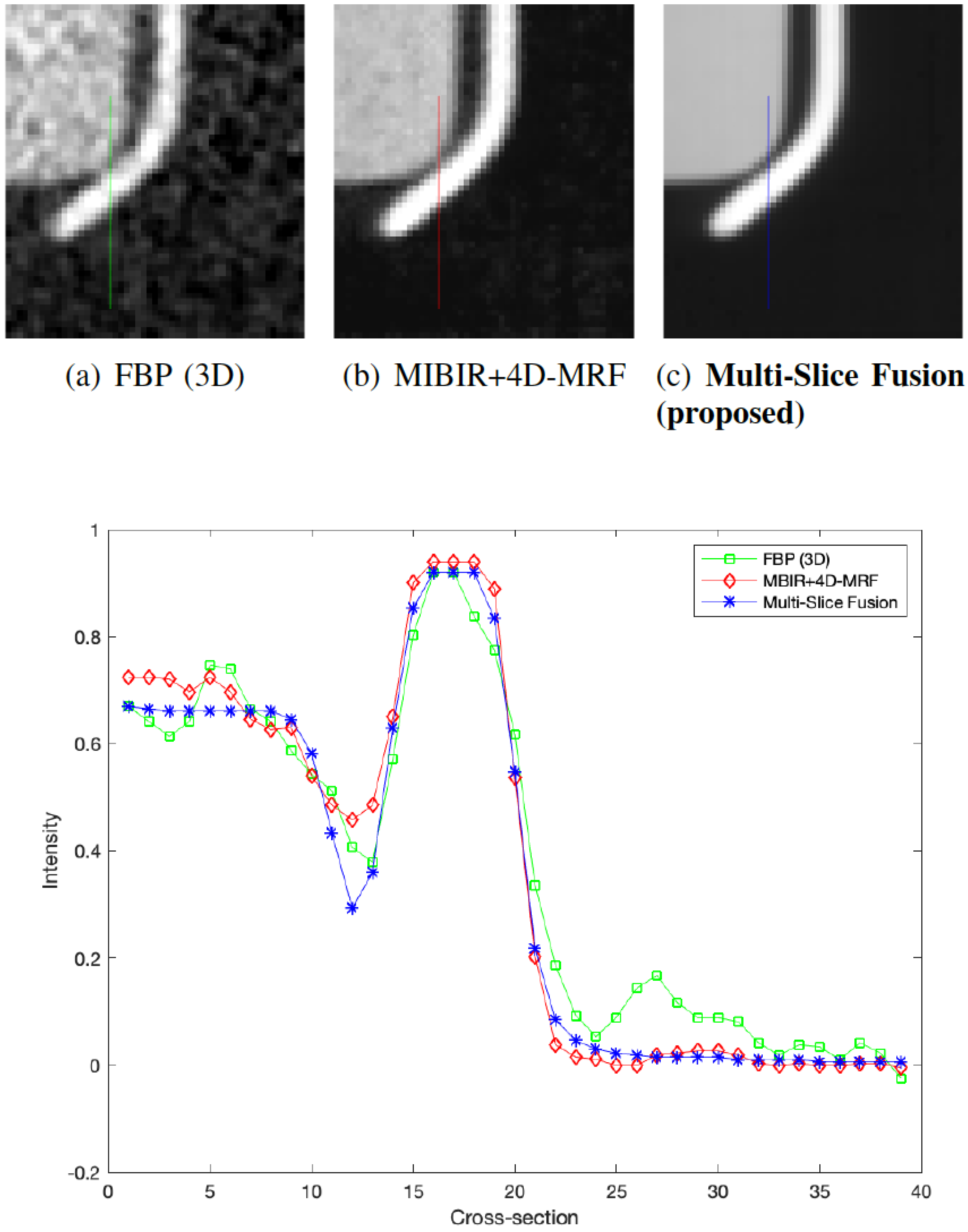
(c) **Multi-slice fusion (proposed)**

(d) MBIR+$H_{\mathrm{xy,t}}$ (Missing Feature)

(e) MBIR+$H_{\mathrm{yz,t}}$ (Horizontal Streaks)

(f) MBIR+$H_{\mathrm{zx,t}}$ (Vertical Streaks)

Figure 2.11. Comparison of different methods for Real Data 360°: vial. Each image is a slice through the reconstructed vial for one time-point along the spatial xy-plane. Both FBP and MBIR+4D-MRF suffer from obvious windmill artifacts, higher noise and blurred edges. In contrast to that, the multi-slice fusion reconstruction has smooth and uniform textures while preserving edge definition. MBIR+$H_{\mathrm{yz,t}}$ and MBIR+$H_{\mathrm{zx,t}}$ suffer from horizontal and vertical streaks. MBIR+$H_{\mathrm{xy,t}}$ cannot reconstruct the outer ring since the slice displayed is at the edge of the aluminum seal and the xy-plane does not contain sufficient information. Multi-slice fusion can resolve the edges of the rings better than either of MBIR+$H_{\mathrm{xy,t}}$, MBIR+$H_{\mathrm{yz,t}}$, and MBIR+$H_{\mathrm{zx,t}}$ since it has information from all the spatial coordinates.



(a) FBP (3D)

(b) MIBIR+4D-MRF

(c) **Multi-Slice Fusion (proposed)**

(d) Plot of cross-section

Figure 2.12. Plot of cross-section through the vial at a time when the aluminum and glass have physically separated. Multi-slice fusion is able to resolve the junction between materials better while simultaneously producing a smoother reconstruction within materials compared to MBIR+4D-MRF and FBP.

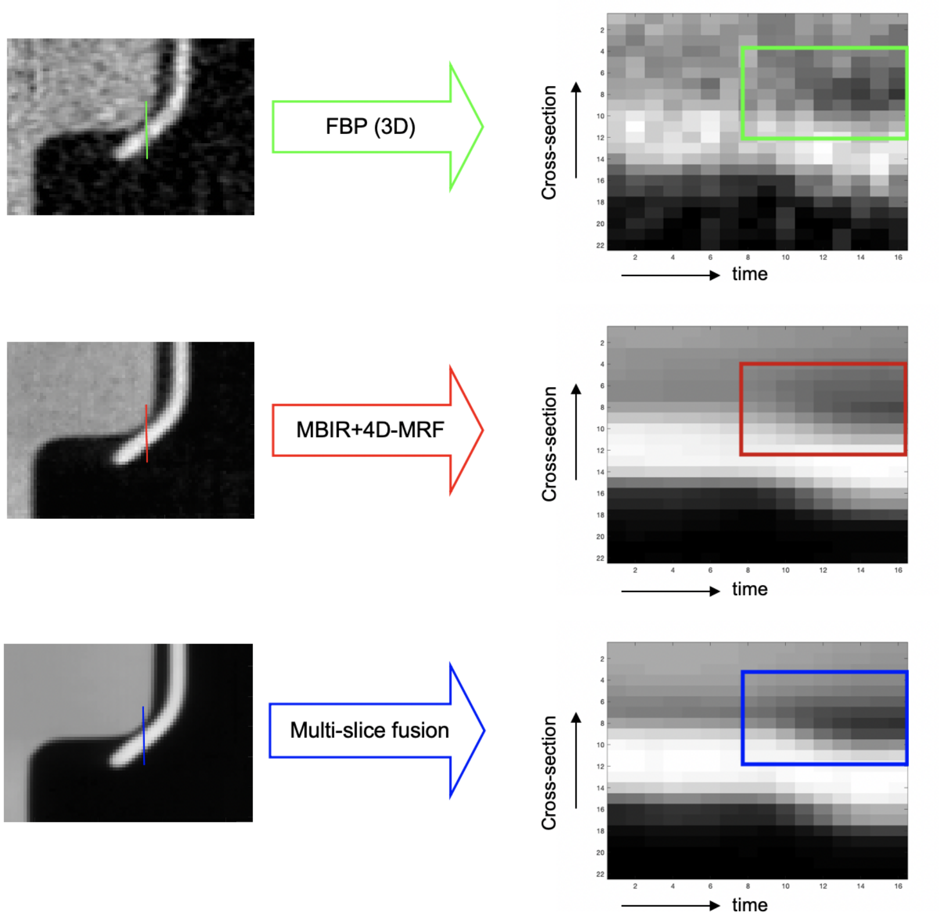


Figure 2.13. Illustration of temporal resolution for real data 360° : vial. We plot a cross-section through the vial with time for each method: multi-slice fusion, MBIR+4D-MRF, FBP. Multi-slice fusion results in improved space-time resolution of the separation of aluminum and glass.

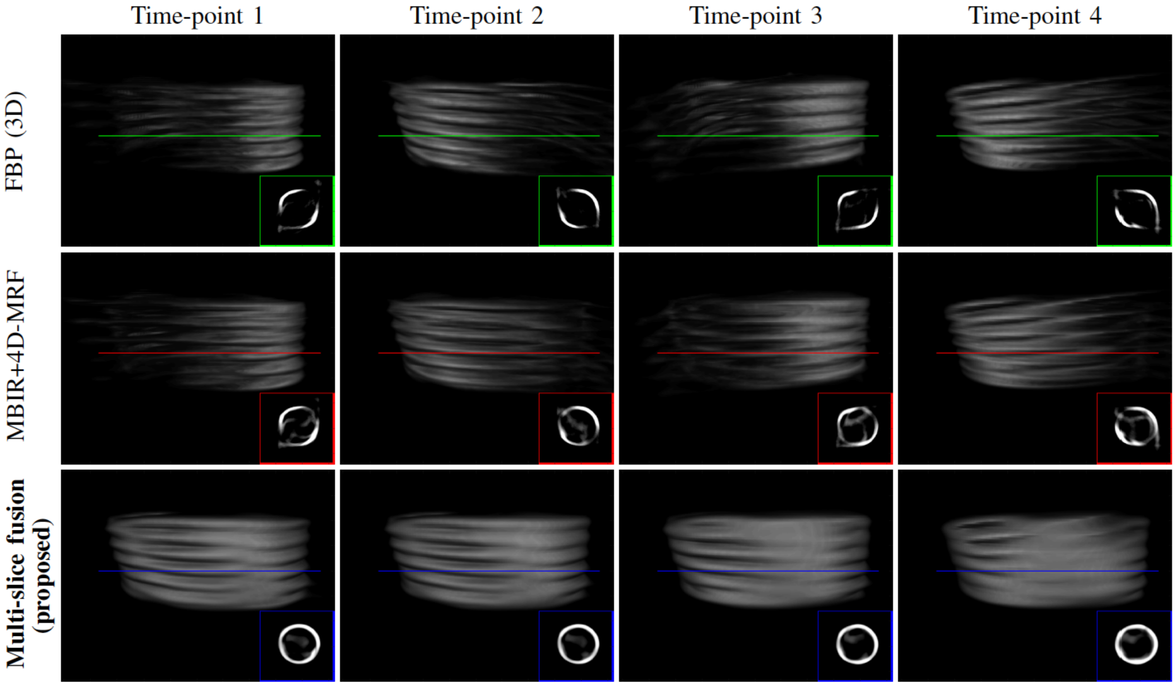


Figure 2.14. Volume rendering of the reconstructed spring and its cross-section for four time-points. A 90° limited set of views is used to reconstruct each time-point. The FBP reconstruction contains severe limited-angle artifacts. MBIR+4D-MRF mitigates some limited-angle artifacts but some artifacts remain.

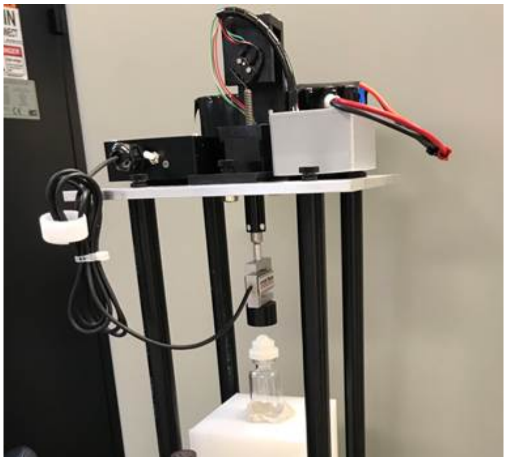


Figure 2.15. Experimental setup for Real Data 360°: Vial Compression. The vial is undergoing dynamic compression during the scan, to capture the mechanical response of the components. The glass vial (center) and the actuator (top) are held together by a frame constructed of tubes and plates. The tubes were placed outside the field of view of the CT scanner, thus causing artifacts in the reconstruction. We describe a correction for this in Appendix A.

