Sparsity-Based Methods for Image Reconstruction and Processing in Cone-Beam Computed Tomography

by

Davood Karimi

M.Sc., The University of Manitoba, 2013

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate and Postdoctoral Studies

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

July 2016

© Davood Karimi 2016

Abstract

X-ray computed tomography (CT) is an essential tool in modern medicine. As the scale and diversity of the medical applications of CT continue to increase, the quest for reducing the radiation dose becomes of extreme importance. However, producing high-quality images from low-dose scans has proven to be a serious challenge. Therefore, further research in developing more effective image reconstruction and processing algorithms for CT is necessary.

This dissertation explores the potential of patch-based image models and total variation (TV) regularization for improving the quality of low-dose CT images. It proposes novel algorithms for 1) denoising and interpolation of CT projection measurements (known as the sinogram), 2) denoising and restoration of reconstructed CT images, and 3) iterative CT image reconstruction.

For sinogram denoising, patch-based and TV-based algorithms are proposed. For interpolation of undersampled projections, an algorithm based on both patch-based and TV-based image models is proposed. Experiments show that the proposed algorithms substantially improve the quality of CT images reconstructed from low-dose scans and achieve state-of-the-art results in sinogram denoising and interpolation.

To suppress streak artifacts in CT images reconstructed from low-dose scans, an algorithm based on sparse representation in coupled learned dictionaries is proposed. Moreover, a structured dictionary is proposed for denoising and restoration of reconstructed CT images. These algorithms significantly improve the image quality and prove that highly effective CT post-processing algorithms can be devised with the help of learned overcomplete dictionaries.

This dissertation also proposes two iterative reconstruction algorithms that are based on variance-reduced stochastic gradient descent. One algorithm employs TV regularization only and proposes a stochastic-deterministic approach for image recovery. The other obtains better results by using both TV and patch-based regularizations. Both algorithms achieve convergence behavior and reconstruction results that are better than widely used itera-

tive reconstruction algorithms compared to. Our results show that variance-reduced stochastic gradient descent algorithms can form the basis of very efficient iterative CT reconstruction algorithms.

This dissertation shows that sparsity-based methods, especially patch-based methods, have a great potential in improving the image quality in low-dose CT. Therefore, these methods can play a key role in the future success of CT.

Preface

This dissertation presents the research conducted by Davood Karimi, with the help and supervision of Prof. Rabab K. Ward. Below is a list of the scientific articles written by Davood Karimi during the course of his doctoral studies at the University of British Columbia.

Part of Chapter 2 was published in paper J7. The work presented in Chapter 3 has been published in paper J6. The contents of Chapter 4 appear in papers J1 and J2. Chapter 6 was published as J5. Chapter 7 was published as C2. Finally, the contents of Chapter 8 appear in J3, J4, and C1.

J1: Davood Karimi, Pierre Deman, Rabab Ward, and Nancy Ford. A sinogram denoising algorithm for low-dose computed tomography. BMC Medical Imaging, 16(11):1–14, 2016.

J2: Davood Karimi and Rabab Ward. A denoising algorithm for projection measurements in cone-beam computed tomography. Computers in Biology and Medicine, 69:71–82, 2016.

J3: Davood Karimi and Rabab Ward. On the computational implementation of forward and back-projection operations for cone-beam computed tomography. Medical & Biological Engineering & Computing, pages 1–12, 2015.

J4: Davood Karimi and RababWard. A hybrid stochastic-deterministic gradient descent algorithm for image reconstruction in cone-beam computed tomography. Biomedical Physics and Engineering Express, 2(1): 015008, 2016.

J5: Davood Karimi and RababWard. Reducing streak artifacts in computed tomography via sparse representation in coupled dictionaries. Medical Physics, 43(3):1473-1486, 2016.

J6: Davood Karimi and RababWard. Sinogram denoising via simultaneous sparse representation in learned dictionaries. Physics in Medicine and Biology, 61(9):3536-53, 2016.

J7: Davood Karimi and Rabab Ward. Patch-based models and algorithms for image processing- a review of the basic principles and methods, and their application in computed tomography. pages 113, 2016.

C1: Davood Karimi, Rabab Ward, and Nancy Ford. A weighted stochastic gradient descent algorithm for image reconstruction in 3D computed tomography. In World Congress on Medical Physics and Biomedical Engineering, June 7-12, 2015, Toronto, Canada, pages 70–73. Springer, 2015.

C2: Davood Karimi and RababWard. A novel structured dictionary for fast processing of 3D medical images, with application to computed tomography restoration and denoising. In SPIE Medical Imaging, 2016.

J1: Davood Karimi is the primary author and the main contributor. Pierre Deman contributed to the design of the experiments. Pierre Deman, Dr. Rabab Ward, and Dr. Nancy Ford provided technical feedback and helped with the writing of the manuscript.

The rest of papers: Davood Karimi is the primary author and the main contributor. Dr. Rabab Ward provided technical feedback and helped with the writing of the papers.

Table of Contents

Abstract ii

Preface iv

Table of Contents vi

List of Tables x

List of Figures xii

Glossary xvi

Acknowledgements xviii

Dedication xix

1 Introduction 1
- 1.1 A brief history of x-ray computed tomography 1
- 1.2 Imaging model 2
- 1.3 Image reconstruction algorithms in CT 4
- 1.4 The goals of this dissertation 6

2 Literature Review 8
- 2.1 Image processing with learned overcomplete dictionaries . . . 9
  - 2.1.1 Sparse representation in analytical dictionaries 9
  - 2.1.2 Learned overcomplete dictionaries 11
  - 2.1.3 Applications of learned dictionaries 17
- 2.2 Non-local patch-based image processing 23
- 2.3 Other patch-based methods 27
- 2.4 Patch-based methods for Poisson noise 28
- 2.5 Total variation (TV) 32
- 2.6 Published research on sparsity-based methods in CT 33
  - 2.6.1 Pre-processing methods 34

2.6.2 Iterative reconstruction methods 36
2.6.3 Post-processing methods 44
2.7 Final remarks 48

3 Sinogram Denoising with Learned Dictionaries 51
3.1 Introduction 51
3.2 The proposed algorithm 53
3.2.1 Clustering 53
3.2.2 Dictionary learning 57
3.2.3 Denoising 59
3.3 Evaluation 60
3.3.1 Simulation experiment 62
3.3.2 Experiment with micro-CT scan of a rat 66
3.3.3 Experiment with micro-CT scan of a phantom 69
3.4 Discussion 72

4 Sinogram Denoising using Total Variation 80
4.1 Introduction 80
4.2 Approach 1- Employing higher-order derivatives 81
4.2.1 The proposed algorithm 81
4.2.2 Simulation experiment 87
4.2.3 Experiments with real micro-CT data 89
4.2.4 Discussion 97
4.3 Approach 2- Locally adaptive regularization 99
4.3.1 The proposed algorithm 99
4.3.2 Simulation experiment 103
4.3.3 Experiment with real micro-CT data 106
4.3.4 Discussion 113

5 Sinogram Interpolation 115
5.1 Introduction 115
5.2 The proposed algorithm 117
5.2.1 Regularization in terms of sinogram self-similarity . . 117
5.2.2 Regularization in terms of sinogram smoothness . . . 119
5.2.3 Optimization algorithm 120
5.3 Results and discussion 121
5.3.1 Experiment with simulated data 121
5.3.2 Experiment with real CT data 123

6 Reducing Streak Artifacts using Coupled Dictionaries . . 126
6.1 Introduction 126
6.2 Methods 127
6.2.1 The proposed approach 127
6.2.2 The dictionary learning algorithm 129
6.3 Evaluation 133
6.4 Discussion 142

7 Two-Level Dictionary for Fast CT Image Denoising 148
7.1 Introduction 148
7.2 The proposed algorithm 149
7.3 Results and discussion 153
7.3.1 Denoising 153
7.3.2 Restoration 154

8 TV-Regularized Iterative Reconstruction 159
8.1 Introduction 159
8.1.1 Motivation and background 159
8.1.2 Formulation of the problem 160
8.1.3 Stochastic gradient descent method 162
8.1.4 Variance-reduced stochastic gradient descent 165
8.2 Methods 167
8.2.1 The proposed algorithm 167
8.2.2 Implementation details 170
8.2.3 Evaluation 173
8.3 Results 174
8.3.1 Simulated data 174
8.3.2 Micro-CT scan of the physical phantom 174
8.3.3 Micro-CT scan of a rat 177
8.4 Discussion 181

9 Iterative Reconstruction with Nonlocal Regularization . . 188
9.1 Introduction 188
9.2 Methods 191
9.2.1 Problem formulation 191
9.2.2 Optimization algorithm 194
9.3 Results and Discussion 196
9.3.1 Simulated data 196
9.3.2 Real data 199

10 Conclusions 202
10.1 Contributions of this dissertation 202
10.1.1 Pre-processing algorithms 202
10.1.2 Post-processing algorithms 203
10.1.3 Iterative reconstruction algorithms 204
10.2 Future work 205
10.2.1 Pre-processing algorithms 205
10.2.2 Post-processing algorithms 206
10.2.3 Iterative reconstruction algorithms 206

Bibliography 208

List of Tables

3.1 Comparison of sinogram denoising algorithms in the projection domain. 62
3.2 Comparison of sinogram denoising and image denoising algorithms on simulated data. 63
3.3 Comparison of sinogram denoising and image denoising algorithms in terms of spatial resolution. 66
3.4 Comparison of sinogram denoising and image denoising algorithms on the micro-CT scan of a rat. 68
3.5 Comparison of sinogram denoising and image denoising algorithms on the micro-CT scan of a physical phantom. 72

4.1 Evaluation of second-order TV sinogram denoising with simulated data in sinogram domain. 88
4.2 Evaluation of second-order TV sinogram denoising with simulated data in image domain. 90
4.3 Evaluation of second-order TV sinogram denoising with micro-CT scan of a phantom. 91
4.4 Evaluation of second-order TV sinogram denoising with micro-CT scan of a rat. 95

5.1 Objective evaluation of image quality improvement with sinogram interpolation on simulated data. 122
5.2 Objective evaluation of image quality improvement with sinogram interpolation on real micro-CT data. 124

6.1 Evaluation of dictionary-based streak artifact suppression on micro-CT scan of a rat. 137
6.2 Evaluation of dictionary-based streak artifact suppression on micro-CT scan of a phantom. 138
6.3 Evaluation of the generalizability of learned parameters in dictionary-based streak artifact suppression. 142

7.1 Evaluation of the two-level learned dictionary with simulation data. 153
7.2 Evaluation of the two-level learned dictionary with micro-CT data. 156

8.1 Objective evaluation of the hybrid stochastic-deterministic reconstruction algorithm with simulation data. 176
8.2 Evaluation of the hybrid stochastic-deterministic reconstruction algorithm with micro-CT data of a phantom. 176
8.3 Evaluation of the hybrid stochastic-deterministic reconstruction algorithm with micro-CT data of a rat. 181

9.1 Image quality metrics for the experiment with simulated data. 196
9.2 Image quality metrics for the experiment with real data. . . . 199

List of Figures

1.1 A schematic representation of cone-beam CT geometry. . . . 3

3.1 Stacking of cone-beam projections to exploit intra-projection and inter-projection correlations. 52
3.2 Demonstration of the effectiveness of the proposed dimensionality reduction mapping for projection signals. 55
3.3 Visual comparison of sinogram denoising and image denoising on simulated data. 64
3.4 Comparison of sinogram denoising and image denoising algorithms in terms of spatial resolution on simulated data. . . . 65
3.5 Performance comparison of sinogram denoising and image denoising on micro-CT scan of a rat. 70
3.6 Visual comparison of sinogram denoising and image denoising on micro-CT data. 73
3.7 Comparison of sinogram denoising and image denoising in terms of spatial resolution on micro-CT data. 74
3.8 Generalizability of the learned dictionary in dictionary-bases sinogram denoising. 77
3.9 Effect of the dictionary size in dictionary-based sinogram denoising. 78
3.10 Visual comparison of dictionary atoms learned from CT projections and CT images. 79

4.1 A visual depiction of the piecewise-smooth nature of CT projections. 82
4.2 Evaluation of second-order TV sinogram denoising with simulated data. 89
4.3 Evaluation of second-order TV sinogram denoising with micro-CT scan of a phantom. 92
4.4 Profiles of the images reconstructed after second-order TV sinogram denoising. 93
4.5 Parameter tuning for second-order TV sinogram denoising. . 94

4.6 Image-domain spatial resolution as influenced by TV-based sinogram denoising. 95
4.7 Evaluation of second-order TV sinogram denoising with micro-CT scan of a rat. 96
4.8 Evaluation of second-order TV sinogram denoising with low-dose micro-CT scan of a rat. 97
4.9 Parameter tuning for first and second-order TV sinogram denoising. 98
4.10 Visual evaluation of locally-adaptive TV sinogram denoising on simulated data. 104
4.11 Objective evaluation of locally-adaptive TV sinogram denoising on simulated data. 105
4.12 Objective evaluation of locally-adaptive TV sinogram denoising on micro-CT scan of a phantom. 107
4.13 Visual evaluation of locally-adaptive TV sinogram denoising on micro-CT scan of a phantom. 108
4.14 Profiles of the images reconstructed after locally-adaptive TV sinogram denoising. 109
4.15 Trade-off between noise and spatial resolution for locally-adaptive TV sinogram denoising. 110
4.16 Objective evaluation of locally-adaptive TV sinogram denoising on micro-CT scan of a rat. 111
4.17 Visual evaluation of locally-adaptive TV sinogram denoising on micro-CT scan of a rat. 111
4.18 Trade-off between noise and spatial resolution for locally-adaptive TV sinogram denoising evaluated on micro-CT scan of a rat. 112

5.1 A schematic representation of the sinogram interpolation problem. 116
5.2 Block matching between the undersampled scan to be interpolated and a high-dose reference scan. 118
5.3 Effect of sinogram interpolation on the quality of brain phantom images. 123
5.4 Effect of sinogram interpolation on the quality of rat images reconstructed from micro-CT data. 125

6.1 Schematic representation of dictionary-based algorithm for streak artifact suppression. 129

6.2 Evaluation of dictionary-based algorithm for streak artifact suppression on training data. 134
6.3 Evaluation of dictionary-based algorithm for streak artifact suppression on test data. 136
6.4 Visual evaluation of dictionary-based algorithm for streak artifact suppression on micro-CT scan of a phantom. 139
6.5 Assessment of generalizability of the parameters of dictionary-based algorithm for streak artifact suppression. 141
6.6 Visual comparison of two models with different complexities for dictionary-based streak artifact suppression. 143

7.1 Schematic representation of two-level dictionary. 150
7.2 Visual inspection of the denoising performance of two-level dictionary on simulation data. 154
7.3 Visual inspection of the denoising performance of two-level dictionary on micro-CT data. 155
7.4 Visual inspection of the performance of two-level dictionary for suppressing ring artifacts. 157
7.5 Effect of dictionary size and sparsity level on the performance of the two-level dictionary for suppressing ring artifacts. . . . 158

8.1 Convergence behavior of the stochastic-deterministic iterative reconstruction on simulation data. 175
8.2 Visual inspection of the reconstruction results of the stochastic-deterministic iterative reconstruction on simulation data. . . 177
8.3 Convergence of the stochastic-deterministic iterative reconstruction on micro-CT scan of a phantom. 178
8.4 Reconstruction results of the stochastic-deterministic iterative reconstruction on micro-CT scan of a phantom. 179
8.5 Convergence of the stochastic-deterministic iterative reconstruction on micro-CT scan of a rat. 180
8.6 Reconstruction results of the stochastic-deterministic iterative reconstruction on micro-CT scan of a rat. 182
8.7 Effects of batch size and regularization parameter on the convergence of stochastic-deterministic iterative reconstruction. . 185
8.8 Effects of batch size and regularization parameter on the reconstruction results of stochastic-deterministic iterative reconstruction. 186

9.1 The proposed initialization for the Generalized PatchMatch algorithm. 193
9.2 Reconstruction results for the experiment with the brain phantom. 197
9.3 RMSE plots for reconstruction of the brain phantom. 198
9.4 Reconstruction results of the experiment with the rat scan. . 200

Glossary

ANLM	Adaptive non-local means
CBCT	Cone-beam computed tomography
CNR	Contrast to noise ration
CT	Computed tomography
DCT	Discrete cosine transform
FBP	Filtered back-projection
FDK	Feldkamp-Davis-Kress
FGD	Full gradient descent
GPU	Graphical processing unit
MAP	Maximum a posteriori
MI	Mutual information
MRI	Magnetic resonance imaging
MTF	Modulation transfer function
NLM	Non-local means
NS	Noise strength
OMP	Orthogonal matching pursuit
PCA	Principal component analysis
PET	Positron emission tomography
PSNR	Peak signal to noise ration
RMSE	Root mean square error

ROF	Rudin-Osher-Fatemi
ROI	Region of interest
SAG	Stochastic average gradient
SGD	Stochastic gradient descent
SNR	Signal to noise ration
SR	Spatial resolution
SSIM	structural similarity index
SVD	Singular value decomposition
SVRG	Stochastic variance-reduced gradient
TV	Total variation
VR-SGD	Variance-reduced stochastic gradient descent

Acknowledgements

I would like to sincerely thank my thesis advisor, Prof. Rabab Ward, for all her support, guidance, advice, and encouragement during the course of my Ph.D. studies.

All of the micro-CT data that were used in this dissertation were provided by the Centre for High-Throughput Phenogenomics at the University of British Columbia. I am very grateful to Dr. Nancy Ford, the Director of the Centre, for her constant support. I am also very thankful to Mr. John Schipilow and Dr. Pierre Deman who carried out most of the micro-CT scanning.

Sincere thanks to the members of my thesis committee: Dr. Nancy Ford, Dr. Vikram Krishnamurthy, and Dr. Jane Wang, the University Examiners: Dr. Shuo Tang and Dr. Ozgur Yilmaz, the Chair of the Final Oral Defence: Dr. Anna Celler, and the External Examiner: Dr. Charles Bouman of Purdue University, for their valuable feedback that helped improve the research presented in this dissertation.

I was supported by the University of British Columbia through a Four-Year Fellowship (4YF) and by the Natural Sciences and Engineering Research Council of Canada (NSERC) through an Alexander Graham Bell Canada Graduate Scholarship (CGS-D). Their support is greatly appreciated.