3. CODEX: A MODULAR FRAMEWORK FOR JOINT TEMPORAL DE-BLURRING AND TOMOGRAPHIC RECONSTRUCTION

3.1 Introduction

Computed tomography (CT) imaging has been widely used in a variety of applications to study the internal structure of static and dynamic objects. For rapidly changing objects, good temporal resolution is crucial in order to resolve the reconstructed object accurately. Model based iterative reconstruction techniques have led to significant improvements in temporal resolution for time-resolved CT through the use of novel view-sampling [33] and improved prior modeling **multi**, [33], [35], [64]. However, even with these improvements, the temporal resolution is fundamentally limited by the rate of collection of projection measurements.

In an ideal CT system, each view is assumed to measure the projection at a single angle. A step-and-shoot acquisition strategy conforms with this assumption by rotating the object to each desired angle and stopping the rotation before taking a measurement. Step-and-shoot results in a slow acquisition due to the frequent stops in rotation and a wastage of the photon flux when the object is being rotated. A more practical acquisition strategy is fly-scanning [65], where the object rotates continuously while measurements are being taken. However, this results in motion blurred measurements, and consequently reconstructions that are affected by motion artifacts.

The rotation of the object introduces a motion blur in the measurements that is difficult to deblur directly using simple approaches. Simple deconvolution operations [66], [67] along the view-angle dimension are not possible when the view angles are sparse which is typical in time-resolved CT imaging. In the photography literature [68], [69], motion blurred photographs are often deblurred by modeling the motion blur as a spatial convolution with a point-spread function and then inverting the convolution. However, the blurring operation in the radiographs cannot be modeled as a spatial convolution. This is because each X-ray detector pixel measures the combined signal from many parts of the object with a differ-

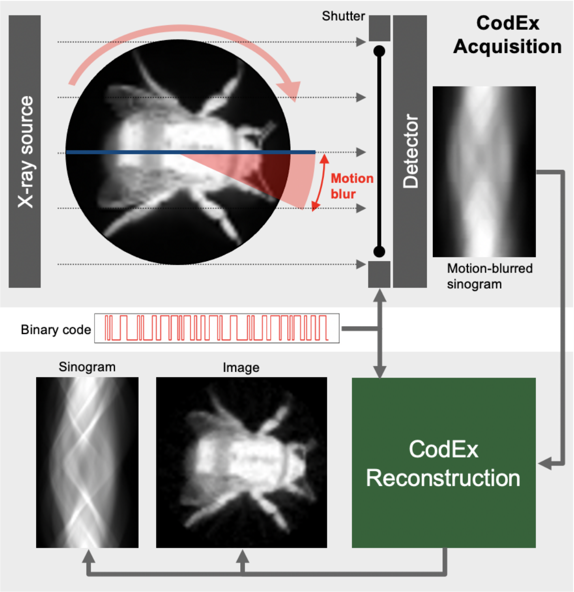


Figure 3.1. Illustration of our method, CodEx. CodEx is a synergistic combination of a coded acquisition and a non-convex Bayesian reconstruction. During acquisition, CodEx flutters the exposure on and off rapidly in a known binary code during each view, resulting in a coded motion blur. CodEx subsequently uses the knowledge of the code to solve a non-convex iterative reconstruction problem to reconstruct the underlying object.

ent velocity vector due to the rotation. Consequently, spatial deblurring approaches from photography cannot be used to deblur the motion blurred 2D radiographs.

Model based iterative reconstruction methods can be used to model the motion blur within the reconstruction problem and solve it. However, for the problem of transmission tomography, the attenuation of the X-ray beam is exponential, so this results in a non-convex cost function [70] that is difficult to optimize. Some approaches use linear approximations to the forward model [71], [72] in order to make the computation simpler, but the approximation error can be too large in some circumstances [71]. Regardless of the method used to invert the motion blurred measurements, the inversion process is inherently ill-posed due to the nature of the motion blur [71].

In this paper, we introduce CodEx, a novel method for coded exposure CT acquisition and reconstruction. Figure 3.1 illustrates our approach. During acquisition, CodEx flutters the exposure on and off rapidly in a known binary code during each view, resulting in a coded motion blur. CodEx subsequently uses the knowledge of the code to solve a non-convex iterative reconstruction problem to reconstruct the underlying object. The CodEx reconstruction method uses the alternating direction method of multipliers (ADMM) to split the inverse problem into iterative deblurring and CT reconstruction sub-problems. This results in a modular reconstruction algorithm, making it easy to extend to different CT geometries.

We present results on different simulated as well as experimental data to evaluate our method, as well as to answer questions about the choice of a good code. The results indicate that a well chosen code results in a reduced loss of information during acquisition, leading to improved reconstruction quality.

The rest of the paper is organized as follows. In section 3.2, we introduce the general problem of motion blur due to integration over angles, as well as coded acquisition in order to mitigate the effect of the motion blur. In section 3.3, we introduce our CodEx formulation that models the coded motion blur and inverts it during the reconstruction process. In section 3.4, we describe our interlaced view-angle sampling scheme that provides a practical approach for performing coded exposure acquisition with interlaced view angles. Finally, in section 3.5, we present results on simulated and experimental data to evaluate our method.

3.2 Problem Description

In a conventional computed tomography (CT) setting, a single view is assumed to measure the projection of an object at a single angle. To conform with this assumption, a step-and-shoot scanning is done where the object is rotated to each desired angle, a view measurement is taken, and the rotation is resumed. However, this leads to slow acquisition and wasted photons. A more practical approach is fly-scanning, where the object is continuously rotated while the view measurements are being taken. However, in this case, the detector integrates the incoming photon-flux over a range of angles instead of a single angle, as shown in Figure 3.2. The resulting vector of M_d expected photon-counts at the detector at angle θ_0 can be written as

$$\bar{\lambda}^{\text{boxcar}}_{\theta_0} = \frac{\lambda^0}{\Delta\theta} \int_{\theta_0}^{\theta_0+\Delta\theta} \exp\left\{-A_\theta\, x\right\} d\theta \ , \tag{3.1}$$

where $A_\theta \in \mathbb{R}^{M_d \times N}$ performs the forward projection of the image x at angle θ, λ^0 is the photon-flux of the X-ray source, and $\Delta\theta$ is the blur-angle.

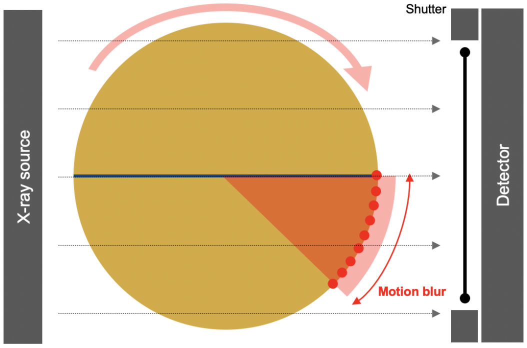


Figure 3.2. Illustration of motion blur due to rotation. In step-and-shoot acquisition, each measurement depends on the projection at a single angle (blue line). In fly-scanning, the detector integrates photons over a range of projection angles (red sector). This integration can be approximated as a discrete sum over micro-projection angles (red dots).

We can approximate the integration in equation (3.1) using a discrete sum over K closely spaced angles as

$$\bar{\lambda}^{\text{boxcar}}_{\theta_0} = \frac{\lambda^0}{K} \sum_{k=0}^{K-1} \exp\left\{-A_{\theta_0 + \frac{k\Delta\theta}{K}}\, x\right\} \ . \tag{3.2}$$

We will refer to these K closely spaced angles as micro-projection angles henceforth.

As $\Delta\theta$ becomes larger, we collect more photons in a single measurement, but this comes at the cost of blurred measurements, which leads to a reconstruction with motion artifacts. To overcome this problem, our approach will be to modulate the photon flux at each of the K micro-projection angles by a binary code. This can be done by pulsating the X-ray source [73], using the electronic shutter of the detector sensor, or with a mechanical shutter on the beam path [74] as shown in Figure 3.2. In this case, the expected photon-counts at the detector at angle θ_0 can be written as

$$\bar{\lambda}_{\theta_0} = \frac{\lambda^0}{K} \sum_{k=0}^{K-1} c_k \exp\left\{ -A_{\theta_0 + \frac{k\Delta\theta}{K}}\, x \right\} \ , \tag{3.3}$$

where $c = [c_0, c_1, \cdots, c_{K-1}]$ is the binary code used to modulate the photons. In this case, each measurement is formed by a coded sum over K non-overlapping micro-projection angles. Notice that equation (3.3) reduces to the boxcar case of equation (3.2) when $c = [1, 1, \cdots, 1]$. On the other hand, equation (3.3) reduces to step-and-shoot scanning when $c = [1, 0, \cdots, 0]$. Modulating the photon flux results in loss of photons compared to the boxcar case of $c = [1, 1, \cdots, 1]$, but many more photons are collected relative to the step-and-shoot case of $c = [1, 0, \cdots, 0]$. A good choice of code c can result in an invertible blur while improving the signal to noise ratio (SNR) of the measurements.

3.3 CodEx Formulation

In this section, we will introduce CodEx, a synergistic combination of coded acquisition and CT reconstruction. During the acquisition process, we collect M_θ 2D radiograph measurements at M_θ different measurement angles. As Figure 3.2 illustrates, each radiograph measures the projection of the object across a range of angles which can be written as a function of K micro-projections. Some of the M_θ measurements can have overlapping projections and share the same micro-projections. Without loss of generality, we assume that there are $N_\theta \leq KM_\theta$ unique micro-projection angles in $[0, \pi]$ out of the KM_θ maximum possible micro-projection angles.

3.3.1 Measurement Model

For each measurement, we use a binary code $c = [c_0, c_1, \cdots, c_{K-1}]$ to modulate the photon-flux over K micro-projection angles as shown in equation (3.3). The resulting vector of expected photon counts for all the measurement angles can be written as

$$\bar{\lambda} = \bar{c}\frac{\lambda^0}{K} C \exp\{-Ax\} \ , \tag{3.4}$$

where, $\bar{\lambda} \in \mathbb{R}^{M_\theta M_d}$ is the vector of expected photon counts for the M_θ measurement angles and M_d detector pixels, $C \in \mathbb{R}^{M_\theta M_d \times N_\theta M_d}$ is a sparse matrix that performs the coded sum in equation (3.3) and is normalized such that each row of C sums to 1, $\bar{c} = \sum_{k=0}^{K-1} c_k$ is the normalizing constant, $A \in \mathbb{R}^{N_\theta M_d \times N}$ projects the image $x \in \mathbb{R}^N$ for the N_θ unique micro-projection angles. The structure of the matrix C depends on how the M_θ measurement angles are arranged, and how each measurement relates to the micro-projections at K micro-projection angles. Section 3.4 provides details on the structure of C for a practical interlaced view-sampling strategy.

The vector of incident photon counts at the detector, $\lambda \in \mathbb{R}^{M_d M_\theta}$ are given by

$$\lambda \sim \mathrm{Pois}(\bar{\lambda}) \ , \tag{3.5}$$

where $\mathrm{Pois}(\bar{\lambda})$ denotes an element wise Poisson distribution with mean $\bar{\lambda}$. A high expected photon-count $\bar{\lambda}$ results in a higher signal to noise ratio (SNR) in the measurements. The goal of the coded projections is to increase the SNR of the acquired data while also allowing for accurate deblurring of the summed micro-projections. A good code c introduces an invertible blur and allows us to preserve high frequency information while at the same time collecting more photons to reduce noise [69].