Greatest thanks to my family for what words cannot describe.

Dedication

To my parents, for their dedication.

Chapter 1

Introduction

1.1 A brief history of x-ray computed tomography

Computed tomography (CT) refers to creating images of the cross sections of an object using transmission or reflection data. These data are usually referred to as the *projections* of the object. For the projection data to be sufficient for reconstructing the object's image, the object needs to be illuminated from many different directions. The problem of reconstructing the image of an object from its projections has various applications, from reconstructing the structure of molecules from data collected with electron microscopes to reconstructing maps of radio emissions of celestial objects from data collected with radio telescopes [129]. However, the most important applications of CT have been in the field of medicine, where the impact of CT has been nothing short of revolutionary. Today, physicians and surgeons are able to view the internal organs of their patients with a precision and safety that was impossible to imagine before the advent of CT.

The fundamental difference between different medical imaging modalities is the property of the material (i.e., tissue) that they image. X-ray CT, which is the focus of this dissertation, is based on the tissue's ability to attenuate x-ray photons. X-rays had been discovered by the German physicist Wilhelm Rontgen in 1895. Rontgen, who won the first Nobel Prize in Physics for this discovery, realized that x-rays could reveal the skeletal structure of the body parts because bones and soft tissue had different x-ray attenuation properties. However, the first commercial CT scanners appeared in the early 1970s, finally winning the 1979 Nobel Prize in Medicine for Allan Cormack and Godfrey Hounsfield for independently inventing CT.

Today, x-ray CT is an indispensable tool in medicine. In fact, the words CT and computed tomography are used to refer to x-ray CT with no confusion. In the rest of this dissertation, too, the qualifier "x-ray" is dropped, assuming the implicit knowledge that the whole dissertation is devoted to x-ray computed tomography. Since its commercial introduction more than

40 years ago, diagnostic and therapeutic applications of CT have continued to grow. In the past two decades especially, great advancements have been made in CT scanner technology and the available computational resources. Moreover, new scanning methods such as dual-source and dual-energy CT have become commercially available. Today, very fast scanning of large volumes has become possible. This has led to a dramatic increase in CT usage in clinical settings. It is estimated that globally more than 50,000 dual-energy x-ray CT scanners are in operation [290]. In the USA alone, the number of CT scans made annually increased from 19 million to 62 million between 1993 and 2006 [226].

1.2 Imaging model

Although the algorithms proposed in this dissertation apply to most or all CT geometries, the focus of this dissertation is on cone-beam computed tomography (CBCT). CBCT is a relatively new scan geometry that has found applications as diverse as image-guided radiation therapy, dentistry, breast CT, and microtomography [52, 108, 137, 283]. Figure 1.1 shows a schematic representation of CBCT. Divergent x-rays penetrate the object and become attenuated before being detected by an array of detectors. The equation relating the detected photon number to the line integral of the attenuation coefficient is [269]:

$$\frac{N_d^i}{N_0^i} = \exp\left(-\int_i \mu ds\right) \tag{1.1}$$

where N_0^i and N_d^i denote, respectively, the emitted and detected photon numbers for the ray from the x-ray source to the detector bin i and $\int_i \mu ds$ is the line integral of the attenuation coefficient (μ) along that ray. By discretizing the imaged object, the following approximation to (1.1) can be made:

$$\log\left(\frac{N_0^i}{N_d^i}\right) = \sum_{k=1}^{K} a_{i,k}\, x_k \tag{1.2}$$

where x_k is the value of the unknown image at voxel k and $a_{i,k}$ is the length of intersection of ray i with this voxel. The equations for all measurements can be combined and conveniently written in matrix form as:

$$y = Ax + w \tag{1.3}$$

where y represents the vector of measurements (also known as the sinogram), x is the unknown image, A represents the projection matrix, and w is the measurement noise.

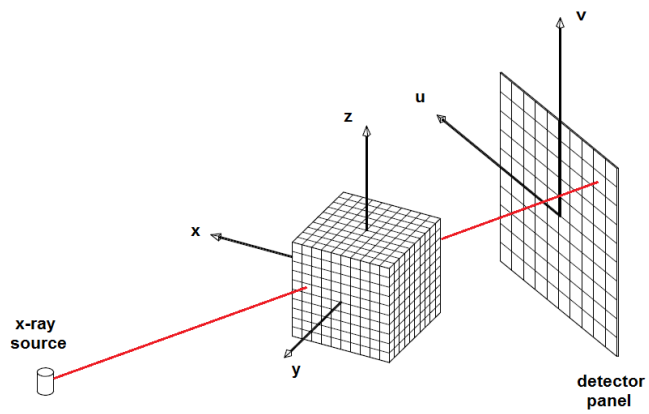

Figure 1.1: A schematic representation of cone-beam CT geometry.

The discretization approach mentioned above has several shortcomings. For example, it does not consider the finite size of the x-ray source and the detector area. Furthermore, exact computation of the intersection lengths of rays with voxels is computationally very costly for large-scale 3D CT. Therefore, several efficient implementations of the system matrix A have been proposed [78, 185, 193, 221]. For large-scale 3D CT, matrix A is too large to be saved in computer memory. Instead, these algorithms implement multiplication with matrix A and its transpose by computing the matrix elements on-the-fly.

Even though in theory N_d follows a Poisson distribution, due to many complicating factors including the polychromatic nature of the x-ray source and the electronic noise, an accurate model of the raw data takes the form of a compound Poisson, shifted Poisson, or Poisson+Gaussian distribution [174]. For many practical applications, an adequate noise model is obtained by adding a Gaussian noise (to simulate the electronic noise) to the theoretical values of N_d. More realistic modeling, especially in low-dose CT, is much more complex and will need to take into account very subtle phenomena, which are the subject of much research [244, 312, 351]. An alternative approach is to consider the ratio of the photon counts after the logarithm transformation. Even though N_0^i and N_d^i are Poisson distributed, the noise in the sinogram (i.e., after the logarithm transformation) can be modeled as

a Gaussian-distributed random variable with zero mean and a variance that follows [201, 204, 326]:

$$\sigma_i^2 = \frac{\exp(\bar{y}_i)}{N_0^i} \tag{1.4}$$

In this equation, $\bar{y}_i$ is the expected value of the sinogram datum at detector i. In general, a system-specific constant η is needed to fit the measurements [326]:

$$\sigma_i^2 = f_i \exp\left(\frac{\bar{y}_i}{\eta}\right) \tag{1.5}$$

where f_i, similar to $1/N_0^i$ in (1.4), mainly accounts for the effect of bowtie filtration.

1.3 Image reconstruction algorithms in CT

A central component in every CT system is the suite of image reconstruction and processing algorithms, whose task it to reconstruct the image of the object from its projection measurements. These algorithms have also continually evolved over time. The first CT scanners relied on simple iterative algorithms that aimed at recovering the unknown image as a solution of a system of linear equations. Many of these basic iterative methods had been developed by mathematicians like Kaczmarz well before the advent of CT. As the size of CT images grew, analytical filtered-backprojection (FBP) methods became more common and they are still widely used in practice [247]. For CBCT, the well-known Feldkamp-Davis-Kress (FDK) filtered-backprojection algorithm is still widely used [107, 307, 310]. These methods, which are based on the Fourier slice theorem, require a large number of projections to produce a high-quality image, but they are much faster than iterative methods.

The speed advantage of FBP methods has become less significant in recent years as the power of personal computers has increased and new hardware options such as graphical processing units (GPUs) have become available. On the other hand, with a consistent growth in medical CT usage, many studies have shown that the radiation dose levels used in CT may be harmful to the patients [24, 299]. Reducing the radiation dose can be accomplished by reducing the number of projection measurements and/or by reducing the radiation dose for each projection. However, the images reconstructed from such under-sampled or noisy measurements with FBP

methods will have a poor diagnostic quality. As a result of these developments, there has been a renewal of interest in statistical and iterative image reconstruction methods because they have the potential to produce high-quality images from low-dose scans [19, 149]. Furthermore, even though in the beginning most of the algorithms used in CT were image reconstruction algorithms, gradually image processing algorithms were used for denoising, restoration, or otherwise improving the projection measurements and the reconstructed images. Many of these algorithms are borrowed from the research on image processing for natural images. Even today, algorithms that have been developed for processing of natural images are often applied in CT with little or no modifications.

Turning to more effective image reconstruction and processing algorithms is not the only approach to radiation dose reduction. There are indeed other approaches, such as improving the system hardware and imaging protocols [222, 223]. However, there are very strong additional reasons that encourage research on better algorithms for CT. To begin with, the advantage of iterative/statistical reconstruction algorithms is not limited to low-dose CT. There are other situations where FBP methods fail and one has to resort to more sophisticated iterative/statistical reconstruction methods. Examples include non-standard scanning geometries such as those with irregular or limited angular sampling (e.g., in tomosynthesis) or when some of the measurements are missing or corrupted. Another important factor, as we mentioned above, is the increased availability of high-performance computational hardware. The potential of these advancements in computer hardware for CT has begun to be understood and there is a growing body of research to investigate the significance of this increased computational power for CT [144, 224, 261, 272]. Lastly, a very important factor that is more related to the subject of this dissertation, is the introduction of new theories and methods in signal and image processing and applied mathematics that can lead to more powerful and more flexible algorithms for CT. For example, new optimization algorithms that have much faster theoretical convergence rates have led to state-of-the-art image reconstruction algorithms in recent years [160, 263]. Another example of recent advancements in signal and image processing that has already had a great impact on CT is the new developments in sparsity-based models. In these models, the image is transformed from its native representation in terms of pixel/voxel values into a different space where it has a more concise and more meaningful representation. It is hoped that such a representation provides a more effective description of the relevant image features, thereby improving the achievable results in various image processing tasks. Even though this is an old idea in

image processing, recent decades and years have witnessed the emergence of new models and algorithms that have called for a reassessment of the potential of sparsity-based methods in CT.

1.4 The goals of this dissertation

The goal of this dissertation is to advance the state of the art in the application of sparsity-based methods in low-dose CT. Even though sparsity-based methods have been widely used for image reconstruction and processing in CT, more recent sparsity-based models and optimization algorithms have the potential to substantially improve the current state of the art. The goal of this dissertation is to make significant contributions in this direction. Some of the novelties of the research that is reported in this dissertation are summarized below.

- This dissertation relies heavily on learned overcomplete dictionaries. Compared with analytical dictionaries such as wavelets, learned dictionaries have a much higher representational power and flexibility. In recent years, these dictionaries have been shown to achieve state-of-the-art results in many image processing tasks. However, the potential of learned dictionaries for CT has not been fully appreciated. This dissertation tries to explore this potential. This dissertation also draws heavily upon other patch-based models and algorithms, especially those that exploit nonlocal patch similarities. Patch-based models have emerged as one of the most powerful models in image processing in the past decade. However, little research has been reported on the application of these models in CT. This dissertation tries to investigate the potential of patch-based methods for image reconstruction and processing in CT.

- The great majority of the published research have focused on iterative reconstruction algorithms and image-domain post-processing algorithms. Comparatively, many fewer studies have been reported on denoising, restoration, or otherwise improving the projection measurements. In this dissertation, we pay particular attention to this gap in research.

- For iterative image reconstruction, this dissertation makes use of new stochastic optimization algorithms. Stochastic/incremental optimization methods have been used to accelerate various CT reconstruction

algorithms over the past two decades. This dissertation shows that the new class of variance-reduced stochastic gradient descent algorithms are superior to the traditional stochastic optimization methods for CT reconstruction.

Chapter 2

Literature Review

This chapter starts by reviewing the basic principles of the three main image models that are used in this dissertation. These models are based on sparse representation in learned dictionaries, nonlocal patch-based models, and total variation (TV). Then, example studies that have used these models for image reconstruction and processing in CT are reviewed.

The first two of the three image models mentioned above belong to *patch-based* models. In patch-based image processing, the units upon which operation are carried are small image patches, which in the case of 3D images are also referred to as blocks. In the great majority of applications square patches or cubic blocks are used, even though other patch shapes can also be employed. For simplicity of presentation, we will use the term "patch" unless when talking explicitly about 3D images. The number of pixels/voxels in a patch in patch-based image processing methods is usually on the order of tens or a few hundreds. A typical patch size would be 8×8 pixels for 2D images or $8 \times 8 \times 8$ voxels for 3D images.

Broadly speaking, in patch-based methods the image is first divided into small patches. Then, each patch is processed either separately on its own or jointly with patches that are very similar to it. The final output image is then formed by assembling the processed patches. In patch-based denoising, for instance, one can divide the image into small overlapping patches, denoise each patch independently, and then build the final denoised image by averaging the denoised patches. There are many reasons for focusing on small patches rather than on the whole image. First, because of the curse of dimensionality, it is much easier and more reliable to learn a model for small image patches than for very large patches or for the whole image. Secondly, for many models, computations are significantly reduced if they are applied on small patches rather than on the whole image. In addition, research in the past decade has shown that working with small patches can result in very effective algorithms that outperform competing methods in a wide range of image processing tasks. For example, as we will explain later in this chapter, patch-based denoising methods are currently considered to be the state of the art, achieving close-to-optimal denoising performance.

Patch-based methods have been among the most heavily researched methods in the field of image processing in recent years and they have produced state-of-the-art results in many tasks including denoising, restoration, super-resolution, inpainting, and reconstruction. However, these methods have received very little attention in CT. Even though there has been a limited effort in using patch-based methods in CT, the results of the published studies have been very promising. Given the great success of patch-based methods in various image-processing applications, they seem to have the potential to substantially improve the current state of the art algorithms in CT.

The word "patch-based" may be ambiguous because it can potentially refer to any image model or algorithm that works with small patches. For example, image compression algorithms such as JPEG work on small image patches. However, the word patch-based has recently been used to refer to certain classes of methods. In order to explain the central concepts of these methods, we will first describe the two main frameworks in patch-based image processing: (1) sparse representation of image patches in learned overcomplete dictionaries, (2) models based on nonlocal patch similarities. These two frameworks do not cover all patch-based image processing methods. However, most of these methods have their roots in one or both of these two frameworks.

2.1 Image processing with learned overcomplete dictionaries

2.1.1 Sparse representation in analytical dictionaries

A signal $x \in \mathbb{R}^m$ is said to have a sparse representation in a dictionary $D \in \mathbb{R}^{m \times n}$ if it can be accurately approximated by a linear combination of a small number of its columns. Mathematically, this means that there exists a vector γ such that $x \cong D\gamma$ and $\|\gamma\|_0 \ll n$. Here, $\|\gamma\|_0$ denotes the the number of nonzero entries of γ and is usually referred to as the ℓ_0-norm of γ, although it is not a true norm. This means that only a small number of columns of D are sufficient for accurate representation of the signal x. The ability to represent a high-dimensional signal as a linear combination of a small number of building blocks is a very powerful concept and it is at the center of many of the most widely used algorithms in signal and image processing. Columns of the dictionary D are commonly referred to as atoms. If these atoms comprise a set of linearly independent vectors and

if they span the whole space of $\mathbb{R}^m$, then they are called basis vectors and D is called a basis. Moreover, if the basis vectors are mutually orthogonal, D is called an orthogonal basis.

Bases, and orthogonal bases in particular, have interesting analytical properties that makes them easy to analyze. Moreover, for many of the orthogonal bases that are commonly used in signal and image processing, very fast computational algorithms have been developed. This computational advantage made these bases very appealing when the computational resources were limited. Over the past two decades, and especially in the past decade, there has been a significant shift of interest towards dictionaries that can adapt to a given class of signals using a learning strategy. The dictionaries obtained in this way lack the analytical and computational advantages of orthogonal bases, but they have much higher representational power. Therefore, they usually lead to superior results for many image processing tasks. Before we explain these dictionaries, we briefly review the history of sparsity-inducing transforms in image processing. More detailed treatment of this background can be found in [218, 273].

Sparsity-based models are as old as digital signal processing itself. Starting in the 1960s, the Fourier transform was used in signal processing because it could diagonalize the linear time-invariant filters, which were widespread in signal processing. Adoption of the Fourier transform was significantly accelerated by the invention of the Fast Fourier Transform in 1965 [138]. Fourier transform represents a signal as a sum of sinusoids of different frequencies. Suppressing the high-frequency components of this representation, for example, is a simple denoising method. This is, however, not a good model for natural images because these Fourier basis functions are not efficient for representing sharp edges. In fact, a single edge results in a large number of non-zero Fourier coefficients. Therefore, denoising using Fourier filtering leads to blurred images. An efficient representation of localized features needs bases that include elements with concentrated support. This gave rise to the Short-Time Fourier Transform (STFT) [5, 16] and, more importantly, the wavelet transform [74, 217]. The wavelet transform was the major manifestation of a revolution in signal processing that is referred to as multi-scale or multi-resolution signal processing. The main idea in this paradigm is that many signals, and in particular natural images, contain relevant features on many different scales. Both the Fourier transform and the wavelet transform can be interpreted as the representation of a signal in a dictionary. For the Fourier transform, for example, the dictionary atoms include sinusoids of different frequencies.

Despite its tremendous success, the wavelet transform suffers from im-

portant shortcomings for analyzing higher-dimensional signals such as natural images. Even though the wavelet transform possesses important optimality properties for one-dimensional signals, it is much less effective for higher-dimensional signals. This is because in higher dimensions, the wavelet transform is a separable extension of the one-dimensional transform along different dimensions. As a result, for example the 2D wavelet transform is suitable for representing points but it is not effective for representing edges. This is a major shortcoming because the main features in natural images are composed of edges. Therefore, there was a need for sparsity-inducing transforms or dictionaries that could efficiently represent these types of features. Consequently, great research effort was devoted to designing transforms/dictionaries especially suitable for natural images. Some of the proposed transforms that have been more successful for image processing applications include the complex wavelet transform [164], the curvelet transform [41, 42], the contourlet transform [86] and its extension to 3D images known as surfacelet [198], the shearlet transform [96, 171], and the bandlet transform [176].

The transforms mentioned above have had a great impact on the field of image processing and are still used in practice. They have also been used in CT [e.g., 41, 110, 266, 321]. However, learned overcomplete dictionaries achieve much better results in practice by breaking some of the restrictions that are naturally imposed by these analytical dictionaries. The restriction of orthogonality, for instance, requires the number of atoms in the dictionary to be no more than the dimensionality of the signal. The consequences of these limitations had already been realized by researchers working on wavelets. This realization led to developments such as stationary wavelet transform, steerable wavelet transform, and wavelet packets, which greatly improved upon the orthogonal wavelet transform [64, 230, 295]. However, these transforms are still based on fixed constructions and do not have the freedom and adaptability of learned dictionaries that we will explain below.

2.1.2 Learned overcomplete dictionaries

The basic idea of adapting the dictionary to the signal is not completely new. One can argue that the Principal Component Analysis (PCA) method [131], which is also known as the Karhunen–Love Transform (KLT) in signal processing, is an example of learning a dictionary from the training data. However, this transform also is limited in terms of the dictionary structure and the number of atoms in the dictionary. Specifically, the atoms in a PCA dictionary are necessarily orthogonal and their number is at most equal to

the signal dimensionality.

The modern story of dictionary learning begins with a paper by Olshausen and Field [245]. The question posed in that paper was: if we assume that small patches of natural images have a sparse representation in a dictionary D and try to learn this dictionary from a set of training patches, what would the learned dictionary atoms look like? They found that the learned dictionary consisted of atoms that were spatially localized, oriented, and bandpass. This was a remarkable discovery because these are exactly the characteristics of simple-cell receptive fields in the mammalian visual cortex. Although similar patterns existed in Gabor filters [75, 76], Olshausen and Field had been able to show that these structures can be explained using only one assumption: sparsity.

Suppose that we are given a set of training signals and would like to learn a dictionary for sparse representation of these signals. We stack these training signals as columns of a matrix, which we denote with X. Each column of X is a referred to as a training signal. In image processing applications, each training signal is a patch (for 2D images) or block (in the case of 3D images) that is vectorized to form a column of X. Using the matrix of training signals, a dictionary can be learned through the following optimization problem.

$$\underset{D\in\mathscr{D},\Gamma}{\textbf{minimize}}\, \|X - D\Gamma\|_F^2 + \lambda\|\Gamma\|_1 \tag{2.1}$$

In the above equation, X denotes the matrix of training data, Γ is the matrix of representation coefficients of the training signals in D, and $\mathscr{D}$ is the set of matrices whose columns have a unit Euclidean norm. The ith column of Γ is the vector of representation coefficients of the ith column of X (i.e., the ith training signal) in D. The notations $\|.\|_F$ and $\|.\|_1$ denote, respectively, the Frobenius norm and the ℓ_1 norm. The constraint $D \in \mathscr{D}$ is necessary to avoid scale ambiguity because without this constraint the objective function can be made smaller by decreasing Γ by an arbitrary factor and increasing D by the same factor. The first term in the objective function requires that the training signals be accurately represented by the columns of D and the second terms promotes sparsity, encouraging that a small number of columns of D are used in the representation of each training signal.

There are many possible variations of the optimization problem presented in Equation (2.1), some of which will be explained in this chapter. For example the ℓ_1 penalty on Γ is sometimes replaced with an ℓ_0 penalty. In fact, it can be shown that variations of this problem include problems as

diverse as PCA, clustering or vector quantization, independent component analysis, archetypal analysis, and non-negative matrix factorization (see for example [9, 211]). The most important fact about the optimization problem in (2.1) is that it is not jointly convex with respect to D and Γ. Therefore, only a stationary point can be hoped for and the global optimum is not guaranteed. However, this problem is convex with respect to D and Γ individually. Therefore, many dictionary learning problems adopt an alternating minimization approach. In other words, the objective function is minimized with respect to one of the two variables while keeping the other fixed. The first such method was the method of optimal directions (MOD) [103]. In each iteration of MOD, the objective function is first minimized with respect to Γ by solving a separate sparse coding problem for each training signal:

$$\Gamma_i^{k+1} = \underset{\gamma}{\textbf{argmin}} \, \|X_i - D^k \gamma\|_2^2 \quad \textbf{subject to:} \quad \|\gamma\|_0 \leq K \tag{2.2}$$

In the above equation, and in the rest of this chapter, we use subscripts on matrices to index their columns. Therefore, X_i indicates the ith column of X, which is the ith training signal and Γ_i is the ith column of Γ, which is the vector of representation coefficients of X_i in D. We will use superscripts to indicate iteration number. Once all columns of Γ are updated, Γ is kept fixed and the dictionary is updated. This update is in the form of a least-squares problem that has a closed-form solution:

$$D^{k+1} = X(\Gamma_i^{k+1})^\dagger \tag{2.3}$$

where $\dagger$ denotes the Moore-Penrose pseudo-inverse.

Before moving on, we need to say two brief words about the optimization problem in (2.2). This optimization problem is one formulation of the sparse coding problem that is a central part of any image processing method that makes use of learned overcomplete dictionaries. Because of their ubiquity, there has been a very large body of research on the properties of these problems and solution methods. Often the ℓ_0 norm in this equation is replaced with an ℓ_1 norm. This problem is known as Least Absolute Shrinkage and Selection Operator (LASSO) in statistics [315] and Basis Pursuit Denoising (BPDN) in signal processing [54]. A review of these methods is beyond the scope of this dissertation. Therefore, we will only mention or describe the relevant algorithms where necessary. A recent review of these methods can be found in [9]. In MOD, this step is solved using the orthogonal matching pursuit (OMP) [252] or the focal underdetermined system solver (FOCUSS) [119].

Another dictionary-learning algorithm that has shown to be more efficient than MOD is the K-SVD algorithm [2]. K-SVD is arguably the most widely used dictionary learning algorithm today. Similar to MOD, each iteration of the K-SVD algorithm updates each column of Γ by solving a sparse coding problem similar to (2.2). However, unlike the MOD that updates all dictionary atoms at once, K-SVD updates each dictionary atom (i.e., each column of D) sequentially. Assuming all dictionary atoms are fixed except for the ith atom, the cost function in (2.1) can be written as:

$$\begin{aligned} \|X - D\Gamma\|_F^2 &= \left\| X - \sum_{j=1}^{N} D_j \Gamma_j^T \right\|_F^2 \\ &= \left\| X - \sum_{j=1, j\neq i}^{N} D_j \Gamma_j^T - D_i \Gamma_i^T \right\|_F^2 = \|E_i - D_i \Gamma_i^T\|_F^2 \end{aligned} \tag{2.4}$$

In the K-SVD algorithm this is minimized using an SVD decomposition of the matrix E_i after restricting it to the training signals that are using D_i in their representation. The reason behind this restriction is that it will preserve the sparsity of the representation coefficients. Let us denote the restricted version of E_i with E_i^R and assume that the SVD decomposition of E_i^R is $E_i^R = U\Delta V^T$. Then, U_1 and $\Delta(1,1)V_1$ provide the updates of D_i and Γ_i^T, where U_1 and V_1 denote the first columns of U and V, respectively, and $\Delta(1,1)$ is the largest singular value of E_i^R.

A major problem with methods like MOD and K-SVD is that they are computationally intensive. Even though efficient implementations of these algorithms have been developed [275], the amount of computations becomes very excessive when the number of training signals and the signal dimensionality grow. Therefore, a number of studies have proposed algorithms that are particularly designed for learning dictionaries from huge datasets in reasonable time [213, 313]. The algorithm proposed in [212, 213], for example, is based on stochastic optimization algorithms that are particularly suitable for large-scale problems. Instead of solving the optimization problem by considering the whole training data, it randomly picks one training signal (i.e., one column of X) and approximately minimizes the objective function using that one training signal. Convincing theoretical and empirical evidence regarding the convergence of this dictionary learning approach have been presented in [213].

Another important class of dictionary learning algorithms are maximum-likelihood algorithms, which are in fact among the first methods suggested

for learning dictionaries from data [167, 183, 184]. These methods assume that each training signal is produced by a model of the form:

$$X_i = D\Gamma_i + w_i \tag{2.5}$$

where w_i is a Gaussian-distributed white noise. To encourage sparsity of the representation coefficients (Γ_i), these methods assume a sparsity-promoting prior such as a Cauchy or Laplace distribution for entries of Γ. Additionally, these approaches assume that the entries of Γ are independent and identically distributed and that each signal X_i is drawn independently. A dictionary can then be learned by maximizing the data likelihood $p(X|D)$ or the posterior $p(D|X)$. Quite often, the resulting likelihood function is very difficult to maximize and it is further simplified before applying the optimization algorithm.

It can be argued that the maximum-likelihood methods explained above are not truly Bayesian methods because they yield a point estimate rather than the full posterior distribution [211]. As a result, in recent years several fully-Bayesian dictionary learning methods have been proposed [135, 358, 359]. In these algorithms, priors are placed on all model parameters, i.e., not only on the dictionary atoms D_i and sparse representation vectors Γ_i, but also on all other model parameters such as the number of dictionary atoms and the noise variance for each training signal. The most important priors assumed in these models are usually Gaussian priors with Gamma hyper-priors for the dictionary atoms (D_i) and representation coefficients (Γ_i), and a Beta-Bernoulli process for the support of Γ_i [135, 359]. Full posterior density of the model parameters and hyper-parameters are iteratively estimated via Gibbs sampling. Compared to all other dictionary learning methods described above, these fully-Bayesian methods are significantly more computationally demanding. On the other hand, their robustness with respect to poor initialization and their ability to learn some important parameters such as the noise variance makes them potentially very useful for certain applications [304, 337].

There are many variations and enhancements of dictionary learning that we cannot describe in detail due to space limitations. However, we briefly mention three important variations. The first is the structured dictionary learning. The main idea here is not only to learn the dictionary atoms but also the interaction between the learned dictionary atoms. For example, a common structure that is assumed between the atoms is a tree structure, where each atom is the descendant/parent of some other atoms [141, 142]. During the dictionary usage, then, an atom will participate in the sparse

code of a signal if and only if its parent atom does so. Obviously, the basic ℓ_0 and ℓ_1 norms are not capable of modelling these interactions between dictionary atoms. The success of structured dictionary learning, therefore, has been made possible by algorithms for structured sparse coding [134, 140]. Another common structure is the grid structure that enforces a neighborhood relation between atoms [153, 313]. The second variation that is of great importance is multi-scale dictionary learning. Extending the basic dictionary learning scheme to consider different patch sizes has been shown to significantly improve the performance of the dictionary-based image processing [216, 246]. Moreover, this extension to multiple scales has been suggested as an approach to addressing some of the theoretical flaws in the dictionary-based image processing [250]. The third important variation that we mention here includes dictionaries that have a fast application. As we mentioned above, learned dictionaries do not possess such desired structural properties as orthogonality. As a result, they are much more costly to apply than analytical dictionaries. Therefore, several dictionary structures have been proposed with the goal of reducing the computational cost during dictionary usage [1, 274]. These dictionaries can be particularly useful for processing of large 3D images.

As final remarks on dictionary learning, we should first mention that there is no strong theoretical justification behind most dictionary learning algorithms. In particular, there is no theoretical guarantee that these algorithms are robust or that the learned dictionary should work well in practical applications. In practice, learning a good dictionary certainly requires a sufficient amount of training data and the minimum amount of data needed grows at least linearly with the number of dictionary atoms [274, 276]. Uniqueness of the learned dictionary, however, is only guaranteed for an exponential number of training signals [3]. In fact, the theory of dictionary learning is considered to be one of the major open problems in the field of sparse representation [98]. Secondly, pre-processing of training image patches has proved to significantly influence the types of structures that emerge in the learned dictionary and the performance of the learned dictionary in practice. Three of the most commonly used pre-processing operations include: (i) removing of the patch mean, also known as centering [100], (ii) variance normalization which is preceded with centering [139, 258], and (iii) de-correlating the pixel values within a patch, referred to as whitening [20, 136]. The overall effect of all these three operations is to amplify the high-frequency structure such as edges, resulting in more high-frequency patterns in the learned dictionary [211].

2.1.3 Applications of learned dictionaries

Learned overcomplete dictionaries have been employed in various image processing and computer vision applications in the past ten years. There are monographs that review and explain these applications in detail [99, 211]. Because of space limitations, we describe the basic formulations for image denoising, image inpainting, and image scale-up. Not only these three tasks are among the most successful applications of learned dictionaries in image processing, they are also very instructive in terms of how these dictionaries can be used to accomplish various image processing tasks.

Image denoising

Suppose that we have measured a noisy image $x = x_0 + w$, where x_0 is the true underlying image and w is the additive noise that is assumed to be white Gaussian. The prior assumption in denoising using a dictionary D is that every patch in the image has a sparse representation in D. If we denote a typical patch with p, this would mean that there exists a sparse vector γ such that $||p - D\gamma||_2^2 < \epsilon$, where ϵ is proportional to the noise variance [100, 215]. Using this prior on every patch in the image, the maximum a posteriori (MAP) estimation of the true image can be found as the solution of the following problem [99]:

$$\left\{\widehat{x_0}, \{\widehat{\gamma}_i\}_{i=1}^N\right\} = \underset{z,\{\gamma_i\}_{i=1}^N}{\mathbf{argmin}}\, \lambda\|z - x\|_2^2 + \sum_{i=1}^N \left(\|R_i z - D\gamma_i\|_2^2 + \|\gamma_i\|_0\right) \quad (2.6)$$

where R_i represents a binary matrix that extracts and vectorizes the ith patch from the image. This is a very common notation that is used to simplify the presentation of this type of equations and we will use it in the rest of this dissertation. N is the total number of extracted patches. It is common to use overlapping patches to avoid discontinuity artifacts at the patch boundaries. In fact, unless the computational time is a concern, it is recommended that maximum overlap is used such that adjacent extracted patches are shifted by only one pixel in each direction. This means extracting all possible patches from the image.

The objective function in Equation (2.6) is easy to understand. The first term requires the denoised image to be close to the measurement,

x, and the second term requires that every patch extracted from this image to have a sparse representation in the dictionary D. The common approach to solving this optimization problem is an approximate block-coordinate minimization. First, we initialize z to the noisy measurement ($z = x$). Keeping z fixed, the objective function is minimized with respect to $\{\gamma_i\}_{i=1}^N$. This step is simplified because it is equivalent to N independent problems, one for each patch, that can be solved using sparse coding algorithms. Then $\{\gamma_i\}_{i=1}^N$ are kept fixed and the objective function is minimized with respect to z. This minimization has a closed-form solution:

$$\widehat{x_0} = \left(\lambda I + \sum_{i=1}^{N} R_i^T R_i\right)^{-1} \left(\lambda x + \sum_{i=1}^{N} R_i^T D \hat{\gamma}_i\right) \tag{2.7}$$

There is no need to form and invert a matrix to solve this equation. It is basically equivalent to returning the denoised patches to their right place on the image canvas and performing a weighted averaging. The weighted averaging simply takes into account the overlapping of the patches and a weighted averaging with the noisy image x (with weight λ).

The minimization with respect to $\{\gamma_i\}_{i=1}^N$ and z can be performed iteratively by using $\widehat{x_0}$ obtained from (2.7) as the new estimate of the image. However, this will run into difficulties because the noise distribution in $\widehat{x_0}$ is unknown and it is certainly not white Gaussian. Therefore, $\widehat{x_0}$ obtained from (2.7) is usually used as the estimate of the underlying image x_0.

Image inpainting

Let us denote the true underlying image with x_0 and assume that the observed image x not only contains noise, but also some pixels are not observed or are corrupted to the extent that the measurements of those pixels should be ignored. The model used for this scenario is $x = Mx_0 + w$ where w is the additive noise and M is a mask matrix, which is a binary matrix that removes the unobserved/corrupted pixels. The goal is to recover x_0 from x. Similar to the denoising problem above, one can use the prior assumptions that patches of x_0 have a sparse representation in a dictionary D. The MAP estimate of x_0 can be

found as a solution of the following problem [99]:

$$\left\{\widehat{x_0}, \{\widehat{\gamma}_i\}_{i=1}^N\right\} = \underset{z,\{\gamma_i\}_{i=1}^N}{\mathbf{argmin}} \lambda\|Mz - x\|_2^2 + \sum_{i=1}^N \left(\|R_i z - D\gamma_i\|_2^2 + \|\gamma_i\|_0\right) \tag{2.8}$$

An approximate solution can be found using an approach rather similar to that described above for the denoising problem. Specifically, we start with an initialization $z = M^T x$. Then, assuming that z is fixed, we solve N independent sparse coding problems to find estimates of $\{\gamma_i\}_{i=1}^N$. The only issue here is that this initial z will be corrupted at the locations of unobserved pixels. Therefore, the estimation of $\{\gamma_i\}_{i=1}^N$ needs to take this into account by introducing a local mask matrix for each patch:

$$\hat{\gamma}_i = \underset{\gamma}{\mathbf{argmin}} \|M_i(R_i z - D\gamma)\|_2^2 \quad \textbf{subject to:} \quad \|\gamma\|_0 \leq K \tag{2.9}$$

Once $\{\gamma_i\}_{i=1}^N$ are estimated, an approximation to the underlying full image is found as:

$$\hat{x_0} = \left(\lambda M^T M + \sum_{i=1}^N R_i^T R_i\right)^{-1} \left(\lambda M^T x + \sum_{i=1}^N R_i^T D\hat{\gamma}_i\right) \tag{2.10}$$

which has a simple interpretation similar to (2.7). The method described above has been shown to be very effective in many studies [215, 216, 254].

Image scale-up (super-resolution)

As we saw above, the applications of learned dictionaries for image denoising and inpainting can be quite straightforward. Nevertheless, application of learned dictionaries for image processing may involve much more elaborate approaches, even for the simple tasks such as denoising. As an example of a slightly more complex task, in this section we explain the image scale-up. Image scale-up can serve as a good example of more elaborate applications of learned dictionaries in

image processing. Moreover, it has been one of the most successful applications of learned dictionaries to date [345].

Suppose x_h is a high-resolution image. A blurred low-resolution version of this image can be modeled as $x_l = SHx_h$, where H and S denote the blur and down-sampling operators. Given the measured low-resolution image, which can also include additive noise (i.e., $x_l = SHx_h + w$), the goal is to recover the high-resolution image. This problem is usually called the image scale-up problem, and it is also referred to as image super-resolution.