In order to derive the forward model, we first convert the photon-count measurements into projection measurements as is typically done in tomography. In order to do this, we normalize by the "blank scan" obtained when the object is removed (i.e., $x = 0$), and we take the negative log to form

$$y = -\log\left\{\frac{\lambda}{\bar{c}\lambda^0/K}\right\} \ , \tag{3.6}$$

where $y \in \mathbb{R}^{M_d M_\theta}$ is the vector of projection measurements for all M_θ views, and $\bar{c} = \sum_{i=0}^{K-1} c_k$ results from the assumption that $x = 0$ in equation (3.3). If the photon counts are large, then we can make the approximation [30], [34] that

$$\begin{aligned} E\left[y|x\right] &\approx -\log\left\{C\exp\left\{-Ax\right\}\right\} \\ Var\left[y|x\right] &\approx D^{-1} \ . \end{aligned} \tag{3.7}$$

where $D = \text{diag}\{\lambda\}$. In practice, the true photon counts, λ are often unknown. Consequently, we set the precision matrix D as

$$D = \text{diag}\{w\exp\left\{-y\right\}\} \ , \tag{3.8}$$

where the scalar w is empirically chosen [33], [34], [75].

Using the Gaussian approximation to the likelihood function[30], the non-linear forward model is then given by

$$-\log p(y|x) = \frac{1}{2}\left\|y + \log\left\{C\exp\left\{-Ax\right\}\right\}\right\|_D^2 + \text{const}(y) \ . \tag{3.9}$$

3.3.2 MAP Estimate

The X-ray attenuation coefficient image x^* can be reconstructed by computing a Maximum A Posteriori (MAP) estimate as

$$\begin{aligned} x^* &= \arg\min_x\left\{-\log p(y|x) - \log p(x)\right\} \\ &= \arg\min_x\left\{\frac{1}{2}\left\|y + \log\left\{C\exp\left\{-Ax\right\}\right\}\right\|_D^2 + h(x)\right\} \end{aligned} \tag{3.10}$$

where $h(x)$ is the regularization or prior model and the forward model term $-\log p(y|x)$ follows from equation (3.9). Notice that the non-linear forward model makes direct optimization of the cost function in equation (3.10) challenging, but in the following sections, we propose a modular algorithm for solving (3.10) that makes the problem tractable.

3.3.3 ADMM Formulation

In order to simplify the optimization in equation (3.10), we split the cost function into two parts with the following constraint

$$p = Ax \ , \tag{3.11}$$

where $p \in \mathbb{R}^{N_\theta M_d}$ is the projection of the image x at the N_θ finely spaced micro-projection angles. In other words, p is the full set of unobserved micro-projections of the object. We thus form the following equivalent problem.

$$\begin{aligned} x^*, p^* =& \arg\min_{x,p} \left\{ \frac{1}{2} \left\| y + \log\left\{ C \exp\left\{ -p \right\} \right\} \right\|_D^2 + h(x) \right\} \\ &\text{s.t } p = Ax \end{aligned} \tag{3.12}$$

Next, we will use the alternating directions method of multipliers (ADMM) method [58] to solve the constrained optimization of equation (3.12). The augmented Lagrangian for this problem is given by

$$\begin{aligned} L(p, x, u) =& \frac{1}{2} \left\| y + \log\left\{ C \exp\left\{ -p \right\} \right\} \right\|_D^2 + h(x) \\ &+ \frac{1}{2\sigma^2} \left\| p - A\,x + u \right\|^2 , \end{aligned} \tag{3.13}$$

where σ is a tunable parameter, and u is the scaled dual variable. The ADMM algorithm for this problem can then be formulated as Algorithm 2.

Algorithm 2: ADMM formulation for coded exposure reconstruction

```
Initialize: $p$, $x$, $u$
while not converged do
    $p \leftarrow \arg\min_p L(p, x, u)$
    $x \leftarrow \arg\min_x L(p, x, u)$
    $u \leftarrow u + p - Ax$
$x^* \leftarrow x$
```

3.3.4 Modular Implementation

Note that the optimization sub-problems in Algorithm 2 can be simplified as

$$\begin{aligned}\arg\min_p L(p,x,u) = \arg\min_p \Big\{ & \frac{1}{2}\left\|y + \log\left\{C\exp\left\{-p\right\}\right\}\right\|_D^2 \\ & + \frac{1}{2\sigma^2}\left\|p-(Ax-u)\right\|^2 \Big\}, \end{aligned} \tag{3.14}$$

$$\begin{aligned}\arg\min_x L(p,x,u) = \arg\min_x \Big\{ & \frac{1}{2\sigma^2}\left\|(p+u)-Ax\right\|^2 \\ & + h(x) \Big\}. \end{aligned} \tag{3.15}$$

We can rewrite the optimization problems in equations (3.14) and (3.15) in a more compact form as

$$\arg\min_p L(p,x,u) = F_d(Ax-u), \tag{3.16}$$

$$\arg\min_x L(p,x,u) = F_t(p+u), \tag{3.17}$$

where the operators F_d and F_t are defined as

$$\begin{aligned}F_d(\tilde{p}) = \arg\min_p \Bigg\{ & \frac{1}{2}\left\|y + \log\left\{C\exp\left\{-p\right\}\right\}\right\|_D^2 \\ & + \frac{1}{2\sigma^2}\left\|p-\tilde{p}\right\|^2 \Bigg\}, \end{aligned} \tag{3.18}$$

$$F_t(\tilde{p}) = \arg\min_x \left\{ \frac{1}{2\sigma^2}\left\|\tilde{p}-Ax\right\|^2 + h(x) \right\}, \tag{3.19}$$

where $\tilde{p}$ is a representative variable.

Both operators F_d and F_t have intuitive interpretations. From its form, function F_d can be interpreted as the MAP deblurring function. Intuitively, F_d computes the MAP estimate of the micro-projections p given the coded blurred measurements y and a prior distribution of $N(\tilde{p}, \sigma^2 I)$. In other words, F_d is a function that recovers the full set of unobserved micro-projections in the proximity of $\tilde{p}$. On the other hand, the function $x = F_t(p)$ has the simple interpretation of being a function that computes the regularized tomographic reconstruction, x, given the micro-projections p.

```
Algorithm 3: CodEx reconstruction algorithm
Initialize: $p$, $x$, $u$
while not converged do
    $p \leftarrow \tilde{F}_d(Ax - u; p)$
    $x \leftarrow \tilde{F}_t(p + u; x)$
    $u \leftarrow u + p - Ax$
$x^* \leftarrow x$
```

Algorithm (3) shows the complete CodEx reconstruction algorithm. Since it is impractical to minimize the functions in equations (3.18) and (3.19) completely, we perform partial updates starting from an initial condition. The operator $\tilde{F}_d(\tilde{p}; p_{\text{init}})$ denotes the computation of $F_d(\tilde{p})$ for a fixed number of partial iterations starting from an initial condition of p_{init}. Similarly, the operator $\tilde{F}_t(\tilde{p}; x_{\text{init}})$ denotes the computation of $F_t(\tilde{p})$ for a fixed number of partial iterations starting from an initial condition of x_{init}.

Algorithm 4 outlines the computation of the deblurring function, $\tilde{F}_d(\tilde{p}; p_{\text{init}})$ that performs a partial update minimization of equation (3.18) starting from an initial value of p_{init}. We use a gradient descent approach with a backtracking line-search [76] to perform the optimization. The cost function is denoted by

$$f_d(p) = \frac{1}{2} \left\| y + \log \left\{ C \exp \left\{ -p \right\} \right\} \right\|_D^2 + \frac{1}{2\sigma^2} \left\| p - \tilde{p} \right\|^2 .$$

The gradient of the cost function $f_d(p)$ is denoted by

$$g = -\text{diag} \left(\exp \left\{ -p \right\} \right) C^\top \text{diag} \left(C \exp \left\{ -p \right\} \right)^{-1} De \; . \tag{3.20}$$

The cost function $f_d(.)$ is evaluated several times for each iteration in order to choose an appropriate step-size of η starting from an initial step-size of η_0.

Algorithm 5 outlines the computation of the tomographic reconstruction function, $\tilde{F}_t(\tilde{p}; x_{\text{init}})$ that performs a partial update minimization of equation (3.19) starting from an initial value

Algorithm 4: Computation of the deblurring function $\tilde{F}_d$

```
Input: Initial micro-projections: $p_{\text{init}}$
       Proximal micro-projections input: $\tilde{p}$
Output: Final micro-projections: $p^*$
Initialize: $p \leftarrow p_{\text{init}}$
for $i \leftarrow 1$ to $n_p$ do
    $e \leftarrow y + \log\{C \exp\{-p\}\}$
    $g = -\text{diag}(\exp\{-p\})\, C^\top \text{diag}(C \exp\{-p\})^{-1} De$
    $\eta = \eta_0$
    while $f_d(p - \eta g) > f_d(p) - \eta\epsilon \|g\|^2$ do
        $\eta \leftarrow \eta/2$
    $p \leftarrow p - \eta g$
$p^* \leftarrow p$
```

of x_{init}. The optimization can be performed using any off-the-shelf software module that can perform regularized inversion. More implementation details are given in section 3.5.

```
**Algorithm 5:** Computation of the tomographic reconstruction function $\tilde{F}_t$
    **Input:** Initial reconstruction: $x_{\text{init}}$
               Projections data: $\tilde{p}$

    **Output:** Final reconstruction: $x^*$
    Initialize: $x \leftarrow x_{\text{init}}$
    **for** $i \leftarrow 1$ *to* $n_t$ **do**
        $e \leftarrow p - Ax$
        $x \leftarrow x + \text{Update}(e, A)$
    $x^* \leftarrow x$
```

Even though the coded deblurring and CT reconstruction problems are tightly coupled, the modular structure of Algorithm (3) separates them into deblurring and CT reconstruction sub-problems that must be solved repeatedly until convergence. Note that the deblurring operator F_d performs the optimization in equation (3.18) purely in the projection domain and is independent of the CT geometry. Only the tomographic reconstruction operator F_t, and the forward projection operator A in Algorithm (3) depend on the CT geometry under consideration. Our approach can therefore be easily extended to other CT geometries by incorporating a different reconstruction operator F_t, and a forward projection operator A specific to that CT geometry.

3.4 Interlaced View Sampling

Recently proposed interlaced view sampling schemes [33], [40] have demonstrated improved reconstruction quality for time-resolved tomography compared to traditional progressive view sampling. In interlaced view sampling, the view measurements are collected over multiple rotations of the object, rather than a single rotation as progressive sampling. This allows a wider range of angular measurements per unit time, thereby improving the reconstruction quality for time-resolved tomography [33].

Interlaced view sampling schemes typically require a large spacing between consecutive view angles, causing considerable motion blur in fly-scanning. CodEx is thus well suited for interlaced view sampling as the resulting motion blur can be inverted to improve the reconstruction quality.

However, using an irregularly spaced interlaced view sampling can make the CodEx reconstruction very computationally expensive. To see why this is, consider equation (3.3) that formulates the expected photon counts at a view-angle as a discrete sum over K micro-projection angles. If the view-angles are not regularly spaced, then the micro-projection angles for different view-angles will not match. In the worst case, the measurements at M_θ view-angles will be a function of KM_θ micro-projection angles. This increases the memory footprint of the projection domain variables $p, u \in \mathbb{R}^{KM_\theta M_d}$ in Algorithm 3 drastically, as well as increases the computational complexity of the operators F_d and F_t in equations (3.18) (3.19) considerably.

In order to improve the computational efficiency of CodEx reconstruction with interlaced views, we propose a interlaced view sampling scheme with regularly spaced view angles. The M_θ interlaced measurement angles are given as

$$\theta_i = \frac{\pi i K}{N_\theta} \text{ , for } i = 0, \cdots, M_\theta - 1, \tag{3.21}$$

where K is the code length of coded exposure and N_θ is the number of unique micro-projection angles in $[0, \pi]$ chosen such that $\gcd(K, N_\theta) = 1$. Not every choice of N_θ will yield a good interlaced angle scheme that offers a uniform coverage in the $[0, \pi]$ angular range. Theorem B.0.1 in Appendix B shows that ensuring $\gcd(K, N_\theta) = 1$ allows the view angles θ_i for $i = 0, \ldots, (N_\theta - 1)$ to cover every unique angle $\frac{\pi j}{N_\theta}$ in $[0, \pi]$ for $j = 0, \ldots, (N_\theta - 1)$.