The first image scale-up algorithm that used learned dictionaries was suggested in [345]. This algorithm is based on learning two dictionaries, one for sparse representation of the patches of the high-resolution image and one for sparse representation of the patches of the low-resolution image. Let us denote these dictionaries with D_h and D_l, respectively. The basic assumption in this algorithm is that sparse representation of a low-resolution patch in D_l is identical to the sparse representation of its corresponding high-resolution patch in D_h. Therefore, given a low-resolution image x_l, one can divide it into patches and use each low-resolution patch to estimate its corresponding high-resolution patch. Let us denote the ith patch extracted from x_l with X_i^l and its corresponding high-resolution patch with X_i^h. One first finds the sparse representation of X_i^l in D_l using any sparse coding algorithm such that $X_i^l \cong D\gamma_i$. Then, by assumption, γ_i is also the sparse representation of X_i^h in D_h. Therefore, the estimate of X_i^h will be: $\widehat{X_i^h} \cong D_h\gamma_i$. These estimated high-resolution patches are then placed on the canvas of the high-resolution image and the high-resolution image is formed via a weighted averaging similar to that in the denoising application above. The procedure that we explained here for estimating the high-resolution patches from their low-resolution counterparts is the simplest approach. In practice, this procedure is applied with slight modifications that significantly improve the results [99, 345, 352].

The main assumption in the above algorithm was that the sparse codes of the low-resolution and high-resolution patches were identical. This is an assumption that has to be enforced during dictionary learning. In other words, the dictionaries D_h and D_l are learned such that this condition is satisfied. The dictionary learning approach suggested in

[345] is:

$$\underset{D_h, D_l, \Gamma}{\textbf{minimize}} \frac{1}{m_h}\|X^h - D_h\Gamma\|_F^2 + \frac{1}{m_l}\|X^l - D_l\Gamma\|_F^2 + \lambda\|\Gamma\|_1 \tag{2.11}$$

where X^h and X^l represent the matrices of training signals. The ith column of X^h is the vectorized version of a patch extracted from a high-resolution image and the ith column of X^l is the vectorized version of the corresponding low-resolution patch. m_h and m_l are the lengths of the high-resolution and low-resolution training signals and are included in the objective function to properly balance the two terms. The important choice in the objective function in (2.11) is to use the same Γ in the first and the second terms of the objective function. It is easy to understand how this choice forces the learned dictionaries D_h and D_l to be such that the corresponding high-resolution and low-resolution patches have the same sparse representation.

The above algorithm achieved surprisingly good results [345]. However, it was soon realized that the assumption of this algorithm on the sparse representations was too restrictive and that better results could be obtained by relaxing those assumptions. For instance, one study suggested a linear relation between the sparse representations of low-resolution and high-resolution patches and obtained better results [327]. The dictionary learning formulation for this algorithm had the following form:

$$\begin{aligned}\underset{\{D_h, \Gamma^h, D_l, \Gamma^l, W\}}{\textbf{minimize}} \quad & \Big(||X^h - D_h\Gamma^h||_F^2 + ||X^l - D_l\Gamma^l||_F^2 \\ & + \lambda_h||\Gamma^h||_1 + \lambda_l||\Gamma^l||_1 \\ & + \lambda_W||\Gamma^l - W\Gamma^h||_F^2 + \alpha||W||_F^2 \Big)\end{aligned} \tag{2.12}$$

It is easy to see that here the assumption is not that the sparse representation of high-resolution patches (Γ^h) is the same as the sparse representation of the low-resolution patches (Γ^l), but that there is a linear relationship between them. This linear relation is represented by the matrix W. This results in a much more general and more flexible model. On the other hand, this is also a more difficult model to learn because it requires learning of the matrix W, in addition to the

two dictionaries. In [327], a block-coordinate optimization algorithm was suggested for solving this problem and it was shown to produce very good results.

There have also been other approaches to relaxing the relationship between the sparse codes of high-resolution and low-resolution patches. For instance, one study suggested a bilinear relation involving two matrices [132]. Another study suggested a statistical inference technique to predict the sparse code of the high-resolution patches from low-resolution ones [253]. Both of these approaches reported very good results. In general, image scale-up with the help of learned dictionaries has shown to outperform other competing methods and it is a good example of the power of learned dictionaries in modeling natural images.

Other applications

In the above, we explained three applications of learned dictionaries. However, these dictionaries have proved highly effective in many other applications as well. Some of these other applications include image demosaicking [215, 225], deblurring [69, 88], compressed sensing [55, 93, 267], morphological component analysis [99], compression [36, 298], classification [147, 265], cross-domain image synthesis [362] and removal of various types of artifacts from the image [151, 350].

For many image processing tasks, such as denoising and compression, application of the dictionary is relatively straightforward. However, there are also more complex tasks for which learning and application of overcomplete dictionaries are much more complex. It has been suggested in [210, 211] that many of these applications can be considered as instances of classification or regression problems. The authors of [210] coin the term "task-driven dictionary learning" to describe these applications and suggest that a general optimization formulation for these problems is of the form:

$$\underset{D\in\mathcal{D},W\in\mathcal{W}}{\textbf{minimize}}\ \mathcal{L}(Y,W,\hat{\Gamma})+\lambda\|W\|_F^2 \tag{2.13}$$

In the above formulation, $\hat{\Gamma}$ is the matrix of representation coefficients of the training signals, obtained by solving a problem such as (2.2). The cost function $\mathcal{L}$ quantifies the error in the prediction of the target variables Y from the sparse codes $\hat{\Gamma}$, and W denotes the model

parameters. For a classification problem, Y represents the labels of the training signals, whereas in a regression setting Y represents real-valued vectors. For example, the image scale-up problem that we have presented above is an example of the regression setting where Y represents the vectors of the pixel values of the high-resolution patches. The second term in the above objective function is a regularization term on model parameters that is meant to avoid overfitting and numerical instability.

Therefore, in task-driven dictionary learning, the goal is to learn the dictionary not only for sparse representation of the signal, but also so that it can be employed for accurate prediction of the target variables, Y. The general optimization problem in (2.13) is very difficult to solve. In addition to the fact that the objective function is non-convex, the dependence of $\mathcal{L}$ on D is through $\hat{\Gamma}$, which is in turn obtained by solving (2.2). In the asymptotic case when the amount of training data is very large, it has been shown that this general optimization problem is differentiable and can be effectively solved using stochastic gradient descent [210]. It has been shown that this approach can lead to very good results in a range of classification and regression tasks such as compressed sensing, handwritten digit classification, and inverse halftoning [210].

2.2 Non-local patch-based image processing

Natural images contain abundant self-similarities. In terms of image patches, this means that for every patch in a natural image we can probably find many similar patches in the same image. The main idea in non-local patch-based image processing is to exploit this self-similarity by finding/collecting similar patches and processing them jointly. The idea of exploiting patch similarities and the notion of nonlocal filtering are not very new [70, 97, 316, 330, 349]. However, it was the non-local means (NLM) denoising algorithm proposed in [37] that started the new wave of research in this field. Even though the basic idea behind NLM denoising is very simple and intuitive, it achieves remarkable denoising results and it has created a great deal of interest in the image processing community.

Let us denote the noisy image with $x = x_0 + w$, where, as before, x_0 denotes the true image. We also denote the ith pixel of x with $x(i)$ and a patch/block centered on $x(i)$ with $x[i]$. We will use similar notations in the rest of this dissertation. The NLM algorithm considers overlapping patches,

each patch centered on one pixel. The value of the ith pixel in the underlying image, $x_0(i)$, is estimated as a weighted average of the center pixels of all the patches as follows:

$$\widehat{x_0}(i) = \sum_{j=1}^{N} \frac{G_a(\|x[j] - x[i]\|_F^2)}{\sum_{j=1}^{N} G_a(\|x[j] - x[i]\|_F^2)} x(j) \tag{2.14}$$

where G_a denotes a Gaussian kernel with bandwidth a and N is the total number of patches. The intuition behind this algorithm is very simple: similar patches are likely to have similar pixels at their centers. Therefore, in order to estimate the true value of the ith pixel, the algorithm performs a weighted averaging of the values of all pixels, with the weight being related to the similarity of each patch with the patch centered on the ith pixel. Although in theory all patches can be included in the denoising of the ith pixel, as shown in (2.14), in practice only patches from a small neighborhood around this pixel are included. In fact, many of the methods that are based on NLM denoising first find several patches that are similar to $x[i]$. Only those patches that are similar enough to $x[i]$ are used in computing $\widehat{x_0}(i)$. Therefore, a practical implementation of the NLM denoising will be:

$$\begin{aligned} \widehat{x_0}(i) &= \sum_{j \in S_i} \frac{G_a(\|x[j] - x[i]\|_F^2)}{\sum_{j \in S_i} G_a(\|x[j] - x[i]\|_F^2)} x(j) \\ &\textbf{where:} \quad S_i = \{j |\ j \in \mathcal{N}_i \ \&\ \|x[j] - x[i]\|_F \leq \epsilon\} \end{aligned} \tag{2.15}$$

where $\mathcal{N}_i$ is a small neighborhood around the ith pixel and ϵ is a noise-dependent threshold.

The idea behind the NLM has proved to be an extremely powerful model for natural images. For the denoising task, NLM filtering and its extensions have led to the best denoising results [228, 289]. Some studies have shown that the current state-of-the-art algorithms are approaching the theoretical performance limits of denoising [51, 182]. Some of the recent extensions of the basic NLM denoising include Bayesian/probabilistic extensions of the method [178, 333], spatially adaptive selection of the algorithm parameters [91, 95], combining NLM denoising with TV denoising [306], and the use of non-square patches that has been shown to improve the results around edges and high-contrast features [83]. Some of the most productive extensions of the NLM scheme involve exploiting the power of learned dictionaries. We will discuss these methods in the next section.

Nonlocal patch-based methods are very computationally demanding. Therefore, a large number of research papers have focused on speedup strategies. A very effective strategy was proposed in [73]. This strategy is based on building a temporary image that holds the discrete integration of the squared differences of the noisy image for all patch translations. This integral image is used for fast computation of the patch differences $(x[j] - x[i])$, which is the main computational burden in nonlocal patch-based methods. A large number of papers have focused on reducing the computational cost of NLM denoising by classifying/clustering the image patches before starting the denoising process [27, 77, 207]. The justification behind this approach is that the computational bottleneck of NLM denoising is the search for similar patches. Therefore, these methods aim at clustering the patches so that the search for similar patches becomes less computationally demanding. Most of these methods compute a few features from each patch to obtain a concise representation of the patches. Typical features include average gray value and gradient orientation. During denoising, for each patch a set of similar patches is found using the clustered patches. One study compared various tree structures for fast finding of similar patches in an image and found that vantage point trees are superior to other tree structures [170]. Another class of highly efficient algorithms for finding similar patches are stochastic in nature. These methods can be much faster than the deterministic techniques we mentioned above, but they are less accurate. Perhaps the most widely used algorithm in this category is the PatchMatch algorithm and its extensions [12, 13].

The NLM algorithm and its extensions that we will explain in the next section have been recognized as the state-of-the-art methods for image denoising. However, the idea of exploiting the patch similarities has been used for many other image processing tasks. For instance, it has been shown that nonlocal patch similarities can be used to develop highly effective regularizations for inverse problems and iterative image reconstruction algorithms [114, 194, 229, 255, 354]. Below, we briefly explain two of these algorithms.

Let us consider the inverse problem of estimating an unknown image x from the measurements $y = Ax + w$, where w is the additive noise. The matrix A represents the known forward model that can be, for example, a blur matrix (in image deblurring) or the projection matrix (in tomography). In [255], it is suggested to recover x by solving the optimization problem:

$$\hat{x} = \underset{x}{\mathbf{argmin}} \, \|y - Ax\|_2^2 + \lambda \sum_i \sum_j \sqrt{w_{i,j}} |(x(i) - x(j)| \tag{2.16}$$

where $w_{i,j}$ are the nonlocal patch-based weights that are computed in a fashion similar to NLM denoising:

$$w_{i,j} = \frac{1}{Z} \exp\left(- \frac{\|x[i] - x[j]\|}{2\sigma^2} \right) \tag{2.17}$$

where Z is a normalizing factor. Therefore, the regularization term in (2.16) is a non-local total variation on a graph where the graph weights are based on nonlocal patch similarities. The difficulty with solving this optimization problem is that the weights themselves depend on the unknown image, x. The algorithm suggested in [255] iteratively estimates the weights from the latest image estimate and then updates the image based on the new weights using a proximal gradient method. In summary, given the image estimate at the kth iteration, $\hat{x}^k$, the weights are estimated from this image. Then, the image is updated using a proximal gradient iteration [46, 66] :

$$\begin{aligned} \hat{x}^{k+1} =& \mathbf{Prox}_{\mu\lambda J}\left(\hat{x}^k + \mu A^T \big(y - A\hat{x}^k\big) \right) \\ \mathbf{Prox}_J(x) =& \mathbf{argmin}_z \frac{1}{2}\|x - z\|_2^2 + J(z) \end{aligned} \tag{2.18}$$

where J is the regularization term in (2.16) and μ is the step size. Having computed the new estimate $\hat{x}^{k+1}$, the patch-based weights are re-computed and the algorithm continues. This algorithm showed very good results on three types of inverse problems including compressed sensing, inpainting, and image scale-up [254].

In [348], the following optimization problem was suggested for recovering the unknown image x.

$$\hat{x} = \mathbf{argmin}_x \|y - Ax\|_2^2 + \lambda \sum_i \sum_{j \in \mathcal{N}_i} \|x[i] - x[j]\|^p \tag{2.19}$$

where $p \leq 1$ and $\mathcal{N}_i$ is a neighborhood around the ith pixel. An iterative majorization-minimization algorithm is suggested for solving (2.19). Majorization of the regularization term will lead to the following quadratic surrogate problem:

$$\hat{x} = \mathbf{argmin}_x \|y - Ax\|_2^2 + \lambda\, x^T S x \tag{2.20}$$

where S is a sparse matrix representing the patch similarities. The algorithm alternates between minimization of (2.20) using a conjugate gradient descent method and updating the matrix S from the new image estimate.

As we mentioned above, nonlocal patch similarities have been shown to be very useful for many image processing tasks. Because of space limitations, in this section we focused on image denoising and inverse problems, which are more relevant to CT. However, we should mention that in recent years, the idea of exploiting nonlocal patch similarities has been applied to many image processing tasks and this is currently a very active area of research. Some examples of these applications include image enhancements [38], deblurring [163], inpainting [124], and super-resolution [262].

2.3 Other patch-based methods

The large number and diversity of patch-based image processing algorithms that have been developed in the past ten years makes it impossible to review them all here. Nonetheless, most of these algorithms are based on sparse representation of patches in learned dictionaries (Section 2.1) and/or exploiting nonlocal patch similarities (Section 2.2). In this section, we try to provide a broad overview of some of the extensions of these ideas and other patch-based methods.

To begin with, it is natural to combine the two ideas of learned dictionaries and non-local filtering to enjoy the benefits of both methods. Research in this direction has proven to be very fruitful. The first algorithm to explicitly follow this approach was "the non-local sparse model" proposed in [214]. This method collects similar patches of the image, as in NLM denoising. However, unlike NLM that performs a weighted averaging, the non-local sparse model uses sparse coding of similar patches in a learned dictionary. The basic assumption in the non-local sparse model is that similar patches should use similar dictionary atoms in their representations. Therefore, simultaneous sparse coding techniques (e.g., [318, 319]) are applied on groups of similar patches.

The idea of combining the benefits of non-local patch similarities and of learned dictionaries has been explored by many studies in the recent years [50, 84, 87, 88, 311, 347]. Most of these methods have reported state-of-the-art results. Although the details of these algorithms are different, the main ideas can be simply explained in terms of the non-local patch similarities and sparse representation in learned dictionaries. The K-LLD algorithm [50], for example, uses steering kernel regression method to find structurally similar patches and then uses PCA to learn a suitable dictionary for each set of similar patches. The Adaptive Sparse Domain Selection (ASDS) algorithm [88], on the other hand, clusters the training patches and learns a sub-

dictionary for each cluster using PCA. For a new patch, then, ASDS selects the most relevant sub-dictionary for sparse coding of that patch. The idea of using PCA for building the dictionaries in these methods has received great attention because the learned dictionaries will be orthonogonal. In [84], global, local, and hierarchical implementations of PCA dictionaries were studied. It was found the local-PCA (i.e., PCA applied on patches selected from a sliding window) led to the best results.

A very successful patch-based image denoising algorithm, that has similarities with the non-local sparse model, is the BM3D algorithm [71]. Even though BM3D has been proposed in 2007, it is still regarded as the state-of-the-art image denoising algorithm. Similar to the non-local sparse model, BM3D collects similar patches and filters them jointly. However, unlike the non-local sparse model, it uses orthogonal DCT dictionaries instead of learned overcomplete dictionaries. Moreover, BM3D works in two steps. First, patch-matching and filtering is performed on the original noisy image to obtain an intermediate denoised image. Then, a new round of denoising is performed. This time, the intermediate image is used for finding similar patches. The algorithm includes other components such as Wiener filtering and weighted averaging [71]. Further improvements to the original BM3D algorithm and an extension to 3D images (called the BM4D algorithm) have also been proposed [72, 205].

2.4 Patch-based methods for Poisson noise

In this section, we focus on the patch-based methods for the case when the noise follows a Poisson distribution. The reason for devoting a section to this topic is that, as we explained in Section 1.2, the noise in CT projection measurements has a complex distribution that can be best approximated as a Poisson noise or, after log-transformationm, as a Gaussian noise with signal-dependent variance [204, 326]. In any case, application of the patch-based image processing methods to the projection measurements in CT requires careful consideration of the complex noise distribution. Unfortunately, most of the patch-based image processing methods, including all algorithms that we have described so far in this chapter, have been proposed for Gaussian noise. Moreover, most of these algorithms (with the exception of fully-Bayesian methods described in Section 2.1.2) assume that the Gaussian noise has a uniform variance. Comparatively, the research on patch-based methods for the case of Poisson noise has been very limited and most of these limited works have been published very recently.

An important first obstacle facing the application of patch-based methods to the case of Poisson noise is the choice of an appropriate patch similarity measure. Methods that depend on nonlocal patch similarities need a patch similarity measure to find similar patches. Likewise, when we use sparse representation of the patches in a learned dictionary we often need a patch similarity measure. This is needed, for example, for finding the sparse representation of the patch in the dictionary using greedy methods. When the noise has a Gaussian distribution, the standard choice is the Euclidean distance, which has a sound theoretical justification and is easy to use.