In order to ensure $\gcd(K, N_\theta) = 1$, we choose N_θ in the following way

$$N_\theta = mK - n \ , \tag{3.22}$$

where the positive integers m and n are chosen so that n and K are co-primes. Once a suitable n is chosen that is co-prime to K, m can be adjusted to tune the angular spacing between view-angles. For a small n, the view-angles θ_i in equation (3.21) are roughly separated by $\frac{\pi}{m}$. Theorem B.0.2 in Appendix B shows that our choice of N_θ satisfies the constraint $\gcd(K, N_\theta) = 1$. Some typical choices of the parameters K, m, n is given in Table 3.1.

Table 3.1. Typical parameter choices for view-angles

Code-length K	m	n	$N_\theta = mK - n$	Blur-angle $\Delta\theta = \frac{K180^\circ}{N_\theta}$
52	2	27	77	121.56°
52	5	27	233	40.17°
52	10	27	493	18.98°
52	20	27	1013	9.24°

Fig 3.3 graphically illustrates the interlaced view-angle scheme in equation (3.21) for the case when $m = 5$, $n = 5$, and $K = 11$. In this case, there are $N_\theta = mK - n = 50$ distinct micro-projection angles in the range of $[0, \pi]$. Notice that each new measurement angle shown by a blue dot is spaced by $K = 11$ discrete angular steps, but the measurement angles do not repeat until all $N_\theta = 50$ distinct micro-projection angles are sampled.

The expression of expected photon-counts at the detector given in equation (3.4) can be written more explicitly for the proposed interlaced view angle sampling as

$$\bar{\lambda}_i = \lambda^0 \sum_{k=0}^{K-1} c_k \exp\left\{-\left(A_{\frac{\pi(iK+k)}{N_\theta}}\right) x\right\}, \text{ for } i = 0, \cdots, M_\theta - 1, \tag{3.23}$$

where $\bar{\lambda}_i \in \mathbb{R}^{M_d}$ is the vector of expected photon-counts at the M_d detector pixels, λ^0 is the photon-flux of the X-ray source, $c = [c_0, c_1, \cdots, c_{K-1}]$ is the binary code used to modulate the photons, and $A_{\frac{\pi i K}{N_\theta}} \in \mathbb{R}^{M_d \times N}$ performs the forward projection of the image x at angle $\frac{\pi i K}{N_\theta}$.

In order to write equation (3.23) in a vectorized form, let us define $\bar{\lambda} \in \mathbb{R}^{M_d M_\theta}$ to be the vector of expected photon-counts at the detector for the M_θ measurement angles as

$$\bar{\lambda} = [\bar{\lambda}_0, \cdots, \bar{\lambda}_{M_\theta - 1}]^\top . \tag{3.24}$$

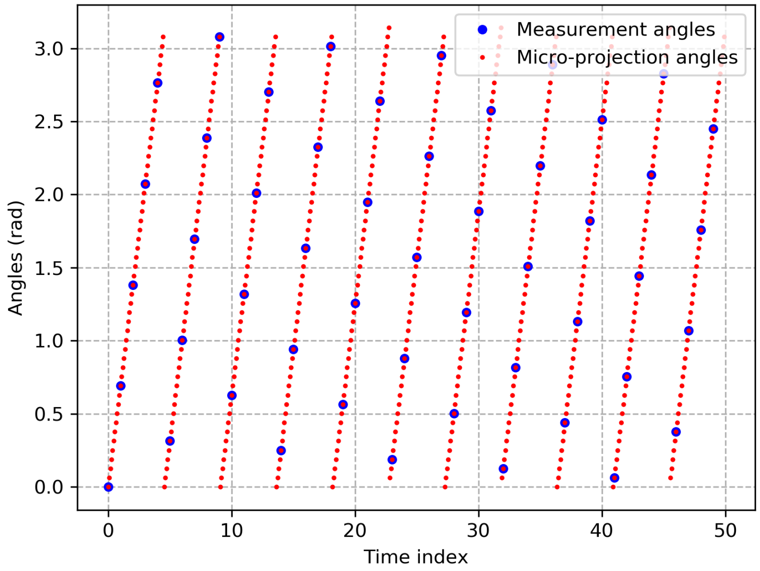


Figure 3.3. Illustration of our View angle sampling for coded exposure CT for $N_\theta = mK - n = 50$ ($K = 11$, $m = 5$, $n = 5$) and $M_\theta = 50$. Each new measurement angle shown by a blue dot is spaced by $K = 11$ discrete angular steps, but the measurement angles do not repeat until all $N_\theta = 50$ distinct micro-projection angles (blue dots) are sampled.

Let us also define the forward projection matrix for the N_θ micro-projection angles as

$$A = [A_0^\top, A_{\frac{\pi}{N_\theta}}^\top, \cdots, A_{\frac{(N_\theta-1)\pi}{N_\theta}}^\top]^\top \ . \tag{3.25}$$

Using the above notation, equation (3.23) can be expressed in a more compact vectorized form like equation (3.4) as

$$\bar{\lambda} = \bar{c}\lambda^0 C \exp\{-Ax\} \ , \tag{3.26}$$

where $\bar{c} = \sum_{i=0}^{K-1} c_k$ is used for normalization, λ^0 is the photon-flux of the X-ray source, and the matrix C that performs the coded sum is defined as

$$
\begin{aligned}
& C_{i(i_\theta, i_r, i_c), j(j_\theta, j_r, j_c)} \\
&= \begin{cases} \frac{c_k}{\bar{c}} & \text{if } \mathrm{mod}(i_\theta K + k, N_\theta) = j_\theta \text{ for some } k < K \ , \\ & \text{and } i_r = j_r, \text{ and } i_c = j_c \\ 0 & \text{otherwise} \end{cases} ,
\end{aligned}
\tag{3.27}
$$

where mod() denotes the modulo operation and $i(i_\theta, i_r, i_c)$ represents the rasterized index i as a function of the angular index i_θ, row index i_r, and column index i_c. Similarly, $j(j_\theta, j_r, j_c)$ represents the rasterized index j as a function of the angular index j_θ, row index j_r, and column index j_c.

3.5 Results

We present results using simulated and binned physical data in order to demonstrate the effectiveness for our CodEx approach. We consider three experimental cases outlined as follows:

1. Simulated data without noise

2. Simulated data with photon noise

3. Binned physical data.

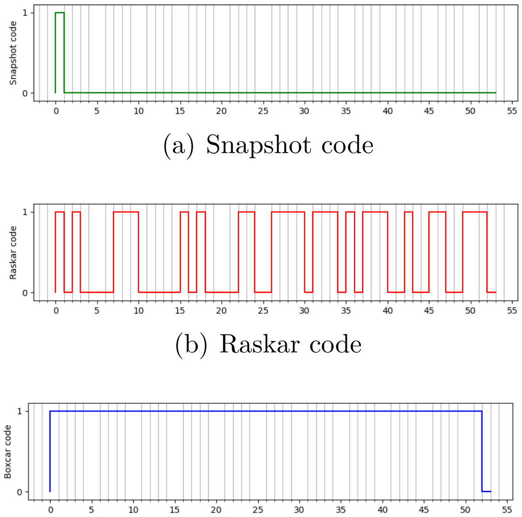
In each experimental case, we compare the result of our Algorithm 3 to the baseline solution in equation (3.28) for different exposure codes. In particular, we consider three exposure codes:

1. Snapshot code: a single one followed by all zeros $(1, 0, \cdots, 0)$

2. Raskar code [69]

3. Boxcar code: all ones $(1, 1, \cdots, 1)$

Figure 3.4 displays these codes as a plot. The original Raskar code [69] is of length 52 but we extend it to larger code-lengths by repeating it by an integer number of times.



(a) Snapshot code

(b) Raskar code

(c) Boxcar code

Figure 3.4. Examples of the binary codes of length 52 that are used for modulating the photon-flux in our experiments.

As a baseline method for comparison, we use an iterative reconstruction method without modelling the coded blur. Mathematically this can be written as

$$x^*_{\text{baseline}} = \arg\min_x \left\{ \frac{\lambda^0}{2} \sum_{i=0}^{M_\theta - 1} \left\| y_i - A_{\frac{\pi i K}{N_\theta}} x \right\|^2 + h(x) \right\}, \tag{3.28}$$

where $y_i \in \mathbb{R}^{M_d}$ is the vector of projection measurements at angle $\frac{\pi i K}{N_\theta}$ in accordance with the view sampling in equation (3.21), $A_{\frac{\pi i K}{N_\theta}} \in \mathbb{R}^{M_d \times N}$ performs forward projection of the image $x \in \mathbb{R}^N$ at angle $\frac{\pi i K}{N_\theta}$. The prior model $h()$ is chosen to be the same as the CodEx reconstruction.

CodEx can utilize any off-the-shelf regularized inversion solver for F_t in equation (3.19) to compute the solution in Algorithm 3. The baseline method in equation (3.28) can similarly be computed using any off-the-shelf regularized inversion solver. In all our experimental results, we use svmbir (https://github.com/cabouman/svmbir) to compute F_t as well as the baseline reconstruction. The svmbir implementation uses a Markov random field based regularization and the reconstruction is computed using a cache optimized iterative coordinate descent [77].

For computing the CodEx reconstruction, we run 1000 ADMM iterations in Algorithm 3. For each ADMM iteration, we run $n_p = 5$ partial updates for computing the $\tilde{F}_d$ step and $n_t = 5$ partial updates for computing the $\tilde{F}_t$ step. For computing the baseline reconstruction, we perform 400 iterations.

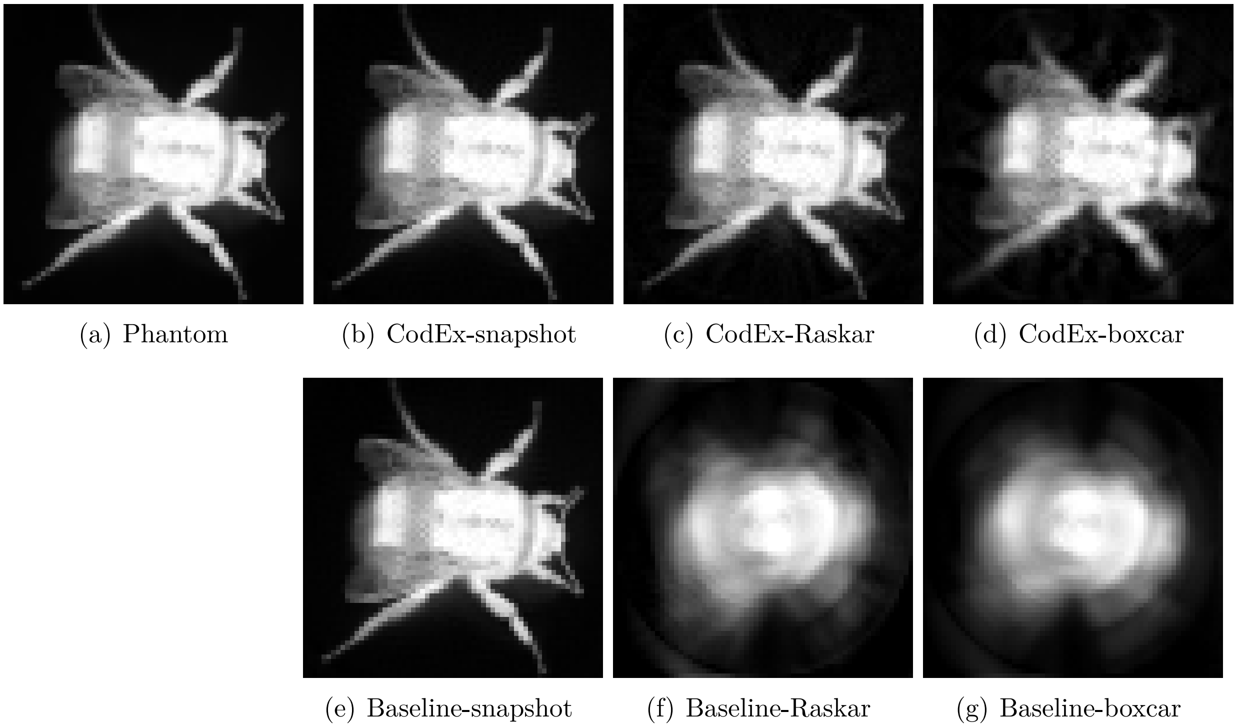


Figure 3.5. Comparison of reconstruction quality for simulated data without noise. Experimental details are in Table 3.2. For the Raskar and boxcar codes, the baseline reconstruction produces a blurred image. In contrast, CodEx is able to reconstruct the image without suffering from severe blurring. The CodEx-boxcar reconstruction has noticeable radial blur due to the non-invertible nature of the boxcar blur kernel. In contrast, the CodEx-Raskar and CodEx-snapshot reconstructions do not suffer from severe radial blur artifacts.