For the non-Gaussian noise distributions, one straight-forward approach is to apply a so-called variance-stabilization transform so that the noise becomes close to Gaussian and then use the Euclidean distance. For the Poisson noise, the commonly-used transforms include the Anscombe transform [7] and the Haar-Fisz transform [111]. If one wants to avoid these transforms and work with the original patches that are contaminated with Poisson noise, the proper choice of patch similarity measure is less obvious. Over the years, many criteria have been suggested for measuring the similarity between patches contaminated with Poisson noise [6]. For the case of low-count Poisson measurements, one study has suggested that the earth mover's distance (EMD) is a good measure of distance between patches [115]. It has been suggested that EMD can be approximated by passing the patches with Poisson noise through a Gaussian filter and then applying the Euclidean distance [115]. One study compared several different patch distance measures for Poisson noise through extensive numerical experiments [85]. It was found that the generalized likelihood ratio (GLR) was the best similarity criterion in terms of the trade-off between the probability of detection and false alarm [85]. GLR has many desirable theoretical properties that make it very appealing as a patch distance measure [82]. For the Poisson noise, this ratio is given by the following Equation:

$$\mathcal{L}_G(x_1, x_2) = \frac{(x_1 + x_2)^{x_1+x_2}}{2^{x_1+x_2} x_1^{x_1} x_2^{x_2}} \tag{2.21}$$

Given two noisy patches $x_1[i]$ and $x_2[i]$, where $i \in \omega$, and assuming that the noise in pixels is independent, this gives the following similarity measure

between the two patches:

$$\begin{aligned}\mathcal{S}(x_1, x_2) = \sum_{i\in\omega} &(x_1[i] + x_2[i]) \log(x_1[i] + x_2[i]) \\ &- (x_1[i]) \log(x_1[i]) - (x_2[i]) \log(x_2[i]) \\ &- (x_1[i] + x_2[i]) \log 2\end{aligned} \tag{2.22}$$

In [82], the GLR-based patch similarity criterion was also compared with six other criteria for non-local patch-based denoising of images with Poisson noise. It was found that using GLR led to the best denoising result when the noise is strong [82]. When the noise was not strong, the results showed that it was better to use a variance-stabilization transform to convert the Poisson noise into Gaussian noise and then to use the Euclidean distance. The algorithm used in [82] for non-local filtering is as follows:

$$\widehat{x_0}(i) = \sum_{j=1}^{N} \frac{\mathcal{L}_G(x[j] - x[i])^{1/h}}{\sum_{j=1}^{N} \mathcal{L}_G(x[j] - x[i])^{1/h}} x(j) \tag{2.23}$$

This algorithm includes the parameter h instead of the kernel bandwidth a in Equation (2.14).

Another nonlocal patch-based denoising algorithm for Poisson noise was suggested in [81]. A main feature of this algorithm is that the patch similarity weights are computed from the original noisy image as well as from a pre-filtered image:

$$\widehat{x_0}(i) = \sum_{j=1}^{N} \frac{w_{i,j}}{\sum_{j=1}^{N} w_{i,j}} x(j) \quad \textbf{where:} \quad w_{i,j} = \exp\left(-\frac{u_{i,j}}{\alpha} - \frac{v_{i,j}}{\beta} \right) \tag{2.24}$$

where $u_{i,j}$ are computed from the noisy image using a likelihood ratio principle and $v_{i,j}$ are computed from a pre-estimate of the true image using the symmetric Kullback-Leibler divergence. It is shown that the optimal values for the parameters α and β can be computed and that this algorithm can achieve state of the art denoising results.

The patch-similarity measure in (2.22) was used to develop a k-medoids denoising algorithm in [44]. The k-medoids algorithm is similar to k-means algorithms. They are different in that k-means uses the centroid of each cluster as the representative of that cluster, whereas the k-medoids algorithm uses data points (i.e., examples) as the representative of the cluster. Moreover, k-medoids can work with any distance measure, not necessarily

the Euclidean distance. It was shown in [44] that the k-medoids algorithm achieved very good Poisson denoising results, outperforming the nonlocal Poisson denoising method of [82] in some tests. The k-medoids algorithm is in fact a special case of the dictionary learning approach. The difference with the dictionary-learning approach is that in the k-medoids algorithm only one atom participates in the representation of each patch.

The reason why the study in [44] limited itself to using only one atom for representation of each patch was the difficulties in sparse coding under the Poisson noise. Suppose that $x_0[i]$ is the ith patch of the true underlying image and $x[i]$ is the measured patch under Poisson noise. If we wish to recover $x_0[i]$ from $x[i]$ via sparse representation in a dictionary D, we need to solve a problem that has the following form [94]:

$$\widehat{\gamma_i} = \underset{\gamma \text{ s.t. } \|\gamma\|_0 \leq T}{\textbf{argmin}} \; \mathbf{1}^T D\gamma_i - x[i]^T \log(D\gamma_i) \quad \textbf{subject to:} \quad D\gamma_i > 0 \tag{2.25}$$

Having found $\widehat{\gamma_i}$, we will have: $\widehat{x_0}[i] = D\widehat{\gamma_i}$. The difficulties of solving this problem have been discussed in [94, 115] and greedy sparse coding algorithms have been proposed for solving this problem. The author of [94] then apply their proposed algorithm for denoising of images with Poisson noise. Even though they use a wavelet basis for D, they achieve impressive results.

A true dictionary learning-based denoising algorithm for images with Poisson noise was suggested in [115]. In that study, a global dictionary is learned from a set of training data. Then, for a given noisy image to be denoised, the algorithm first clusters similar patches. All patches in a cluster are denoised together via simultaneous sparse representation in D. This means that patches that are clustered together are forced to share similar dictionary atoms in their representation. Experiments showed that this method was comparable with or better than competing methods. A slightly similar approach that also combines the ideas of learned dictionaries and non-local filtering is proposed in [280, 281]. In this approach, k-means clustering is used to group similar image patches. A dictionary is learned for each cluster of similar patches using the Poisson-PCA algorithm [65, 297]. For solving the Poisson-PCA problem, which is also known as exponential-PCA, the authors use the Newton's method. This algorithm showed good performance under low-count Poisson noise.

2.5 Total variation (TV)

Total variation (TV), which was first proposed in [277] for image denoising and reconstruction, has become one of the most widely used regularization functions in image processing. For a function $x(t)$ defined on the interval $[0, 1]$, it is defined as [278]:

$$\mathrm{TV}(x) = \sup \sum_i |x(t_i) - x(t_{i-1})| \tag{2.26}$$

where the supremum is computed over all possible partitions of the interval $[0, 1]$. For a piecewise-constant signal, $\mathrm{TV}(x)$ is simply the sum of the magnitudes of the signal jumps. If $x(t)$ is smooth, the following equivalent definition exists:

$$\mathrm{TV}(x) = \int_0^1 \left| \frac{dx}{dt} \right| dt \tag{2.27}$$

For a function $x(s, t)$ of two variables defined on the unit square, the above definition can be extended as:

$$\mathrm{TV}(x) = \int_0^1 \int_0^1 \left\| \left(\frac{\partial x}{\partial s}, \frac{\partial x}{\partial t} \right) \right\| ds\, dt \tag{2.28}$$

Different discretizations have been proposed. Suppose $x \in \mathbb{R}^{N \times N}$ is a 2D image. A common discretization is [46]:

$$\mathrm{TV}(x) = \sum_{1 \le i,j \le N} |(\nabla x)_{i,j}| \tag{2.29}$$

where

$$\begin{aligned} (\nabla x)_{i,j} =& \big((\nabla x)^1_{i,j}, (\nabla x)^2_{i,j}\big) \\ (\nabla x)^1_{i,j} =& \begin{cases} x_{i+1,j} - x_{i,j} & \text{if } i < N \\ 0 & \text{if } i = N \end{cases} \\ (\nabla x)^2_{i,j} =& \begin{cases} x_{i,j+1} - x_{i,j} & \text{if } j < N \\ 0 & \text{if } j = N \end{cases} \end{aligned} \tag{2.30}$$

and for $z = (z_1, z_2) \in \mathbb{R}^2$, $|z| = \sqrt{z_1^2 + z_2^2}$.

Suppose that we obtain measurements $y = Ax + w$, where, as before, A is some operation or transformation such as blurring, sampling, or forward projection in CT and w is additive Gaussian noise with uniform variance.

The maximum a posteriori estimate of x with a total variation prior $P(x) \sim e^{-J(x)}$ is obtained as:

$$x_{\text{MAP}} = \underset{x}{\operatorname{argmin}} \|Ax - y\|_2^2 + \text{TV}(x) \tag{2.31}$$

A special case of this problem is the denoising problem shown below, which corresponds to the case where A is the identity matrix.

$$x_{\text{MAP}} = \underset{x}{\operatorname{argmin}} \|x - y\|_2^2 + \int_\Omega |\nabla x| du \tag{2.32}$$

which is usually referred to as the Rudin-Osher-Fatemi (ROF) model for image denoising.

The main properties of TV include convexity, lower semi-continuity, and homogeneity [45]. Many different algorithms have been suggested for solving this problem. Examples of the optimization approaches that are used to solve this problem include primal-dual methods [168, 278], second-order cone programming [116], dual formulations [46, 361], split Bregman methods [112, 117], and accelerated proximal gradient methods [17, 237].

In general, TV is a good model for recovering blocky images, i.e., images that consist of piecewise-constant features with sharp edges [278]. Many studies have used TV to successfully accomplish various image processing tasks, including denoising [46], deblurring [332], inpainting [249], restoration [29], and reconstruction [303]. However, on images with fine texture and ramp-like features, this model usually performs poorly [113]. Therefore, many studies have tried to improve or modify this model so that it can be useful for more complicated images. Some of the research directions include employing higher-order differentials [21, 23, 34], locally adaptive formulations that try to identify the type of local image features and adjust the action of the algorithm accordingly [53, 89, 120, 125], and combining TV with other regularizations in order to improve its performance [121, 199].

2.6 Published research on sparsity-based methods in CT

This section reviews some of the published research on the application of the sparsity-based models and algorithms described so far in this chapter in CT. We divide these applications into three categories: 1) pre-processing methods, which aim at restoring or denoising the projection measurements, 2) iterative reconstruction methods, and 3) post-processing methods, whose

goal is to enhance, restore, denoise, or otherwise improve the quality of the reconstructed image.

2.6.1 Pre-processing methods

Compared with iterative reconstruction methods and post-processing methods, pre-processing methods account for a much smaller share of the published studies on sparsity-based algorithms for CT. There are two main reasons behind this. The first reason is that the pre-processing methods for CT, in general, face certain difficulties. For example, it is well-known that sharp image features are smoothed in the projection domain. Therefore, the preservation of sharp image features and fine details is more challenging when working in the projection domain. Moreover, many commercial scanners do not allow access to the raw projection data. Therefore, it is more difficult to validate the pre-processing algorithms and apply them in clinical settings. The second reason is that a great majority of the sparsity-based image processing algorithms have been proposed with the assumption of additive Gaussian noise with uniform variance. As we described in Section 2.4, research on patch-based methods for the case of Poisson noise has been much more limited in extent and the algorithms that have been proposed for Poisson noise are very recent and have not yet been absorbed by researchers working on CT.

A patch-based sinogram denoising algorithm was proposed in [292]. A fixed DCT dictionary was used for representation of the sinogram patches. However, the shrinkage rule used for denoising was learned from the training data. The denoised projections were then used to reconstruct the image using an FBP method. A patch-based processing using learned shrinkage functions was then applied on the reconstructed image. The results of the study showed that this rather simple algorithm outperformed some of the well-known iterative CT reconstruction algorithms.

The use of learned dictionaries for inpainting (i.e., upsampling) of the CT projection measurements has also been proposed [188]. The goal of sinogram upsampling is to reduce the x-ray dose used for imaging by acquiring only a fraction of the projections directly and estimating the unobserved projections with upsampling. The assumption used in this algorithm was that patches extracted from the projections admit a sparse representation in a dictionary that could be learned from a set of training sinograms. The approach followed by this study was very similar to the general inpainting approach that we explained in Section 2.1.3. The results of the study showed that dictionary-based upsampling of the projections substantially

improved the quality of the images reconstructed with FBP, outperforming more traditional sinogram interpolation methods based on splines.

As we mentioned above, a challenge for all sinogram denoising/restoration algorithms is preservation of fine image detail. The algorithm presented in [291] has proposed an interesting idea to address this issue. In fact, this study contains several interesting ideas. One of these ideas is that in learning a dictionary for sparse representation of sinogram patches, not only the sinogram-domain error but also the error in the image domain is considered. Specifically, first a dictionary (D_1) is learned considering only the error in the sinogram domain. Let us denote the CT image by x and its sinogram by y. Then D_1 is found by solving:

$$\{D_1, \hat{\Gamma}\} = \underset{D,\Gamma}{\mathbf{argmin}} \|\Gamma\|_0 \quad \textbf{subject to:} \quad \|D\Gamma_i - R_i y\|_2^2 \leq C\sigma_i \quad \forall\, i \tag{2.33}$$

This optimization to find D_1 is carried out using the K-SVD algorithm described in Section 2.1. The only difference here is that the signal-dependent nature of noise, σ_i, should be taken into account in the sparse coding step (C is a tuning parameter). This dictionary is then further optimized by minimizing the reconstruction error in the image domain:

$$D_2 = \underset{D}{\mathbf{argmin}} \left\| \mathcal{FBP}\Big(\sum_i (R_i^T R_i)^{-1} \sum_i R_i^T D\hat{\Gamma} \Big) - x \right\|_{Q,2}^2 \tag{2.34}$$

where we have used $\mathcal{FBP}$ to denote the CT reconstruction algorithm (here, filtered back-projection). Note that the $\hat{\Gamma}$ in the optimization problem (2.34) is that found by solving (2.33). In other words, for finding D_2 we keep the sparse representations fixed and find a dictionary that leads to a better reconstruction of the image, x. The notation $\|.\|_{Q,2}$ denotes a weighted ℓ_2 norm. It is suggested that the weights Q are chosen such that more weights are given to low-contrast features [291].

The x and y in the above equations denote the "training data", which includes a set of high-quality images and their projections. In fact, instead of only one image, a large number of images can be used for better training. Now, suppose that we are given noisy projections of a new object/patient, which we denote with y_{noisy}. It is suggested to denoise y_{noisy} in two steps. First, sparse representations of patches of y_{noisy} in D_1 are obtained. Denoting this with $\hat{\Gamma}$, the final denoised sinogram is obtained as the solution of

the following problem which uses D_2:

$$y_{\text{denoised}} = \underset{y}{\textbf{argmin}}\, \lambda \|y - y_{\text{noisy}}\|_W^2 + \sum_i \|D_2 \hat{\Gamma}_i - R_i y\| \tag{2.35}$$

where W are weights to account for the signal-dependent nature of the noise. This problem has a simple solution similar to Equation (2.7). Experimental results for 2D CT have shown promising results [291].

2.6.2 Iterative reconstruction methods

In recent years, several iterative image reconstruction algorithms involving regularizations in terms of image patches have been proposed for CT. In general, these algorithms have reported very promising results. However, a convincing comparison of these algorithms with other classes of iterative reconstruction algorithms such as those based on TV or other edge-preserving regularizations is still lacking. In this section, we review some of the iterative CT reconstruction algorithms that use patch-based or TV regularization.

A typical example of dictionary-based CT reconstruction algorithms is the algorithm proposed in [339]. That paper suggested recovering the image as a solution of the following optimization problem:

$$\underset{x,D,\Gamma}{\textbf{minimize}} \sum_i w_i([Ax]_i - y_i))^2 + \lambda \left(\sum_k ||R_k x - D\Gamma_k||_2^2 + \nu_k ||\Gamma_k||_0 \right) \tag{2.36}$$

In the above problem, A is the projection matrix and w_is are noise-dependent weights. The first term in the objective function encourages measurement consistency. The remaining terms constitute the regularization, which are very similar to the terms in the formulation of the basic dictionary learning problem in (2.1). In (2.36), the dictionary is learned from the image itself. The authors of [339] solved this problem by alternating minimization with respect to the three variables. Minimization with respect to x is carried out using the separable paraboloid surrogate method suggested in [101]. The problem with this approach, however, is that it requires access to the individual elements of the projection matrix. Although this is a simple requirement for 2D CT, this can be a major problem for large 3D CT because with efficient implementations of forward and back-projection operations it is not convenient to access individual matrix elements [152, 193]. Minimization with respect to D and Γ is performed using

the K-SVD and OMP algorithms, respectively. Alternatively, the dictionary can be learned in advance from a set of training images. This will remove D from the list of the optimization variables in (2.36), substantially simplifying the problem. Both approaches are presented in [339]. Experiments showed that both these approaches led to very good reconstructions, outperforming a TV-based algorithm.

Other formulations that are very similar to the one described above have been shown to be superior to TV-based reconstruction and other standard iterative reconstruction algorithms in electron tomography [4, 192]. Another study had first learned a dictionary from training images, but for the image reconstruction step it did not include the sparsity term in the objective function [105]. In other words, only the first two terms in the objective function in Equation (2.36) were considered. A gradient descent approach was used to solve the problem. That study obtained superior reconstructions with learned dictionaries compared to using a DCT basis.

One study used an optimization approach similar to the one described above, but used box-splines for image representation [279]. In other words, instead of native pixel representation of the image, box spline were used as the basis functions in the image domain. The unknown image x has a representation of the form $x = \sum_i c_i \phi_i$, where ϕ_i is the box spline centered on the ith pixel and c_i is the value of attenuation coefficient for that pixel. The resulting optimization problem is of the following form:

$$\underset{c,\Gamma}{\textbf{minimize}}\, \|Hc - y\|_W^2 + \lambda \left(\sum_k ||R_k c - D\Gamma_k||_2^2 + \nu_k ||\Gamma_k||_0 \right) \tag{2.37}$$

In the above problem, H is the forward model relating the image representation coefficients to the sinogram measurements, y. In other words, H is simply the equivalent of the projection matrix A. The rest of the objective function is the same as that in Equation (2.36). Once the representation coefficients, c, are found by solving (2.37), the image is reconstructed simply as $x = \sum_i c_i \phi_i$. The results of this study showed that this dictionary-based algorithm achieved much better reconstructions than a wavelet-based reconstruction algorithm.

The dual-dictionary methods proposed in [197, 355] rely on two dictionaries. One of the dictionaries (D_l) is composed of patches from CT images reconstructed from a small number of projection views, while the second dictionary (D_h) contains the corresponding patches from a high-quality image. The atoms of the two dictionaries have one-to-one correspondence. The

strategy here is to first find the sparse code of the patches of the image to be reconstructed in D_l and then to recover a good estimate of the patch by multiplying this sparse code with D_h. The dictionaries are not learned here, but they are built by sampling a large number of patches from few-view and high-quality training images. This approach has been reported to achieve better results than TV-based reconstruction algorithms [197].

A different dictionary-based reconstruction algorithm was suggested in [301]. In this algorithm, first a dictionary (D) is learned by solving a problem of the following form:

$$\underset{D,\Gamma}{\textbf{minimize}}\, \|X - D\Gamma\|_F^2 + \lambda\|\Gamma\|_1 \quad \textbf{subject to:} \quad D \in \mathcal{D} \;\;\&\;\; \Gamma \in \mathbb{R}_+ \tag{2.38}$$

where $\mathcal{D}$ can be an ℓ_2 ball and $\mathbb{R}_+$ is the non-negative orthant of the proper size. The above problem is solved using the Alternating Direction Method of Multipliers (ADMM) to find the dictionary. It is reported that learning the dictionary with ADMM is computationally very efficient and largely independent of the initialization. The learned dictionary is then used to regularize the reconstruction algorithm by requiring that the patches of the reconstructed image have a sparse representation in the dictionary. However, unlike most other dictionary-based algorithms, overlapping patches are not used. Instead, a novel regularization term is introduced to avoid the blocking artifacts at the patch borders. Specifically, the optimization problem to recover the image from projection measurements y has this form:

$$\underset{x_\Gamma}{\textbf{minimize}}\, \|Ax_\Gamma - y\|_2^2 + \lambda\|\Gamma\|_1 + \mu\|Lx_\Gamma\|_2^2 \tag{2.39}$$

where, to simplify the notation, we have used x_Γ to emphasize that the reconstructed image depends on the sparse representation matrix, Γ. The matrix L is a matrix that computes the directional derivatives across the patch boundaries. Therefore, the role of the last term in the objective function is to penalize large jumps at the patch boundaries, thereby suppressing blocking artifacts that arise when non-overlapping patches are used. Comparison with TV-based reconstruction showed that this dictionary-based reconstruction algorithm resulted in much better images, preserving fine textural detail that are smeared by TV-based reconstruction. Overall, the algorithm proposed in that paper contains several interesting ideas that can be useful for designing dictionary-based reconstruction algorithms for CT. A later paper studied the sensitivity of this algorithm to such factors as the scale and rotation of features in the training data [300].

An iterative reconstruction algorithm that combines sparse representation of image patches with sinogram smoothing was proposed in [305]. The image is reconstructed as a solution of the following optimization problem:

$$\begin{aligned}\underset{x,y,\Gamma}{\textbf{minimize}} &\|y-\bar{y}\| + \alpha\, y^T W y + \beta\|Ax-y\|_2^2 \\ &+ \lambda\left(\sum_k ||R_k x - D\Gamma_k||_2^2 + \nu_k||\Gamma_k||_0\right)\end{aligned} \tag{2.40}$$

The first two terms, where $\bar{y}$ is the measured noisy sinogram, represent the sinogram Markov random field model [189, 324]. The remaining terms are similar to those we encountered above. As usual, the authors have suggested to solve the above problem using a block-coordinate minimization, where the minimization with respect to the image x is carried out using a conjugate gradients method. That study also suggests interesting variations of the objective function in (2.40), but the experimental evaluations that are presented are very limited.

As the last example of dictionary-based iterative reconstruction algorithms, we include the method based on sparsifying transforms proposed in [256, 257]. Sparsifying transforms are variations of the analysis model for sparsity [268, 331]. In the analysis model, instead of the relation $x = D\gamma$ that we have discussed so far in this chapter, the relation $Dx = \gamma$ is used. In other words, D here acts as an operator on the signal (e.g., the image patch) to find the representation coefficients, γ. In [256, 257], it is suggested that the unknown CT image be recovered as the solution of the following optimization problem:

$$\begin{aligned}&\underset{x,D,\Gamma}{\textbf{minimize}} \sum_i \|DR_i x - \Gamma_i\|_2^2 + \lambda\|\Gamma\|_1 + \alpha H(D) \\ &\textbf{subject to:} \quad \|Ax-y\|_W^2 \leq \epsilon\end{aligned} \tag{2.41}$$

where $H(D)$ is a regularization on the dictionary D, and W represents the weights introduced to account for the signal-dependent noise variance. The results of that study showed that this approach led to results that were comparable with iterative reconstruction with synthesis formulation and TV-based regularization, while also being slightly faster.

In recent years, there has also been a growing attention to the potential of regularization in terms of non-local patch priors for iterative CT reconstruction. In [194], it was suggested to recover the CT image as a solution

of the following optimization problem:

$$\hat{x} = \underset{x}{\mathbf{argmin}} \, \|y - Ax\|_2^2 + \lambda J_{\mathrm{NL}}(x) \tag{2.42}$$

where $J_{\mathrm{NL}}(x)$ is the regularization in terms of patch similarities. Two different forms were suggested for $J_{\mathrm{NL}}(x)$:

$$\begin{aligned} J_{\mathrm{NL/TV}}(x) &= \sum_i \sum_{j \in \mathcal{N}_i} \|\sqrt{w_{i,j}}(x(i) - x(j))\|_2 \\ J_{\mathrm{NL}/H^1}(x) &= \sum_i \sum_{j \in \mathcal{N}_i} w_{i,j} \|x(i) - x(j)\|_2^2 \end{aligned} \tag{2.43}$$

where $w_{i,j}$ are the patch-based similarity weights. For the ith pixel, they are computed from all pixels j in a window around i using:

$$w_{i,j} = \exp\left(- \frac{\|x[i] - x[j]\|}{h^2} \right) \tag{2.44}$$

It is suggested that these weights be computed from a FBP-reconstructed image and that the filter parameter h be chosen based on the local estimate of noise variance. The local noise variance is estimated from the wavelet coefficients of the finest wavelet subband (v_i) according to [90]:

$$h = \frac{\mathrm{median}(|v_i|)}{0.6745} \tag{2.45}$$

The authors of [194] solved the problem (2.42) with either of the regularization functions in (2.43) using a simple gradient descent and found that the recovered CT image had a better visual and objective quality than a standard TV-based iterative reconstruction algorithm.

As simple iterative algorithm that alternates between projections onto convex sets (POCS) to improve measurements consistency and an NLM-type restoration has been proposed in [133] . That algorithm was shown to perform better than a TV-based algorithm but no comparison with the state of the art methods was performed. Another study developed a NLM-type regularization for perfusion CT that relies on a high-quality prior image [202]. The proposed regularization function, shown in the following equation, is in terms of the similarity between patches of the unknown image to be reconstructed from a low-dose scan (x) and the patches of the prior image (x_p).

$$J(x) = \|x - \bar{x}\|_q^q \quad \textbf{where} \quad \bar{x}(i) = \sum_{j \in \mathcal{N}_i} w_{i,j} x_p(j) \tag{2.46}$$

The authors suggest $q = 1.2$. A steepest-descent approach is used to approximately solve this problem. A similar, but more general, algorithm that does not require a prior image was proposed in [353]. The formulation is the same as the above, the main difference being that the weights in the NLM formulation are computed from the image itself. A Gauss-Seidel approach is used to solve the resulting problem. Both of the above NLM-type regularization methods are reported to result in better reconstructions than more conventional regularizations such as Gaussian Markov random field.

Non-local patch-based regularization was also used for the new technique of equally-sloped tomography (EST, [227]) and was shown to improve the quality of the reconstructed image both from small or large number of projections [106]. Nonlocal patch-based regularization substantially improved the CNR, SNR, and spatial resolution of the images reconstructed from 60, 90, and 360 projections in that study.

Patch-based iterative reconstruction algorithms have also been proposed for dynamic CT. In dynamic CT, several successive images of the the same patient are reconstructed. Therefore, there is abundant temporal correlation (i.e., correlation between successive images) in addition to the spatial correlation within each of the images in the sequence. There have been several studies in recent years aimed at exploiting these correlations in terms of patch/block similarities. In general, these studies have reported very promising results.

A reconstruction algorithm with nonlocal patch-based regularization was proposed for dynamic CT in [155]. The proposed regularizer for the kth frame of the image is as follows:

$$\begin{aligned} J(x_k) = &\sum_i \sum_{j \in \mathcal{N}_i} G_a(x_k[i] - x_k[j]) |x_k(i) - x_k(j)|^2 \\ &+ \sum_i \sum_{l \in \{1,2,\ldots,K\} \setminus k} \sum_{j \in \Delta_i} G_a(x_l[i] - x_l[j]) |x_l(i) - x_l(j)|^2 \end{aligned} \tag{2.47}$$

where, as before, $x(i)$ and $x[i]$ denote the ith image pixel and the patch centered on that pixel, respectively. $G_a(.)$ is a Gaussian kernel as in the standard NLM denoising. The first term is a spatial regularization in terms of the patches of the current image frame, x_k. In this term, $\mathcal{N}_i$ is a simple rectangular neighborhood around the ith pixel. The second term (where K is the total number of frames) is a temporal patch-based regularization that involves patches from all other frames in the image sequence. In this term, Δ_i is a neighborhood whose spatial size is pre-fixed but whose temporal extension is found for each pixel such that the probability of finding patches

with similar structural features (e.g., edges) is increased. This is done by dividing the temporal neighborhood into blocks and estimating the structural similarity of these blocks with the patch centered on the ith pixel. Only a small fraction of blocks that are most similar to $x[i]$ are included in Δ_i. A similar approach was proposed in [154] for the case when a high-quality prior image is available. This high-quality prior image does not have to be a CT image and can be acquired in other imaging modalities. The results of experiments with simulated and real data show that this algorithm achieves very good reconstructions.

Temporal non-local-means (TNLM) algorithms were proposed in [146, 314]. These algorithms suggest recovering a set of successive CT images $\{x_k|\ k \in 1 : K\}$ by minimizing an optimization problem that includes (in addition to the measurement fidelity term) the following regularization:

$$J\big(\{x_k\}\big) = \sum_{k=1}^{K} \sum_{i} \sum_{j} w_{i,j} \big(x_k(i) - x_{k+1}(j)\big)^2 \tag{2.48}$$

where, as usual, the weights are computed based on patch similarities:

$$w_{i,j} = \frac{1}{Z} \exp\left(- \frac{\|x[i] - x[j]\|^2}{2h^2} \right) \tag{2.49}$$

An important choice in this algorithm is that only inter-image patch similarities are taken into account and not the intra-image patch similarities. The justification is that the proposed algorithm is for the case when each of the images in the sequence is reconstructed from a small number of projections and, hence, contains much streak artifacts. Therefore, using patches from the same image will amplify the streak artifacts, while using patches from neighboring images will suppress the artifacts. In addition to the iterative reconstruction algorithm, in [146] another very similar algorithm has been suggested that can also be classified as a post-processing algorithm. In this alternative scheme, each of the images in the sequence are first reconstructed from their correponding projections, and then they are post-processed using an optimization algorithm that includes the very same regularization function in (2.48).

A tensor-based iterative reconstruction algorithm was proposed for dynamic CT in [308]. Tensor-based dictionaries are a relatively new type of dictionary that are gaining more popularity. As we have mentioned above, in image processing applications, image patches/blocks are vectorized and used as training/test signals. Tensor-based methods treat the image patches or

blocks in their original form, i.e., without vectorizing them [40, 62]. Therefore, they are expected to better exploit the correlation between adjacent pixels. In [308], a tensor-based algorithm was compared with a standard dictionary for dynamic CT reconstruction and it was found that the tensor-based dictionaries resulted in a slight improvement in the quality of the reconstructed image.

Compared with the reconstruction algorithms that are based on learned dictionaries or nonlocal patch similarities, many more algorithms have used TV regularization terms. This is partly because TV-regularized cost functions are easier to handle using standard optimization algorithms, especially for large-scale 3D image reconstruction. Moreover, the CT community is more familiar with TV-based regularization because it has been used for CT reconstruction for a longer time. Many studies have formulated the reconstruction problem as a regularized least-squares minimization similar to (2.31). Some of the optimization techniques that have been suggested for solving this problem include accelerated first-order methods [61, 143, 251], alternating direction method of multipliers [264], and forward-backward splitting algorithm [145]. Another very commonly used formulation for CT reconstruction is the constrained optimization formulation, where the image TV is minimized under measurement consistency constraints [126, 242, 270]. Most published studies use an alternating algorithm for solving this problem, whereby at each iteration first the image TV is reduced, and this is then followed by a step that enforces the measurement consistency constraint. A simple (and probably inefficient [45]) method that has been adopted in many studies uses a steepest descent for TV minimization followed by projection onto convex sets for measurement consistency [293].

Several studies have combined the TV regularization with regularization in terms of a prior high-quality image in applications such as dynamic CT [22, 123], perfusion imaging [239], and respiratory-gated CT [180, 302]. In general, the existence of a high-quality prior image reduces the number of projection measurements required for reconstructing high-quality images from subsequent scans. Other variations of the standard TV regularization that have been successfully applied for CT reconstruction include nonconvex TV [49, 294] and higher-order TV [346].

In general, TV-based reconstruction methods have proven to be much better than traditional CT reconstruction algorithms, particularly in reconstruction from few-view and noisy projection data. Therefore, many studies have concluded that TV-based reconstruction methods have a great potential for dose reduction in a wide range of CT applications [28, 162, 309, 342]. However, there has been no satisfying comparison between TV and other

edge-preserving or smoothness-promoting regularization functions that are very widely used in CT [32, 80, 160, 325].

2.6.3 Post-processing methods

Many of the sparsity-based algorithms that have been proposed for CT fall into the category of post-processing methods. This is partly because most of these algorithms are directly based on the sparsity-based methods that have been proposed for natural images. Because general sparsity-based image processing algorithms mostly include denoising and restoration algorithms, they are more easily extended as post-processing methods for CT. Moreover, some of the sparsity-based methods, particularly patch-based image processing methods, are very computationally expensive. Therefore, especially for large-scale 3D CT, it is easier to deploy them as one-shot post-processing algorithms than as part of an iterative reconstruction algorithm.

A large number of dictionary-based algorithms have been proposed for CT denoising. The basic denoising algorithm that we described in Section 2.1.3 was used for denoising of abdomen CT images in [58, 60], and head CT images in [59] and showed promising results in all these studies. Straightforward representation of image patches in a learned dictionary followed by weighted averaging resulted in effective suppression of noise and artifacts and a marked improvement in the visual and objective image quality.

Non-local means methods have also been applied for CT image denoising. An early example is [156]. In that study, the authors investigated the effect of different parameters such as the patch size, smoothing strength, and the size of the search window around the current pixel to find similar patches. Among the findings of that study with lung and abdomen CT images was that one can choose the size of the search window for finding similar patches to be as small as 25×25 pixels and still achieve very impressive denoising results. However, this required careful tuning of the denoising parameter (a in Equation (2.15)). Moreover, choosing a small search window also required reducing the patch size to ensure that for every pixel a sufficient number of similar patches is found in the search window. Otherwise, in certain image areas such as around the edges, very little denoising is accomplished. Another study found that with a basic NLM denoising, the x-ray tube current setting can be reduced to one fifth of that in routine abdominal CT imaging without jeopardizing the image quality [56].

An algorithm specially tailored to image-guided radiotherapy was proposed in [343]. Since in this scenario a patient is scanned multiple times, it was suggested that the first scan be performed with standard dose and later

scans with much reduced dose. An NLM-type algorithm was suggested to reduce the noise in the low-dose images. The proposed algorithm denoised the low-dose images by finding similar patches in the image reconstructed from the standard-dose scan. Similarly, in CT perfusion imaging and angiography the same patient is scanned multiple times. A modified NLM algorithm was suggested for these imaging scenarios in [200]. The algorithm proposed in that study registered a standard-dose prior image to the low-dose image at hand. The low-dose image is then denoised using a NLM algorithm where patches are extracted from the registered standard-dose image.

One study suggested adapting the strength of the NLM denoising based on the estimated local noise level [191]. That paper proposed a fast method for approximating the noise level in the reconstructed image and suggested choosing the bandwidth of the Gaussian kernel in the NLM denoising to be proportional to the estimated standard deviation of the noise. Evaluations showed that this algorithm effectively suppressed the noise without degrading the spatial resolution. Using speed-up techniques such as those in [73], this algorithm was able to process large 3D images in a few minutes when implemented on a GPU.

Applying the nonlocal patch-based denoising methods in a spatially adaptive fashion has been proposed by many studies on natural images [157, 158]. For CT images, it is well known that the noise variance in the reconstructed image can vary significantly across the image. Therefore, estimating the local noise variance may improve the performance of the patch-based denoising methods. Another approach for estimating the local noise variance in the CT image was proposed in [15]. In this approach, which is much simpler than the method proposed in [191], even and odd-numbered projections are used to reconstruct two images. Then, assuming the noise in the projections are uncorrelated, the local noise variance is estimated from the difference of the two images.

So far in this section, we have talked about algorithms that have been suggested primarily for removing the noise. However, CT images can also be marred by various types of artifacts that can significantly reduce their diagnostic value [14]. Recently, a few patch-based algorithms have been proposed specifically for suppressing these artifacts. A dictionary-based algorithm for suppressing streak artifacts in CT images is proposed in [57]. The artifact-full image is first decomposed into its high-frequency bands in the horizontal, vertical, and diagonal directions. Sparse representation of patches of each of these bands are computed in three "discriminative" dictionaries that include atoms specifically learned to represent artifacts and genuine image features. Artifacts are suppressed by simply setting to zero

the large coefficients that correspond to the artifact atoms. The results of this study on artifact-full CT images are impressive.