3.5.1 Simulated data without noise

In this section we perform simulated experiments without noise in order to demonstrate the ability to deblur the coded blur introduced during the measurement process. We start with a phantom image, use forward-projection to generate simulated noise-free measurements, and then perform reconstruction from the simulated measurements. In order to generate simulated noise-free measurements, we make use of equations (3.4) and (3.5) with an infinitely large λ^0. We set the diagonal elements of the weight matrix D to be 10^3 in equation (3.18). Details of the experimental setup are given in Table 3.2.

Table 3.2. Experimental setup for simulated data without noise

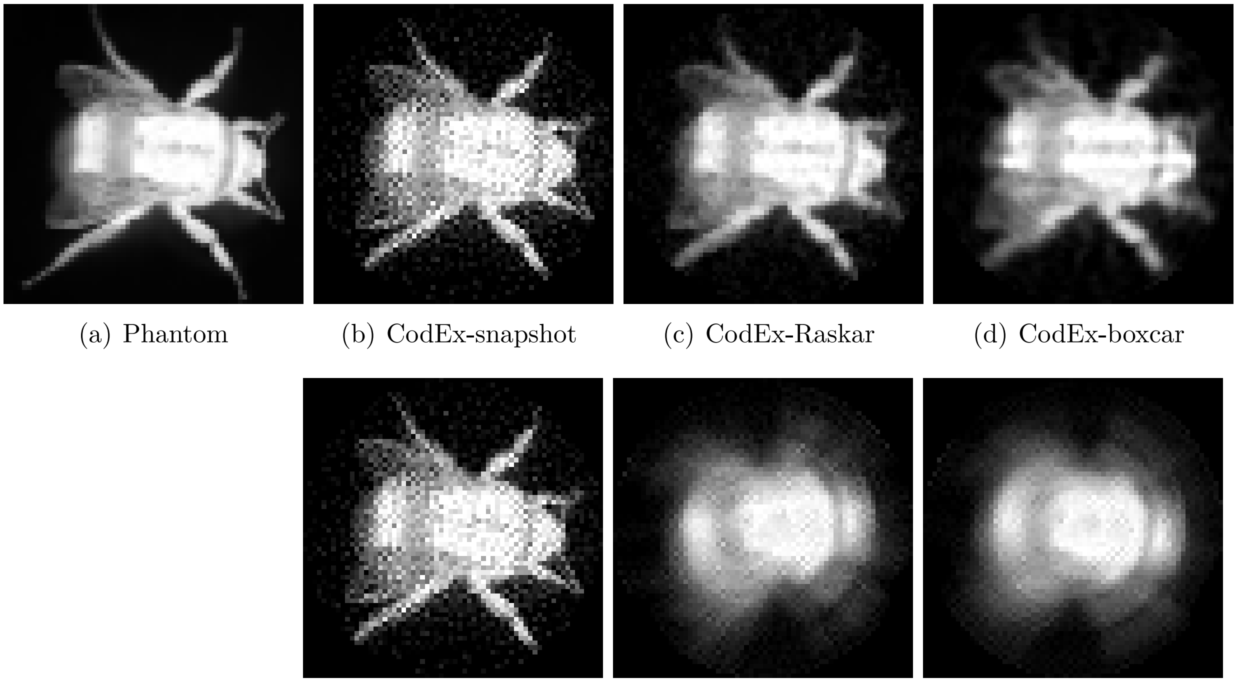
Number of Views, M_θ	233
Number of Unique Angles, N_θ	233
Angular Span	180°
Code Length, K	52
Blur Angle	40.17°
Reconstruction Size (x, y)	64×64

Figure 3.5 provides a qualitative comparison between our method CodEx and the baseline for different exposure codes. For the Raskar and boxcar codes, the baseline reconstruction produces a blurred image. In contrast, CodEx is able to reconstruct the image without suffering from severe blurring. The CodEx-boxcar reconstruction has noticeable radial blur due to the non-invertible nature of the boxcar blur kernel. In contrast, the CodEx-Raskar and CodEx-snapshot reconstructions do not suffer from severe radial blur artifacts.

3.5.2 Simulated data with photon noise

In this section, we perform simulated experiments with Poisson noise in order to simulate real coded-exposure tomographic systems. We start with a phantom image, use forward-projection to generate simulated noisy measurements, and then perform reconstruction from the simulated measurements. In order to generate simulated measurements, we make use of equations (3.4) and (3.5). The experimental details are summarized in Table 3.3.

Figure 3.6 shows a qualitative comparison of CodEx with the baseline for different coded exposures. For Raskar and boxcar codes, the baseline reconstruction leads to a blurred image. In contrast, CodEx-Raskar and CodEx-boxcar reconstructions do not suffer from severe blurring artifacts. The reconstructions with snapshot code suffer from high noise due to the limited photon counts in the measurements. The reconstructions with boxcar code


(a) Phantom (b) CodEx-snapshot (c) CodEx-Raskar (d) CodEx-boxcar

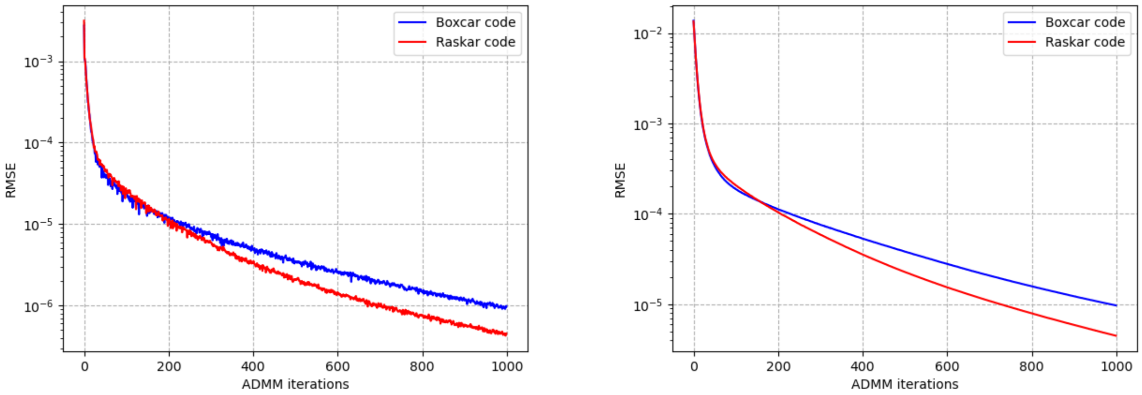
(e) Baseline-snapshot (f) Baseline-Raskar (g) Baseline-boxcar

Figure 3.6. Comparison of reconstruction quality for simulated data with photon noise. Experimental details are in Table 3.3. For Raskar and boxcar codes, the baseline reconstruction leads to a blurred image. In contrast, CodEx-Raskar and CodEx-boxcar reconstructions do not suffer from severe blurring artifacts. The reconstructions with snapshot code suffer from high noise due to the limited photon counts in the measurements. The reconstructions with boxcar code suffer from radial blur artifacts and loss of fine features due to the non-invertible nature of the blur kernel.

Table 3.3. Experimental setup for simulated data with noise

λ^0	10000
Number of Views, M_θ	100
Number of Unique Angles, N_θ	233
Angular Span	180°
Code Length, K	52
Blur Angle	40.17°
Reconstruction Size (x, y)	64×64

suffer from radial blur artifacts and loss of fine features due to the non-invertible nature of the blur kernel.



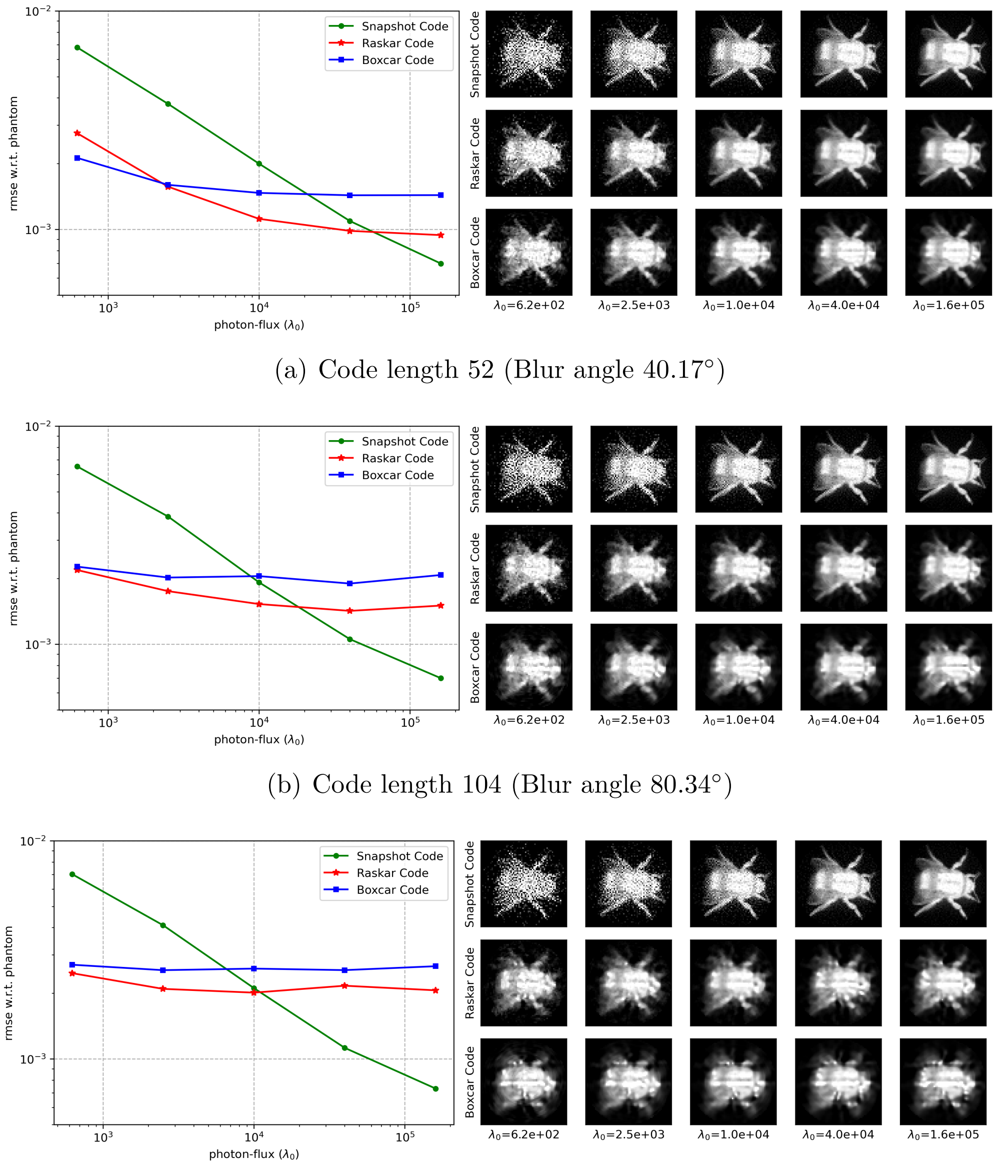
(a) Primal Residual: RMSE(Ax^t, p^t)

(b) Dual Residual: RMSE(Ax^t, Ax^{t-1})

Figure 3.7. Primal and dual residual convergence plots for simulated data with photon noise. Here t refers to the ADMM iteration number. The Raskar code leads to an improved convergence than the boxcar code.

In Figure 3.7, we plot the primal residual, RMSE(Ax^t, p^t) and dual residual, RMSE(Ax^t, Ax^{t-1}) [58] at each ADMM iteration to illustrate the convergence of our method. Here t refers to the ADMM iteration number. The Raskar code leads to an improved convergence than the boxcar code.

In Figure 3.8, we show the effect of the photon-flux (λ^0) and the code length on reconstruction quality for different codes. For each code type snapshot, Raskar, and boxcar, we vary the photon-flux (λ^0) and plot the rmse with respect to the phantom to measure image quality. We also show the resulting image for a visual inspection of image quality. We repeat this process for each code lengths of 52, 104, and 208. The remaining experimental parameters are kept the same as Table 3.3. When the photon-flux (λ^0) is low, the boxcar code and Raskar code produce better image quality than the snapshot code as a result of collecting more photons. When the photon-flux (λ^0) is high, the minor gains from increased photon count do not fully compensate for the loss of information by inverting the coded blur. Consequently at high photon-flux, the snapshot code produce better image quality than the boxcar code and Raskar codes. Increasing the blur angle for a given photon-flux leads to the degradation of image quality for the boxcar code and Raskar codes since inverting the coded blur causes more loss of information. This loss of image quality is less pronounced for the



(a) Code length 52 (Blur angle 40.17°)

(b) Code length 104 (Blur angle 80.34°)

(c) Code length 208 (Blur angle 160.68°)

Figure 3.8. Effect of the photon-flux (λ^0) and the code length on reconstruction quality for different codes. The remaining experimental parameters are kept the same as Table 3.3. When the photon-flux (λ^0) is low, the boxcar code and Raskar code produce better image quality than the snapshot code as a result of collecting more photons. When the photon-flux (λ^0) is high, the minor gains from increased photon count do not fully compensate for the loss of information by inverting the coded blur. Consequently at high photon-flux, the snapshot code produce better image quality than the boxcar code and Raskar codes.

Raskar code since the code is more invertible than the boxcar code. Consequently, when the photon-flux (λ^0) is low, increasing the blur angle for the Raskar code can be beneficial since the loss of information inverting the coded blur is compensated by the lower measurement noise due to increased photon collection.

Table 3.4. Experimental setup for checking robustness with respect to the underlying object to be reconstructed.

λ^0	40000
Number of Views, M_θ	200
Number of Unique Angles, N_θ	493
Angular Span	180°
Code Length, K	104
Blur Angle	37.97°
Reconstruction Size (x, y)	128 × 128

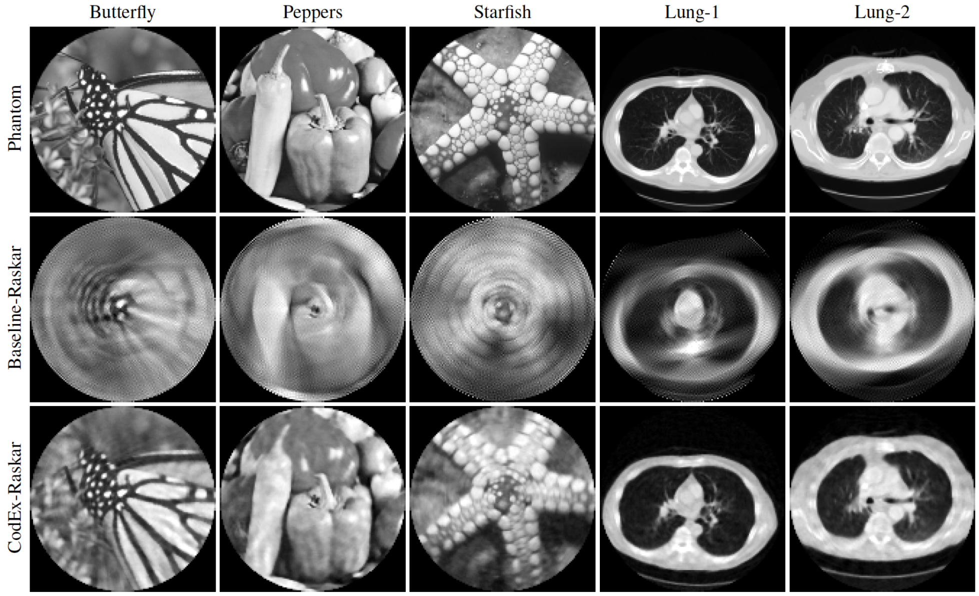
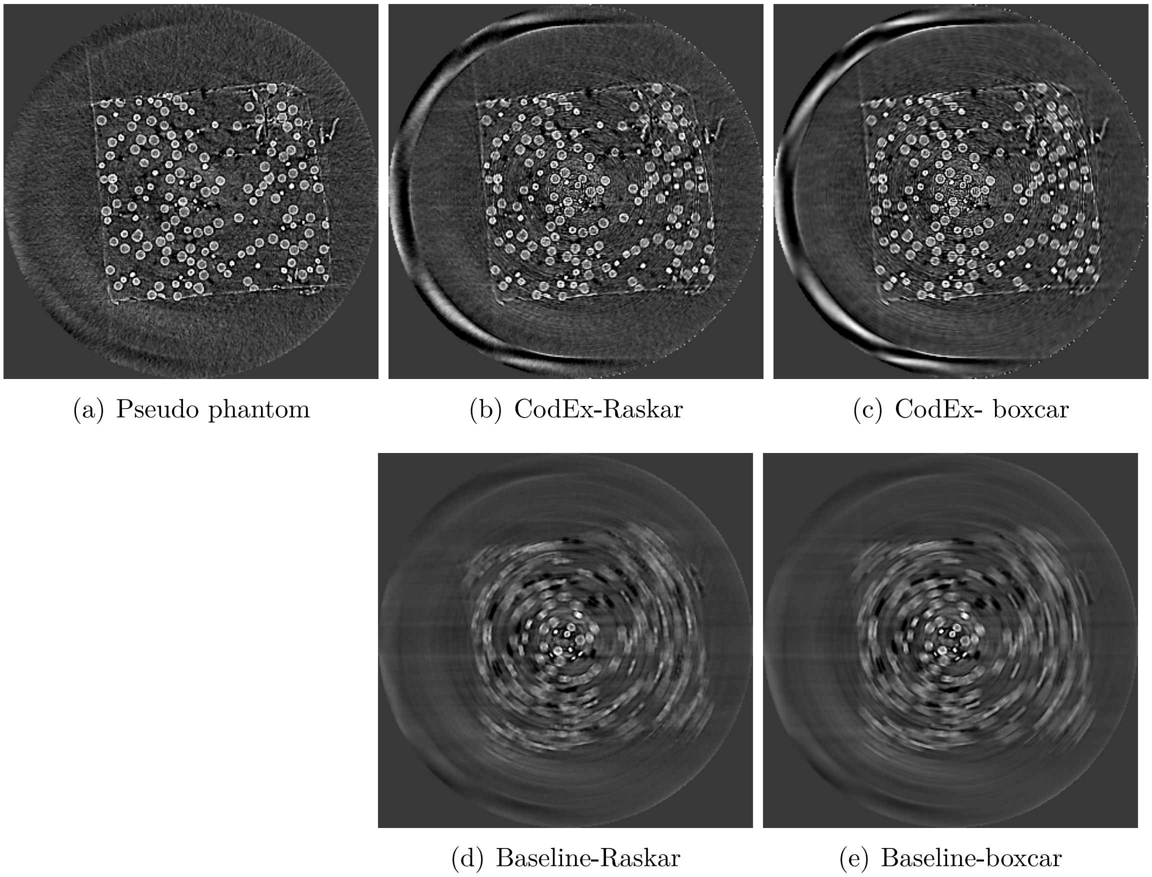


Figure 3.9. Robustness with respect to underlying reconstructed object. Experimental details are summarized in Table 3.4. The CodEx reconstruction is able to recover the major features of the image despite the high angular blur. In contrast, the baseline results in overtly blurred reconstructions.

In order to show the robustness of our method, we perform simulated experiments using different widely-used natural images and CT images [78] as the phantom. For each phantom, we perform forward-projection to generate simulated noisy measurements and then perform reconstruction from the simulated measurements. To generate simulated measurements, we make use of equations (3.4) and (3.5). Experimental details are summarized in Table 3.4. Figure 3.9 compares the CodEx reconstruction with the baseline reconstruction for each phantom image. The CodEx reconstruction is able to recover the major features of the image despite the high angular blur. In contrast, the baseline results in overtly blurred reconstructions.

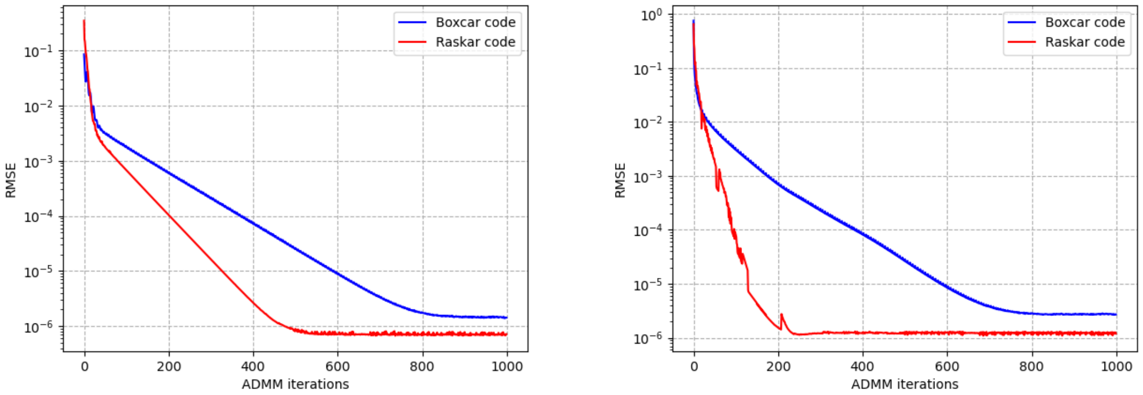
3.5.3 Binned physical data



(a) Pseudo phantom (b) CodEx-Raskar (c) CodEx- boxcar

(d) Baseline-Raskar (e) Baseline-boxcar

Figure 3.10. Comparison of reconstruction quality for binned physical data. Experimental details are summarized in Table 3.5. CodEx reconstructions preserve the major features in the image whereas the baseline leads to overtly blurred reconstructions.



(a) Primal Residual: RMSE(Ax^t, p^t)

(b) Dual Residual: RMSE(Ax^t, Ax^{t-1})

Figure 3.11. Primal and dual residual convergence plots for binned physical data. Here t refers to the ADMM iteration number. The Raskar code leads to an improved convergence than the boxcar code.

In this section, we perform semi-simulated experiments by binning physical experimental data. Binning physical data allows us to generate arbitrary coded measurements without performing physical experiments using coded exposure. We start with an existing experimental data, generate coded measurements by binning, and then perform reconstruction from the coded measurements. The object in consideration contains borosilicate glass spheres of different sizes encased in a polypropylene matrix [79]. The experimental details are summarized in Table 3.5.

Table 3.5. Experimental setup for binned physical data

X-ray Energy	27.4 keV
Exposure Time	0.0001 s
Number of Views, M_θ	1500
Number of Unique Angles, N_θ	1500
Angular Span	180°
Code Length, K	104
Blur Angle	12.48°
Pixel Size	5.2 μm
Reconstruction Size (x, y)	320×320

Using the original experimental measurements at N_θ distinct view-angles, we generate $M_\theta = N_\theta$ coded measurements as

$$y_i = -\log\left\{\sum_{k=0}^{K-1} \frac{c_k}{\bar{c}} \exp\left\{-\tilde{y}_{\mathrm{mod}(iK+k,N_\theta)}\right\}\right\} \ , \qquad (3.29)$$
$$\text{for } i = 0, \cdots, N_\theta - 1 \ ,$$

where $\tilde{y}_{\mathrm{mod}(iK+k,N_\theta)}$ is the vector of projection measurements at angle $\frac{\pi(iK+k)}{N_\theta}$ obtained from a physical experiment, y_i is the vector of coded measurements at the i^{th} view angle, $c = [c_0, \cdots, c_{K-1}]$ is the binary code of length K. Note that since we generate N_θ coded measurements from the N_θ original physical measurements, we can not create extra information by coding and our reconstruction quality with coded measurements can at best be that of using all the original measurements.

Figure 3.10 shows a comparison of our method with the baseline and the phantom for measurements with different exposure codes. For the Raskar and boxcar codes, direct reconstruction produces a blurred image. In contrast, our PnP coded exposure method is able to reconstruct the image without suffering from severe blurring.