A nonlocal patch-based artifact reduction method was suggested in [341]. This method is tailored for suppressing the streak artifacts that arise when the number of projections used for image reconstruction is small and it relies on the existence of a high-quality prior image. The few-view image that is marred by artifacts is first registered to the high-quality reference image using a registration algorithm that uses the SIFT features [196]. The registered reference image is then used to simulate an artifact-full few-view image. To remove the streak artifacts from the current image, its patches are matched with the simulated artifact-full image, but then the corresponding high-quality patches from the reference image are used to build the target image. This algorithm is further extended in [340] to be used when a prior scan from the same patient is not available but a rich database of scans from a large number of patients exists. The results of both these studies on real CT images of human head and lung are very good. Both methods substantially reduced the streaking artifacts in images reconstructed from less than 100 projections.

A major challenge facing the application of patch-based algorithms for large 3D CT images is the computational time. Although we discuss this challenge here under the post-processing methods, they apply equally to pre-processing methods and are indeed even more relevant to iterative reconstruction algorithms. Of course, one obvious approach to reducing the computational load is to work with 2D patches, instead of 3D blocks. However, this will likely hurt the algorithm performance because the voxel correlations in the 3rd dimension are not exploited. Three studies have reported that compared with 2D denoising, 3D denoising of CT images leads to an improvement in PSNR of approximately 1 to 4 dB [186, 187, 274]. Another study used 2D patches to denoise the slices in 3D CT images but patches were used from neighboring slices in addition to patches from the same slice [156]. They found that this approach increased the PSNR by more than 4 dB. Another obvious solution is to use faster hardware such as GPUs. This option has been explored in many studies. For instance, implementation of an NLM-type algorithm on a GPU reduced the computational time by a factor of 35 in one study [191]. Iterative reconstruction algorithms with non-local patch-based regularization terms have also been implemented on GPU [146, 314]. Another remarkable example was shown in [15], where the authors implemented the K-SVD algorithm for CT denoising on Cell Broadband Engine Architecture and achieved speedup factors between 16 and 225 compared with its implementation on a CPU.

There have also been many algorithmic approaches to reducing the computational time. An ingenious and highly efficient method to address this challenge was proposed in [274]. This method, which is named "double sparsity" is based on the observation that the learned dictionary atoms, themselves, have a sparse representation in a standard basis, such as DCT. The authors suggest a dictionary structure of the form $D = \Phi A$, where Φ is a basis with fast implicit implementation and A is a sparse matrix. They show that this dictionary can be efficiently learned using an algorithm similar to the K-SVD algorithm. Denoising of 3D CT images with this dictionary structure leads to speed-up factors of around 30, while also improving the denoising performance. A relatively similar idea is the separable dictionary proposed in [127], where the dictionary to be learned from data is assumed to be the Kronecker product of two smaller dictionaries. By reducing the complexity of sparse coding from $\mathcal{O}(n)$ to $\mathcal{O}(\sqrt{n})$, this dictionary model allows much larger patch/block sizes to be used, or alternatively, it results in significant speedups for equal patch size. A two-level dictionary structure was proposed in [186]. In this method, the learned dictionary atoms are clustered using a k-means algorithm that employs the coherence as the distance measure. For sparse coding of a test patch, a greedy algorithm is used to select the most likely atoms which are then used to obtain the sparse representation of the patch. Another study used the coherence of the dictionary atoms in learning a dictionary on a graph and reported very good results in 3D CT denoising [190].

For dictionary-based methods, the most computationally demanding part of the algorithm during both dictionary learning and usage is the sparse coding step. As we mentioned above, the image is usually divided into overlapping patches/blocks and the sparse representation of each patch/block in the dictionary has to be computed at least once (more than once if the algorithm is iterative). If the dictionary has no structure, which is the general case for overcomplete learned dictionaries, the sparse coding of each patch will require solving a small optimization problem. This will be computationally demanding, especially when the number and size of these patches/blocks are large such as in 3D CT. In recent years, many algorithms have been suggested for sparse coding of large signals in unstructured dictionaries. Some of these algorithms are basically faster implementations of traditional sparse coding algorithms [169, 275], while others are based on more novel ideas [35, 122, 179, 334]. Some of these methods have achieved several orders of magnitude speedups [35, 122]. A description of these algorithms is beyond the scope of this manuscript, but the computational edge that they offer makes patch-based methods more appealing for large-scale CT imaging.

For the NLM algorithms, the major computational bottleneck is the search for similar patches. We have described some of the state-of-the-art methods for reducing the computational load of patch search in Section 2.2. There has been little published research on how these techniques may work on CT images. One study has applied the method of integral image [73] on CT images. The same study reported that if the smoothing strength is properly adjusted, a very small search window and a very small patch size can be used, to obtain significant savings in computation.

2.7 Final remarks

Sparsity-based models have long been used in digital image processing. Recently, learned overcomplete dictionaries have been shown to lead to better results than analytical dictionaries such as wavelets in almost all image processing tasks. Nonlocal patch similarities have also been proven to be extremely useful in many image processing applications. Algorithms based on nonlocal patch similarities are considered to be the state of the art in important applications such as denoising. The practical utility of patch-based models has been demonstrated by hundreds of studies in recent years, many of which have been conducted on medical images. The use of learned overcomplete dictionaries for sparse representation of image patches and use of nonlocal patch similarities are at the core of much of the ongoing research in the field of image processing.

The published studies on the application of these methods for reconstruction and processing of CT images have reported very good results. However, the amount of research on the application of these methods in CT has been far less than that on natural images. Any reader who is familiar with the challenges of reconstruction and processing of CT images will acknowledge that there is an immense potential for these methods to improve the current state of the art algorithms in CT.

In terms of the pre-processing algorithms, there has been only a couple of published papers on patch-based algorithms. This is partly due to the fact that most of the patch-based models and algorithms have been originally proposed for the case of uniform Gaussian noise. For example, greedy sparse coding algorithms that form a central component of methods that use learned overcomplete dictionaries have been proposed for the case of Gaussian noise. As we mentioned in Section 2.4, only recently similar methods for the case of Poisson noise have started to appear. Nonetheless, even with the current tools, patch-based models can serve as useful tools for devel-

oping powerful pre-processing algorithms for CT. Some of the patch-based methods that we have reviewed in Section 2.4 have been applied on very noisy images (i.e., very low-count Poisson noise) and they have achieved impressive results. This might be extremely useful for low-dose CT, which is of special importance in clinical settings.

Iterative CT reconstruction algorithms that have used TV or patch-based regularization terms have reported very promising results. One can say that the published works have already demonstrated the usefulness of patch-based methods for CT reconstruction. However, many of the proposed algorithms have been applied on 2D images. In some cases it is not clear whether a proposed algorithm can be applied to large 3D reconstruction where the efficient implementations of forward and back-projection operations limit the type of iterative algorithm that can be employed. Moreover, little is known about the robustness of these algorithms in terms of the trained dictionary. As we mentioned in Section 2.1.2, the dictionary learning problem is non-convex and, hence, dictionary learning algorithms are not supported by strong theoretical guarantees.

Post-processing accounts for the largest share of the published papers on the application of patch-based methods in CT. Both denoising and restoration (e.g., artifact removal) algorithms have been proposed. Most of these papers have reported good results, even though many of them have used algorithms that have been originally proposed for natural images with little modification. Therefore, it is likely that much better results could be achieved by designing dedicated algorithms for CT. In fact, CT images, especially those reconstructed from low-dose scans, present unique challenges. Specifically, these images are contaminated by very strong noise with a non-uniform and unknown distribution. Moreover, they are also marred by various types of artifacts. This situation calls for carefully-devised algorithms that are tailored for CT images. Although this can be challenging, the success of patch-based methods on natural images can be taken as a strong indication of their potential to tackle these challenges. Patch-based methods have led to the best available denoising algorithms. Moreover, they have been successfully used for suppressing various types of artifacts and anomalies in natural images and videos. Therefore, they are likely to achieve state of the art denoising and restoration results in CT.

In conclusion, this review of the literature shows that sparsity-based and patch-based methods have a great potential to improve the current image reconstruction and image processing algorithms in CT. With an ever increasing usage of CT in clinical applications, it is necessary to reduce the radiation dose used for imaging so that CT can be used to its full potential.

Meanwhile, the increased computational power of modern computers makes it possible to use more sophisticated algorithms for image reconstruction and processing. Therefore, the methods reviewed in this chapter can play a key role in solving some of the major challenges facing CT.

Chapter 3

Sinogram Denoising via Simultaneous Sparse Representation in Learned Dictionaries

3.1 Introduction

In this chapter, we propose a novel algorithm for denoising the projection measurements in 3D CBCT. The noise model that we use in this chapter is the Gaussian model that we described in Section 1.2. Specifically, if we denote the line integral of the linear attenuation coefficient by $y_i = \int_i \mu ds$, then:

$$y_i \sim \mathcal{N}(\bar{y}_i, \sigma_i^2) \quad \text{where} \quad \sigma_i^2 = f_i \exp(\bar{y}_i) \tag{3.1}$$

where f_i is a factor that depends mainly on the effect of bowtie filtration. The value of f_i does not depend on the object being imaged and can be easily estimated from data [326].

Because of the nature of the projection measurements in CBCT, there is a strong correlation between neighboring pixels within a projection. There is also a strong correlation between the value of a pixel in neighboring projections. In order to exploit both of these correlations, we stack the projections together to form a large 3D image as shown in Figure 3.1. We assume that each projection is of size $n_u \times n_v$ pixels and that n_θ equally-spaced projections are acquired. The algorithm that we propose extracts small blocks of size m^3 pixels for processing. Throughout this chapter we use $m = 8$, which is a common patch/block size for dictionary-based image processing. We denote the whole 3D image of size $n_u \times n_v \times n_\theta$ with y. Blocks of size m^3 extracted from y are vectorized and stacked in a matrix denoted by $Y \in \mathbb{R}^{m^3 \times n}$, where n is the number of blocks. We will use Y_i to denote the ith vectorized block, which is the ith column of Y.

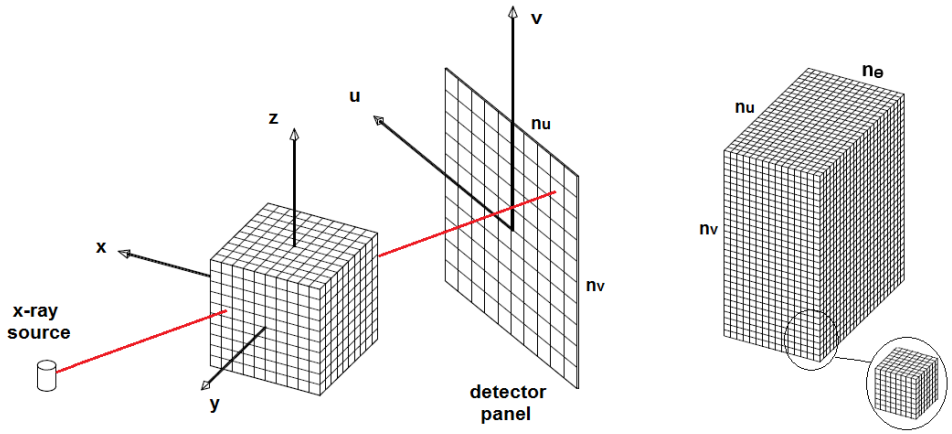

Figure 3.1: A schematic representation of the cone beam CT (left) and the stacked projections (right).

We described the method of image denoising with learned dictionaries in Section 2.1.3 and the method of NLM denoising in Section 2.2. We mentioned that both these methods have been shown to be effective denoising methods. Therefore, the algorithm that is proposed in this chapter combines the advantages of the two methods. As in the NLM algorithm, we find similar blocks and process them together in order to exploit the abundant self-similarities that exist in CT projections. However, unlike NLM that uses weighted averaging and unlike the BM4D algorithm ([71, 205], briefly described in Section 2.3) that uses thresholding in standard bases, we perform simultaneous sparse coding of the similar blocks in a learned dictionary. In this sense, the proposed algorithm is similar to the non-local sparse model proposed in [214].

In brief, using training data we first learn a dictionary for simultaneous sparse coding of similar blocks. Given a new set of noisy projections, we group the similar blocks and denoise them via simultaneous sparse coding in the learned dictionary. The main intuition behind our proposed algorithm is that similar blocks must have a jointly sparse representation in the learned dictionary. In other words, sparse representation of similar blocks must use similar atoms in the learned dictionary.

3.2 The proposed algorithm

3.2.1 Clustering

As mentioned above, the algorithm proposed in this chapter first learns a dictionary for simultaneous sparse representation of similar blocks. Then, this dictionary is used for denoising. Since both of these steps involve clustering of similar blocks, we start by explaining our clustering algorithm.

In the NLM algorithm and similar algorithms such as BM4D, desnoising of the ith pixel requires searching the entire image or a sizable neighborhood around this pixel to find blocks that resemble the block centered on this pixel. Since this search has to be performed for every block, it is computationally very intensive. In the algorithm proposed in this chapter, we follow an approach that is much less computationally demanding. Specifically, we divide the stacked projections into overlapping blocks and cluster all these blocks only once.

Consider a particular block, Y_i, and suppose that we would like to decide whether block Y_j is similar to Y_i so that they can be clustered together. We make two simplifying assumptions: (1) the value of the true projection does not change drastically between neighboring pixels, and (2) the noise in adjacent pixels are independent. These are mild assumptions and are satisfied to a high degree in practice. From these assumptions, we can compute the average noise variance in block Y_i, which we denote as σ_i, using Equation (3.1). Then, if values of Y_i and Y_j are close, the variable $\|Y_i - Y_j\|_2^2/(2\sigma_i^2)$ will follow a central chi-squared distribution with m^3 degrees of freedom, $\chi^2_{m^3}$. Denoting its cumulative distribution function with $F_{m^3}(t)$, a good choice for similarity threshold will be $\alpha = (2\sigma_i^2)F^{-1}_{m^3}(t_0)$ for some parameter t_0 [214, 215]. Here, $F^{-1}_{m^3}$ is the inverse of F_{m^3}. In other words, we decide that Y_i and Y_j are similar if their squared Euclidean distance $\|Y_i - Y_j\|_2^2$ is less than or equal to α. We have empirically found that a value of $t_0 \simeq 0.90$ leads to good clustering results for all noise levels. The same value has been suggested in [214].

A straightforward clustering would require computing the distance $\|Y_i - Y_j\|_2^2$ for each pair of blocks that we wish to compare. Because the dimension of the blocks and their number are large, we suggest a two-step approach to reduce the computation. Our approach is similar in concept to the locality-sensitive hashing methods that are common in the field of data mining for finding nearest neighbors in high-dimensional data [181]. Specifically, we map each block Y_i, which resides in $\mathbb{R}^{512}$, onto $\mathbb{R}^4$ using the following com-

putations:

$$
\begin{aligned}
h_i(1) &= \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m} Y_i(i,j,k) \\
h_i(2) &= \sum_{i=1}^{m/2}\sum_{j=1}^{m}\sum_{k=1}^{m} Y_i(i,j,k) - \sum_{i=m/2+1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m} Y_i(i,j,k) \\
h_i(3) &= \sum_{i=1}^{m}\sum_{j=1}^{m/2}\sum_{k=1}^{m} Y_i(i,j,k) - \sum_{i=1}^{m}\sum_{j=m/2+1}^{m}\sum_{k=1}^{m} Y_i(i,j,k) \\
h_i(4) &= \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m/2} Y_i(i,j,k) - \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=m/2+1}^{m} Y_i(i,j,k)
\end{aligned}
\tag{3.2}
$$

It is easy to see that $h_i(1)$ is a measure of average projection amplitude in the ith block whereas $h_i(2)$, $h_i(3)$, and $h_i(4)$ estimate its slope along the three dimensions. We can interpret these computations as a "projection" of Y_i from $\mathbb{R}^{512}$ onto $\mathbb{R}^4$. In locality-sensitive hashing, high-dimensional signals are projected onto a low-dimensional space where they can be clustered at a much lower computational cost. The choice of projections is highly critical because similar signals in the original high-dimensional space must remain close to each other in the low-dimensional space defined by the projections. Conversely, signals that are far apart in the original space should remain far apart in the low-dimensional space. Our choice of the projections in (3.2) is motivated by the nature of CT projections as smooth and slowly varying signals. Therefore, we hope that, in general, Y_i and Y_j are close together if and only if h_i and h_j are so. In Figure 3.2 we have shown the effectiveness of the proposed mapping on projections simulated from a Shepp-Logan phantom. We selected 10^5 random pairs of blocks from simulated projections of this phantom and computed the Euclidean distances in the Y and h spaces for each pair of blocks. In part (a) of this figure we have shown the result for noise-free projections. In part (b) of the same figure we have shown the result for very noisy projections that were simulated by assuming the number of incident photons to be 100. It is clear that the proposed mapping is highly effective. What is particularly important to us is that blocks that are very close to each other in the Y space are also very close in the h space.

These projections are computed for all Y_i prior to clustering and they are normalized so that each of the four projections has a mean of zero and standard deviation equal to one. During clustering, every time two signals

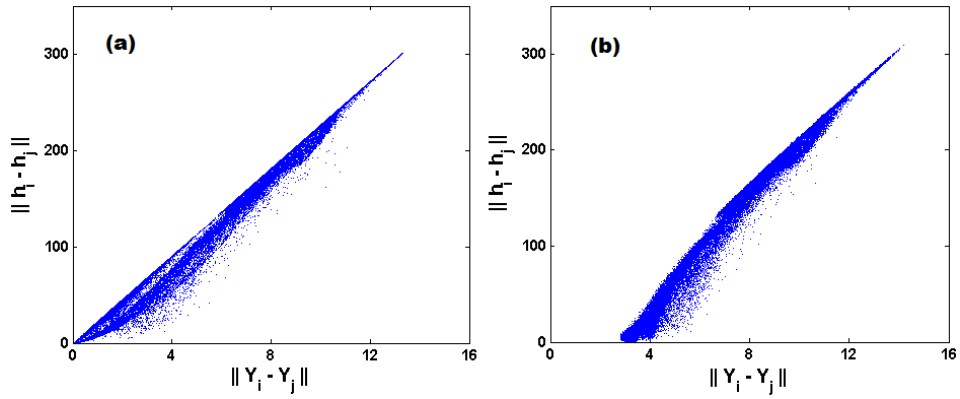

Figure 3.2: The Euclidean distance between blocks extracted from projections of Shepp-Logan phantom in the h space versus that in the Y space. (a) noise-free projections, (b) very noisy projections.