In Figure 3.11 we plot the primal residual, RMSE(Ax^t, p^t) and dual residual, RMSE(Ax^t, Ax^{t-1}) [58] at each ADMM iteration to illustrate the convergence of our method. Here t refers to the ADMM iteration number. The Raskar code leads to an improved convergence than the boxcar code.

3.6 Conclusion

In this paper, we proposed CodEx, a novel method for coded exposure CT acquisition and reconstruction. Coding the exposure during the acquisition allows us to collect more photons while at the same time effectively remove motion artifacts in the reconstruction. The modular structure of the CodEx reconstruction algorithm makes it easy to extend our method to different CT geometries.

REFERENCES

[1] L. Tian, S. A. Hires, T. Mao, D. Huber, M. E. Chiappe, S. H. Chalasani, L. Petreanu, J. Akerboom, S. A. McKinney, E. R. Schreiter, *et al.*, “Imaging neural activity in worms, flies and mice with improved gcamp calcium indicators,” *Nature methods*, vol. 6, no. 12, pp. 875–881, 2009.

[2] J. Akerboom, N. Carreras Calderón, L. Tian, S. Wabnig, M. Prigge, J. Tolö, A. Gordus, M. B. Orger, K. E. Severi, J. J. Macklin, *et al.*, “Genetically encoded calcium indicators for multi-color neural activity imaging and combination with optogenetics,” *Frontiers in molecular neuroscience*, vol. 6, p. 2, 2013.

[3] J. Akerboom, T.-W. Chen, T. J. Wardill, L. Tian, J. S. Marvin, S. Mutlu, N. C. Calderón, F. Esposti, B. G. Borghuis, X. R. Sun, *et al.*, “Optimization of a gcamp calcium indicator for neural activity imaging,” *The Journal of neuroscience*, vol. 32, no. 40, pp. 13 819–13 840, 2012.

[4] T.-W. Chen, T. J. Wardill, Y. Sun, S. R. Pulver, S. L. Renninger, A. Baohan, E. R. Schreiter, R. A. Kerr, M. B. Orger, V. Jayaraman, *et al.*, “Ultrasensitive fluorescent proteins for imaging neuronal activity,” *Nature*, vol. 499, no. 7458, pp. 295–300, 2013.

[5] A. C. Kwan, “What can population calcium imaging tell us about neural circuits?” *Journal of neurophysiology*, vol. 100, no. 6, pp. 2977–2980, 2008.

[6] M. A. Martens, W. Boesmans, and P. V. Berghe, “Calcium imaging at khz frame rates resolves millisecond timing in neuronal circuits and varicosities,” *Biomedical optics express*, vol. 5, no. 8, pp. 2648–2661, 2014.

[7] D. Padfield, J. Rittscher, and B. Roysam, “Spatio-temporal cell segmentation and tracking for automated screening,” in *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, IEEE, 2008, pp. 376–379.

[8] X. Chen, X. Zhou, and S. T. Wong, “Automated segmentation, classification, and tracking of cancer cell nuclei in time-lapse microscopy,” *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 4, pp. 762–766, 2006.

[9] Z. Khan, T. Balch, and F. Dellaert, “Mcmc-based particle filtering for tracking a variable number of interacting targets,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 11, pp. 1805–1819, 2005.

[10] K. Smith, D. Gatica-Perez, and J.-M. Odobez, “Using particles to track varying numbers of interacting people,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, IEEE, vol. 1, 2005, pp. 962–969.

[11] D. P. Mukherjee, N. Ray, and S. T. Acton, "Level set analysis for leukocyte detection and tracking," *IEEE Transactions on Image processing*, vol. 13, no. 4, pp. 562–572, 2004.

[12] A. Dufour, V. Shinin, S. Tajbakhsh, N. Guillén-Aghion, J.-C. Olivo-Marin, and C. Zimmer, "Segmenting and tracking fluorescent cells in dynamic 3-d microscopy with coupled active surfaces," *IEEE Transactions on Image Processing*, vol. 14, no. 9, pp. 1396–1410, 2005.

[13] D. Padfield, J. Rittscher, N. Thomas, and B. Roysam, "Spatio-temporal cell cycle phase analysis using level sets and fast marching methods," *Medical image analysis*, vol. 13, no. 1, pp. 143–155, 2009.

[14] D. Dormann, T. Libotte, C. J. Weijer, and T. Bretschneider, "Simultaneous quantification of cell motility and protein-membrane-association using active contours," *Cell motility and the cytoskeleton*, vol. 52, no. 4, pp. 221–230, 2002.

[15] O. Dzyubachyk, W. A. van Cappellen, J. Essers, W. J. Niessen, and E. Meijering, "Advanced level-set-based cell tracking in time-lapse fluorescence microscopy," *IEEE transactions on medical imaging*, vol. 29, no. 3, pp. 852–867, 2010.

[16] H. Shen, G. Nelson, S. Kennedy, D. Nelson, J. Johnson, D. Spiller, M. R. White, and D. B. Kell, "Automatic tracking of biological cells and compartments using particle filters and active contours," *Chemometrics and Intelligent Laboratory Systems*, vol. 82, no. 1, pp. 276–282, 2006.

[17] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International journal of computer vision*, vol. 1, no. 4, pp. 321–331, 1988.

[18] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE transactions on pattern analysis and machine intelligence*, vol. 17, no. 2, pp. 158–175, 1995.

[19] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International journal of computer vision*, vol. 22, no. 1, pp. 61–79, 1997.

[20] N. Paragios and R. Deriche, "Coupled geodesic active regions for image segmentation: A level set approach," in *European Conference on Computer Vision*, Springer, 2000, pp. 224–240.

[21] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on image processing*, vol. 10, no. 2, pp. 266–277, 2001.

[22] D. Cremers, F. Tischhäuser, J. Weickert, and C. Schnörr, "Diffusion snakes: Introducing statistical shape knowledge into the mumford-shah functional," *International journal of computer vision*, vol. 50, no. 3, pp. 295–313, 2002.

[23] D. Cremers, S. J. Osher, and S. Soatto, "Kernel density estimation and intrinsic alignment for shape priors in level set segmentation," *International Journal of Computer Vision*, vol. 69, no. 3, pp. 335–351, 2006.

[24] S. H. Joshi and A. Srivastava, "Intrinsic bayesian active contours for extraction of object boundaries in images," *International journal of computer vision*, vol. 81, no. 3, pp. 331–355, 2009.

[25] M. E. Leventon, W. E. L. Grimson, and O. Faugeras, "Statistical shape influence in geodesic active contours," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, IEEE, vol. 1, 2000, pp. 316–323.

[26] M. Kulikova, I. Jermyn, X. Descombes, J. Zerubia, and E. Zhizhina, "A marked point process model with strong prior shape information for the extraction of multiple, arbitrarily-shaped objects," in *Signal-Image Technology & Internet-Based Systems (SITIS), 2009 Fifth International Conference on*, IEEE, 2009, pp. 180–186.

[27] M. Kulikova, A. Veillard, L. Roux, and D. Racoceanu, "Nuclei extraction from histopathological images using a marked point process approach," in *SPIE Medical Imaging*, International Society for Optics and Photonics, 2012, pp. 831 428–831 428.

[28] M. Elad, "Sparse and redundant representation," 2010.

[29] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

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

[31] E. Hodneland, T. Kögel, D. M. Frei, H.-H. Gerdes, and A. Lundervold, "Cellsegm-a matlab toolbox for high-throughput 3d cell segmentation," *Source code for biology and medicine*, vol. 8, no. 1, p. 1, 2013.

[32] C. Huang, J. L. Ackerman, Y. Petibon, T. J. Brady, G. El Fakhri, and J. Ouyang, "MR-based motion correction for PET imaging using wired active MR microcoils in simultaneous PET-MR: Phantom study," *Medical physics*, vol. 41, no. 4, 2014.

[33] K. A. Mohan, S. Venkatakrishnan, J. W. Gibbs, E. B. Gulsoy, X. Xiao, M. De Graef, P. W. Voorhees, and C. A. Bouman, “TIMBIR: A method for time-space reconstruction from interlaced views.,” *IEEE Trans. Computational Imaging*, vol. 1, no. 2, pp. 96–111, 2015.

[34] T. Balke, S. Majee, G. T. Buzzard, S. Poveromo, P. Howard, M. A. Groeber, J. McClure, and C. A. Bouman, “Separable models for cone-beam MBIR reconstruction,” *Electronic Imaging*, vol. 2018, no. 15, pp. 181–1, 2018.

[35] S. Majee, T. Balke, C. A. Kemp, G. T. Buzzard, and C. A. Bouman, “4D X-Ray CT reconstruction using multi-slice fusion,” in *2019 IEEE International Conference on Computational Photography (ICCP)*, IEEE, 2019, pp. 1–8.

[36] Z. Nadir, M. S. Brown, M. L. Comer, and C. A. Bouman, “A model-based iterative reconstruction approach to tunable diode laser absorption tomography,” *IEEE Transactions on Computational Imaging*, vol. 3, no. 4, pp. 876–890, 2017.

[37] S. Majee, D. H. Ye, G. T. Buzzard, and C. A. Bouman, “A model based neuron detection approach using sparse location priors,” *Electronic Imaging*, vol. 2017, no. 17, pp. 10–17, 2017.

[38] A. Ziabari, D. H. Ye, K. D. Sauer, J. Thibault, and C. A. Bouman, “2.5D deep learning for CT image reconstruction using a multi-GPU implementation,” in *Signals, Systems, and Computers, 2018 52nd Asilomar Conference on*, IEEE, 2018.

[39] J. Gibbs, K. A. Mohan, E. Gulsoy, A. Shahani, X. Xiao, C. Bouman, M. De Graef, and P. Voorhees, “The three-dimensional morphology of growing dendrites,” *Scientific reports*, vol. 5, p. 11 824, 2015.

[40] G. Zang, R. Idoughi, R. Tao, G. Lubineau, P. Wonka, and W. Heidrich, “Space-time tomography for continuously deforming objects,” *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–14, 2018.

[41] S. J. Kisner, E. Haneda, C. A. Bouman, S. Skatter, M. Kourinny, and S. Bedford, “Model-based CT reconstruction from sparse views,” in *Second International Conference on Image Formation in X-Ray Computed Tomography*, 2012, pp. 444–447.

[42] K. Sauer and C. Bouman, “A local update strategy for iterative reconstruction from projections,” *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 534–548, 1993.

[43] D. Clark and C. Badea, “Convolutional regularization methods for 4D, x-ray CT reconstruction,” in *Medical Imaging 2019: Physics of Medical Imaging*, International Society for Optics and Photonics, vol. 10948, 2019, 109482A.

[44] S. Sreehari, S. V. 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.

[45] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, IEEE, 2013, pp. 945–948.

[46] Y. Sun, B. Wohlberg, and U. S. Kamilov, "An online plug-and-play algorithm for regularized image reconstruction," *arXiv preprint arXiv:1809.04693*, 2018.

[47] U. S. Kamilov, H. Mansour, and B. Wohlberg, "A plug-and-play priors approach for solving nonlinear imaging inverse problems," *IEEE Signal Processing Letters*, vol. 24, no. 12, pp. 1872–1876, 2017.

[48] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Transactions on image processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[49] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, "Video denoising using separable 4D nonlocal spatiotemporal transforms," in *Image Processing: Algorithms and Systems IX*, International Society for Optics and Photonics, vol. 7870, 2011, p. 787 003.

[50] G. T. Buzzard, S. H. Chan, S. Sreehari, and C. A. Bouman, "Plug-and-play unplugged: Optimization-free reconstruction using consensus equilibrium," *SIAM Journal on Imaging Sciences*, vol. 11, no. 3, pp. 2001–2020, 2018.

[51] D. Jiang, W. Dou, L. Vosters, X. Xu, Y. Sun, and T. Tan, "Denoising of 3D magnetic resonance images with multi-channel residual learning of convolutional neural network," *Japanese journal of radiology*, vol. 36, no. 9, pp. 566–574, 2018.

[52] Y. Sun, B. Wohlberg, and U. S. Kamilov, "Plug-in stochastic gradient method," *arXiv preprint arXiv:1811.03659*, 2018.

[53] Y. Sun, S. Xu, Y. Li, L. Tian, B. Wohlberg, and U. S. Kamilov, "Regularized fourier ptychography using an online plug-and-play algorithm," *arXiv preprint arXiv:1811.00120*, 2018.

[54] V. Sridhar, G. T. Buzzard, and C. A. Bouman, "Distributed framework for fast iterative CT reconstruction from view-subsets," *Electronic Imaging*, vol. 2018, no. 15, pp. 102–1, 2018.

[55] J.-J. Moreau, "Proximité et dualité dans un espace hilbertien," *Bulletin de la Société mathématique de France*, vol. 93, pp. 273–299, 1965.

[56] R. E. Williamson, R. H. Crowell, and H. F. Trotter, *Calculus of vector functions*. Prentice-Hall, 1972.

[57] V. Sridhar, X. Wang, G. T. Buzzard, and C. A. Bouman, "Distributed memory framework for fast iterative CT reconstruction from view-subsets using multi-agent consensus equilibrium," *Manuscript in preparation for IEEE Transactions on Computational Imaging*, 2018.

[58] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[59] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[60] P. Getreuer, "Rudin-osher-fatemi total variation denoising using split bregman," *Image Processing On Line*, vol. 2, pp. 74–95, 2012.

[61] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[62] R. Mathaes, H.-C. Mahler, J.-P. Buettiker, H. Roehl, P. Lam, H. Brown, J. Luemkemann, M. Adler, J. Huwyler, A. Streubel, *et al.*, "The pharmaceutical vial capping process: Container closure systems, capping equipment, regulatory framework, and seal quality tests," *European Journal of Pharmaceutics and Biopharmaceutics*, vol. 99, pp. 54–64, 2016.

[63] S. Komura and H. Toyofuku, *Wave spring*, US Patent 5,622,358, Apr. 1997.

[64] S. Majee, T. Balke, C. A. Kemp, G. T. Buzzard, and C. A. Bouman, "Multi-slice fusion for sparse-view and limited-angle 4D CT reconstruction," *IEEE Transactions on Computational Imaging*, vol. 7, pp. 448–461, 2021.

[65] D. J. Ching, M. Hidayetoğlu, T. Biçer, and D. Gürsoy, "Rotation-as-fast-axis scanning-probe x-ray tomography: The importance of angular diversity for fly-scan modes," *Applied optics*, vol. 57, no. 30, pp. 8780–8789, 2018.

[66] W. H. Richardson, "Bayesian-based iterative method of image restoration," *JoSA*, vol. 62, no. 1, pp. 55–59, 1972.

[67] L. B. Lucy, “An iterative technique for the rectification of observed distributions,” *The astronomical journal*, vol. 79, p. 745, 1974.

[68] Q. Shan, J. Jia, and A. Agarwala, “High-quality motion deblurring from a single image,” *Acm transactions on graphics (tog)*, vol. 27, no. 3, pp. 1–10, 2008.

[69] R. Raskar, A. Agrawal, and J. Tumblin, “Coded exposure photography: Motion deblurring using fluttered shutter,” in *ACM SIGGRAPH 2006 Papers*, 2006, pp. 795–804.

[70] I. Steven Tilley, A. Sisniega, J. H. Siewerdsen, and J. W. Stayman, “High-fidelity modeling of detector lag and gantry motion in ct reconstruction,” in *Conference proceedings. International Conference on Image Formation in X-Ray Computed Tomography*, NIH Public Access, vol. 2018, 2018, p. 318.

[71] D. Ching, S. Aslan, V. Nikitin, and D. Gürsoy, “Time-coded aperture for x-ray imaging,” *Optics Letters*, vol. 44, no. 11, pp. 2803–2806, 2019.

[72] J. Cant, W. J. Palenstijn, G. Behiels, and J. Sijbers, “Modeling blurring effects due to continuous gantry rotation: Application to region of interest tomography,” *Medical Physics*, vol. 42, no. 5, pp. 2709–2717, 2015.

[73] M. Pergament, M. Kellert, K. Kruse, J. Wang, G. Palmer, L. Wissmann, U. Wegner, and M. Lederer, “High power burst-mode optical parametric amplifier with arbitrary pulse selection,” *Optics express*, vol. 22, no. 18, pp. 22 202–22 210, 2014.

[74] M. Gembicky, D. Oss, R. Fuchs, and P. Coppens, “A fast mechanical shutter for submicrosecond time-resolved synchrotron experiments,” *Journal of synchrotron radiation*, vol. 12, no. 5, pp. 665–669, 2005.

[75] T. Balke, S. Majee, G. T. Buzzard, S. Poveromo, P. Howard, M. A. Groeber, J. McClure, and C. A. Bouman, “Model-based cone-beam tomography with scatter correction,” *Document in preparation for the IEEE Transactions on Computational Imaging*,

[76] L. Armijo, “Minimization of functions having lipschitz continuous first partial derivatives,” *Pacific Journal of mathematics*, vol. 16, no. 1, pp. 1–3, 1966.

[77] X. Wang, K. A. Mohan, S. J. Kisner, C. Bouman, and S. Midkiff, “Fast voxel line update for time-space image reconstruction,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, pp. 1209–1213.

[78] B. Albertina, M. Watson, C. Holback, R. Jarosz, S. Kirk, Y. Lee, and J. Lemmerman, "Radiology data from the cancer genome atlas lung adenocarcinoma [TCGA-LUAD] collection," *The Cancer Imaging Archive*, 2016. DOI: http://doi.org/10.7937/K9/TCIA.2016.JGNIHEP5.

[79] S. Singh, T. J. Stannard, S. S. Singh, A. S. Singaravelu, X. Xiao, and N. Chawla, "Varied volume fractions of borosilicate glass spheres with diameter gaussian distributed from 38-45 micronsen cased in a polypropylene matrix," Argonne National Lab.(ANL), Argonne, IL (United States), Tech. Rep., 2017.

A. CORRECTION FOR FIXTURES OUTSIDE THE FIELD OF VIEW

Algorithm 6: Blind fixture correction

```
Inputs: Original Sinogram: $y$,
        System Matrix: $A$,
Output: Corrected Sinogram: $y^{\mathrm{corr}}$
$x \leftarrow \mathrm{recon}(y, A)$
$x^{\mathrm{m}} \leftarrow \mathrm{mask}(x)$
$e \leftarrow y - Ax^{\mathrm{m}}$
$p \leftarrow \mathrm{blur}(e)$
$c \leftarrow \arg\min_{c \in \mathbb{R}} \|e - cp\|^2$
$y^{\mathrm{corr}} \leftarrow y - cp$
```

Here we describe our correction for fixtures placed out of the field of view of the scanner. As shown in Figure 2.15, the setup is held together by a fixture constructed of tubes and plates. The tubes were placed outside the field of view of the CT scanner, thus causing artifacts in the reconstruction. Our method performs a blind source separation of the projection of the object from that of the tubes. Our blind separation relies on the fact that the projection of the tubes is spatially smooth. This is true since the tubes themselves do not have sharp features and there is motion blur due to the large distance of the tubes from the rotation axis.

Algorithm 6 shows our correction algorithm for the fixtures. Figure A.1 illustrates the algorithm pictorially. The initial reconstruction x suffers from artifacts within the image and at the edge of the field of view. We mask x using a cylindrical mask slightly smaller than the field of view to obtain the masked image x^m. This is done so that the majority of the artifacts at the edge of the field of view are masked but the object remains unchanged in x^m. Consequently the error sinogram $e = y - Ax^m$ primarily contains the projection of the tubes with some residual projection of the object. The blurring of e filters out the residual object projection but preserves the spatially smooth projection of the tubes. The corrected measurements y^{corr} are found after performing a least squares fit. The correction can be repeated in order to get an improved reconstruction x and consequently an improved correction y^{corr}.

Figure A.1 shows the sinogram and reconstruction both before and after performing the blind correction. Not only does the reconstruction after fixture correction remove the artifacts in the air region, but it also improves the image quality inside the object. It can be seen that the vertical stripes in the object in the yz view of the reconstruction have been eliminated after performing the correction.

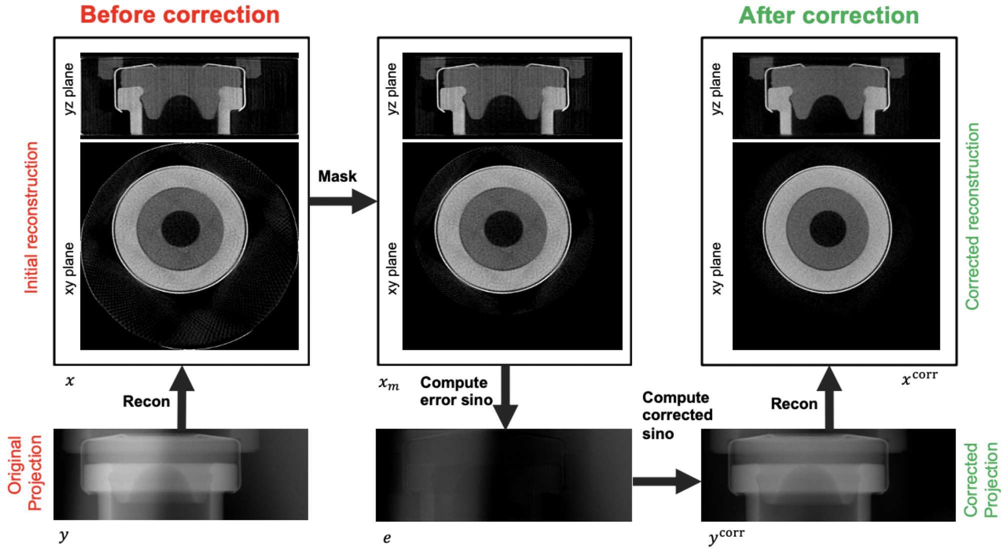


Figure A.1. Pipeline of the blind fixture correction in Algorithm 6. The vertical stripes in the yz-plane of the reconstruction and the ring at the edge of the field of view in the xy-plane of the reconstruction have been rectified after performing the correction.

B. UNIQUENESS PROOF OF INTERLACED VIEW-ANGLES

Theorem B.0.1. *All angles $\theta_i = iK\frac{\pi}{N_\theta}$ are unique (modulo π) if $0 \leq i \leq N_\theta - 1$ and $gcd(K, N_\theta) = 1$*

Proof. Let us assume for the sake of contradiction that there are two integers i, j such that $i \neq j$, $0 \leq i, j \leq N_\theta - 1$, and $\theta_i = \theta_j$ (modulo π).

Using the definition of θ_i, this implies

$$iK = jK + c_1 N_\theta \ , \tag{B.1}$$

where c_1 is an integer constant. Rearranging equation (B.1), we have

$$(i - j)K = c_1 N_\theta \ . \tag{B.2}$$

Now since the left-hand-side of equation (B.2) is a multiple of K, so must be the right-hand-side. However, since $\gcd(K, N_\theta) = 1$, c_1 must be a multiple of K. Let $c_1 = Kc_2$, for some integer constant c_2. Then, equation (B.1) becomes

$$iK = jK + Kc_2 N_\theta \ . \tag{B.3}$$

Dividing by K on both sides give

$$i = j + c_2 N_\theta \ . \tag{B.4}$$

However, our initial assumption of $0 \leq i, j \leq N_\theta - 1$ is a direct contradiction to equation (B.4).

□

Theorem B.0.2. *If K, m, n are integers such that $gcd(K, n) = 1$ then $gcd(K, mK - n) = 1$*

Proof. For the sake of contradiction let us assume $\gcd(K, mK - n) \neq 1$ Thus we have $K = aq$ and $mK - n = bq$ for some integers a, b, q.

Therefore, $n = mK - bq = q(ma - b)$. Thus we have an integer q that divides both K and n making $\gcd(K, n) \neq 1$ and leading to a contradiction.

□

VITA

Soumendu Majee is a Ph.D. candidate at the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA. His PhD advisors are Prof. Charles Bouman and Prof. Gregery Buzzard. He received his B.Tech degree in Electronics and Electrical Communication Engineering from the Indian Institute of Technology, Kharagpur, India, in 2014. He worked briefly at the Advanced Photon Source, Argonne National Laboratory, IL, USA, in 2019. His broad research interests are statistical signal and image processing, computational imaging, and machine learning.