Y_i and Y_j are to be compared, we first compare their projections, h_i and h_j. Only if h_i and h_j are sufficiently close, we will proceed to compare Y_i and Y_j. As we mentioned above, we use the ℓ_2 norm to compare Y_i and Y_j because our noise model allows us to find a reasonable threshold for $\|Y_i - Y_j\|_2^2$. For comparing h_i and h_j, however, we use the ℓ_∞-norm. This is because we expect that two blocks Y_i and Y_j be similar only if their mean amplitude and slope in all three directions are close to each other. Therefore, when we would like to decide if Y_i and Y_j are close enough to be clustered, first we test if $\|h_i - h_j\|_\infty \leq \epsilon_h$ for some threshold ϵ_h, and only if this is the case then we will proceed to test $\|Y_i - Y_j\|_2^2 \leq \alpha$. If this latter test is also satisfied, then we will decide that Y_j can be clustered with Y_i. This two-step scheme will drastically reduce the number of comparisons between 512-dimensional blocks needed for clustering.

The choice of threshold ϵ_h is very important. If ϵ_h is too small, many blocks that are truly close will not be clustered together, reducing the performance of the algorithm. On the other hand, if ϵ_h is too large, the number of comparisons of high-dimensional signals that are not truly similar will increase, increasing the computational time with no gain in the denoising performance. We have empirically found that a value of $\epsilon_h \simeq 0.5$ works well for all noise levels (as we mentioned above, the computed values of h are normalized to have a standard deviation of 1). The reason why this value works well for different noise levels is because the features that constitute

our proposed mapping (h) are not much affected by the noise. As can be seen in Equation (3.2), all four features that constitute h involve a large amount of summing (or, equivalently, averaging) of the elements of Y. As a result, the values of h are very robust to the noise level. Consequently, the proper value of the threshold ϵ_h is also largely unaffected by the noise level.

The proposed clustering algorithm is presented in Algorithm 1. The algorithm sweeps through the columns of Y (we remind the reader that columns of Y are blocks extracted from stacked projections that have been vectorized). If column Y_i has not been clustered yet, a new cluster is defined with Y_i as its representative element. The algorithm will then check all unclustered columns and adds them to the newly-formed cluster if they are sufficiently close to Y_i based on the criteria described above. As we explained above, first all unclustered columns are compared with Y_i in terms of their projections (h). Those that are close to Y_i in terms of their projections, denoted with I_2 in Algorithm 1, are then tested to identify those that are truly close to Y_i in the original space.

input : matrix $Y \in \mathbb{R}^{m^3 \times n}$ containing vectorized blocks as its columns
output: sets of indices of similar blocks, S
$n_{\text{cluster}} = 0$
$S = \{\}$
for $i \leftarrow 1$ **to** n **do**
  if $i \notin S\{k\} \;\; \forall k = 1 : n_{cluster}$ **then**
    $n_{\text{cluster}} = n_{\text{cluster}} + 1$
    $\sigma^2 = f_i \exp(\bar{Y}_i)$
    $\alpha = (2\sigma^2) F^{-1}_{m^3}(0.90)$
    $I_1 = \{j \in [i+1, n] : j \notin S\{k\} \;\; \forall k = 1 : n_{\text{cluster}}\}$
    $I_2 = \{j \in I_1 : \|h_j - h_i\|_\infty \leq \epsilon_h\}$
    $I_3 = \{j \in I_2 : \|Y_j - Y_i\|_2^2 \leq \alpha\}$
    $S\{n_{\text{cluster}}\} = i \cup I_3$
  end
end

Algorithm 1: Algorithm for clustering of the blocks extracted from the stacked projections.

3.2.2 Dictionary learning

As mentioned above, the core idea in the proposed algorithm is to exploit both the self-similarity in the stacked projections and the power of learned dictionaries by collecting similar blocks in the stacked projections and computing their simultaneous sparse representation in a properly designed dictionary. Therefore, the central role of the dictionary in this algorithm is obvious. In this section, we describe how we learn the dictionary.

Given a set of training projections, we first select 3D blocks from different locations in the stacked projections and vectorize them as columns of a matrix Y. In this section, we refer to columns of Y as "training signals". We cluster these training signals using Algorithm 1, and obtain a set $S = \{s_1, s_2, ..., s_{n_{\text{cluster}}}\}$, where each s_i contains the indices of similar training signals. A good dictionary can then be learned by solving the following optimization problem:

$$\underset{D,\Gamma}{\text{minimize}} \; \|Y - D\Gamma\|_F^2 + \lambda \sum_{i=1}^{n_{\text{cluster}}} \|\Gamma_{s_i}\|_{\text{row-0}} \tag{3.3}$$

In the above equation, Γ_{s_i} denotes the matrix Γ restricted to columns indexed by s_i. In other words, Γ_{s_i} is the matrix of representation coefficients of all signals in cluster s_i. $\|.\|_{\text{row-0}}$ is a pseudo-norm that counts the number of non-zero rows; that is, $\|\Gamma_{s_i}\|_{\text{row-0}}$ counts the number of rows of Γ_{s_i} that have at least one non-zero element. In other words, $\|\Gamma_{s_i}\|_{\text{row-0}}$ counts the number of atoms of the dictionary D that participate in the representation of at least one of the signals in cluster s_i. Therefore, using $\|\Gamma_{s_i}\|_{\text{row-0}}$ as a penalty encourages atoms in the set s_i to share the same atoms and to use a small number of atoms in their representation. In summary, the form of the cost function in (3.3) reflects what we expect from the trained dictionary: the first term requires that the dictionary accurately model the training signals, while the second term requires that the training signals have a sparse representation in D and that similar training signals share the same atoms from the dictionary, i.e., to have a joint-sparse representation in D. This is precisely in accordance with the intuition behind our proposed algorithm.

As we mentioned above, (3.3) is a non-convex minimization problem and is commonly solved by alternately minimizing with respect to Γ and D. Below, we will explain how we perform these minimizations. In our presentation, we will assume that $Y \in \mathbb{R}^{m^3 \times N}$ and $D \in \mathbb{R}^{m^3 \times K}$, i.e., the number of training blocks is N and the trained dictionary has K atoms. We also assume that the columns of D are normalized so that they have ℓ_2 norm of unity.

Minimization with respect to D: Our approach here is similar to the approach followed in the K-SVD algorithm [2]. K-SVD algorithm sweeps through the columns of D (i.e., dictionary atoms) and updates each atom along with the coefficients of the signals that use it. Details were given in Section 2.1.2.

Minimization with respect to Γ: With D being fixed, the minimization problem in this step can be written as:

$$\underset{\Gamma}{\text{minimize}} \sum_{i=1}^{n_{\text{cluster}}} \left(\|Y_{s_i} - D\Gamma_{s_i}\|_F^2 + \lambda \|\Gamma_{s_i}\|_{\text{row-0}} \right) \tag{3.4}$$

where, as before, Y_{s_i} and Γ_{s_i} denote the restriction of these matrices to the columns indexed by s_i. From (3.4) we see that the optimization problem in this step consists of n_{cluster} separate subproblems, one for each cluster of signals. Each of these subproblems is a simultaneous sparse coding problem. Since we can estimate the noise variance, we prefer to re-write (3.4) in a constrained form:

$$\left(\underset{\Gamma_{s_i}}{\text{min.}} \ \|\Gamma_{s_i}\|_{\text{row-0}} \quad \text{subject to:} \quad \|Y_{s_i} - D\Gamma_{s_i}\|_F^2 \leq \epsilon_i \right) \ i = 1 : n_{\text{cluster}}$$

The advantage of this constrained minimization over the unconstrained formulation in (3.4) is that λ is unknown and has no physical meaning, whereas ϵ_i is directly related to the noise variance, which we can easily estimate. We solve each of these subproblems using Algorithm 2, which is an extension of the famous OMP algorithm [319]. In this algorithm, $|s_i|$ denotes the cardinality of the set s_i. At each iteration of this algorithm, we select the atom that has the largest cumulative correlation with the signals in Y_{s_i} and then project Y_{s_i} on the subspace spanned by the set of atoms selected so far. We then update the residual and repeat.

An important choice is the dictionary size, i.e., the number of atoms in the dictionary. We used a dictionary size of 1024 in all our experiments. We will discuss the effect of the dictionary size on the performance of the proposed algorithm later in this chapter. Another important decision is the choice of the initial dictionary. It is common to use an overcomplete DCT or wavelet dictionary as the initial dictionary. Our experience shows that the algorithm converges much faster if we build the initial dictionary from the training signals. For example, 1024 training signals can be randomly selected. Alternatively, after clustering the training signals, 1024 clusters can be randomly selected and one random signal from each cluster or the

input : Dictionary D and the set of similar signals Y_{s_i}
output: Γ_{s_i}, sparse representation coefficients of Y_{s_i} in D
$r = Y_{s_i}$
$I = \{\}$
$\sigma^2 = f_i \exp(\bar{Y}_{s_i})$
$\alpha = (\sigma^2) F^{-1}_{|s_i| m^3}(0.90)$
while $||r||_F^2 > \alpha$ **do**
  $\hat{k} = \text{argmax}_k \sum_{j=1}^{|s_i|} |\langle (Y_{s_i})_j, D_k \rangle|$
  $I = I \cup \hat{k}$
  $\Gamma_{s_i} = (D_I^T D_I)^{-1} D_I^T Y_{s_i}$
  $r = Y_{s_i} - D_I \Gamma_{s_i}$
end

Algorithm 2: Simultaneous greedy sparse coding [319].

average of all atoms in each cluster be used in the initial dictionary. With this initialization, at most 50 iterations of the proposed algorithm were enough to converge to a good dictionary.

Moreover, at the end of each iteration of the dictionary learning algorithm we removed from the dictionary the least-used atom (the atom used by the least number of signal groups) and instead added a new dictionary atom that was formed as the average of the most difficult group of signals (the group that used the largest number of dictionary atoms in its representation). Similar strategies are commonly used in dictionary learning algorithms.

3.2.3 Denoising

Once a dictionary is learned for simultaneous sparse representation of similar blocks using the method described above, it can be used for denoising of a newly-acquired projection set. Of course, one can learn the dictionary from the noisy image itself. This is indeed a common approach in dictionary-based image denoising and may lead to slightly better results [100, 215]. However, dictionary learning is very computationally expensive and it would be highly desirable if a pre-trained dictionary could be used, instead of learning a new dictionary for newly-acquired projections. As we will describe later in this chapter, our results show that for CT projections, as long as the scan geometry does not change, a well-trained dictionary can be used for denoising the projections of different objects without a loss of performance.

Denoising of a new set of projections is carried out in three simple steps

described below.

- Partition the stacked projections into overlapping blocks and cluster them using Algorithm 1.
- For each cluster of blocks Y_{s_i}, use the simultaneous sparse coding algorithm outlined in Algorithm 2 to find the representation coefficients of Y_{s_i} in D. Denoting these representation coefficients by $\hat{\Gamma}_{s_i}$, the denoised estimate of the blocks in this cluster will be: $\hat{Y}_{s_i} = D\hat{\Gamma}_{s_i}$.
- The final (denoised) estimate of the projections is computed using a simple averaging:

$$\hat{Y} = \left(\sum_{i=1}^{n_{\text{cluster}}} \sum_{j \in s_i} R_j (D\hat{\Gamma}_{s_i})_j \right) \oslash \left(\sum_{i=1}^{n_{\text{cluster}}} \sum_{j \in s_i} R_j \mathbf{1} \right) \tag{3.5}$$

  where R_j is a binary matrix that places the jth block in its location in the stacked projections, $\oslash$ indicates element-wise division, and $\mathbf{1} \in \mathbb{R}^{m^3}$ is a vector of all ones. The above equation has a simple meaning similar to Equation 2.7.

3.3 Evaluation

We apply the proposed denoising algorithm on simulated and real cone-beam CT projections and compare its performance with the following algorithms:

- **Bilateral filtering**, which has been suggested for sinogram denoising in [219]. Since this is not a patch-based algorithm, we expect that it should be faster than patch-based algorithms but to be less effective in terms of denoising performance. We include it here to contrast the patch-based denoising algorithms with the more traditional sinogram denoising methods. This algorithm estimates the denoised value of the image at pixel k by minimizing a cost function of the form $E(u(k)) = \sum_{k' \in \Omega_k} P_1(k,k')P_2(k,k')$ where Ω_k is a neighborhood around this pixel, and P_1 and P_2 are two cost functions in terms of the spatial distance and difference in pixel values, respectively. In [219] both P_1 and P_2 are suggested to be Gaussians. The bandwidth of P_1 is suggested to be fixed at $w/6$, where w is the neighborhood width, and the bandwidth of P_2 is suggested to be chosen in the range $[0.7, 2.8]$.

- We will apply the dictionary-based approach proposed in [100] to denoise the projection measurements. We will refer to this algorithm as **K-SVD**.

- We will apply the BM4D denoising algorithm [205] on the projection measurements. There are many parameters that influence the performance of this algorithm. We use the set of parameter values that has been named "normal profile" in [205]. These parameter values provide a good balance between speed and performance for this algorithm. For brevity, we will refer to the BM4D algorithm applied on the projections (sinogram) as **BM4D-s**.

- We will apply the BM4D algorithm also in the image domain. In other words, we reconstruct the image from noisy projections and then denoise the reconstructed image. Although this is the same BM4D algorithm, we will refer to it as **BM4D-i** to distinguish it from BM4D-s that is applied on the sinogram.

- An adaptive NLM algorithm was proposed for CT image denoising in [191]. We will apply this algorithm on the images reconstructed from noisy projections. We will refer to this algorithm as **ANLM**. An important difference between ANLM and the basic NLM algorithm is that ANLM relies on a relatively fast approximation of the noise level in the reconstructed image and adjusts the denoising strength based on the estimated noise level.

To have a fair comparison between different algorithms, we will use the same block size (i.e., 8^3) and the same level of overlapping by using a 3-pixel shift so that adjacent overlapping blocks share 5 pixels in each direction. Therefore, the methods that we compare are of two types: 1) the sinogram denoising (pre-processing) algorithms, which include bilateral filtering, K-SVD, BM4D-s, and the proposed algorithm, and 2) the post-processing algorithms that work on the reconstructed image, which include BM4D-i and ANLM. As we will explain below, we apply these algorithms on several sets of simulated and real cone-beam projections. Since the ultimate goal is to achieve a high-quality image, most of our evaluations will be done in the image domain. Therefore, for each of the experiments that we will describe below, we will apply these three types of algorithms as follows: 1) For the sinogram-denoising algorithms (i.e., bilateral filtering, BM4D-s, K-SVD, and the proposed algorithm), after denoising the projections with each algorithm we use the FDK algorithm to reconstruct the image, 2) For

Bilateral filtering	K-SVD	BM4D-s	Proposed algorithm
0.060	0.055	0.051	0.048

Table 3.1: Root-Mean-Square of the difference between the denoised projections and the true projections on the data simulated from the digital brain phantom.

the post-processing algorithms (i.e., BM4D-i and ANLM), we reconstruct the image using the FDK algorithm from noisy projections and then apply these algorithms to denoise the reconstructed image. Therefore, all algorithms will include an FDK reconstruction. We use a Hamming window with the FDK algorithm.

3.3.1 Simulation experiment

We simulated 1440 equally-spaced noisy projections from a 3D digital brain phantom. We generated this phantom using one of the phantoms in the BrainWeb database [63]. The phantom size was $362 \times 434 \times 362$ voxels and the projections were each 650×450 pixels in size. We assumed the number of incident photons $N_0^i = 10^5$ to be constant for all i by ignoring the bowtie filtration. To simulate the electronic noise that becomes more important in low-dose CT, we added a Gaussian noise with a standard deviation of $\sigma = 100$ to the detected photon counts.

For this simulation study, since we know the true phantom projections, we can quantitatively compare the sinogram denoising algorithms in the projection domain. For this purpose, we computed the Root Mean Square of the Error (RMSE), where error is defined as the difference between the denoised and the true (i.e., noise-free) projections. Table 3.1 shows this comparison. Note that this comparison only includes the sinogram denoising algorithms, i.e., bilateral filtering, K-SVD, BM4D-s, and the proposed algorithm. It can be seen from this table that the proposed algorithm has achieved a lower RMSE than the other sinogram-denoising algorithms. As one might expect, bilateral filtering is not as effective as the patch-based denoising algorithms.

A more useful comparison is in terms of the quality of the reconstructed image, where we can compare all three types of algorithms. To assess the quality of the reconstructed images, we compute the RMSE, where error is the difference between the reconstructed image and the true phantom image. We also compute the structural similarity index (SSIM) between the

	RMSE	SSIM	time (h)
Bilateral filtering	0.066	0.735	0.40
KSVD	0.065	0.738	1.84
BM4D-s	0.059	0.765	4.35
Proposed algorithm	0.057	0.774	0.44
BM4D-i	0.055	0.770	0.82
ANLM	0.062	0.758	1.70

Table 3.2: Comparison of different algorithms in terms of the quality of the reconstructed image and the computational time on the noisy scan simulated from the digital brain phantom.

reconstructed image, $\hat{x}$, and the true image, x_0, as follows [329]:

$$\mathrm{SSIM}(\hat{x}, x_0) = \frac{(2\mu_{\hat{x}}\mu_{x_0} + C_1)(2\sigma_{\hat{x}x_0} + C_2)}{(\mu_{\hat{x}}^2 + \mu_{x_0}^2 + C_1) + (\sigma_{\hat{x}}^2 + \sigma_{x_0}^2 + C_2)} \tag{3.6}$$

where μ_x and σ_x represent the mean and standard deviation of x, $\sigma_{\hat{x}x_0}$ is the covariance, and C_1 and C_2 are constants.

The results of the quantitative comparison are presented in Table 3.2. In addition to RMSE and SSIM, we have also presented the computational times. From this table, the proposed algorithm has performed better than the other sinogram denoising algorithms and ANLM. The image produced by BM4D-i is close to the image produced by the proposed algorithm. Another very important observation is that the computational time of the proposed algorithm is much shorter than that of K-SVD, BM4D-s, BM4D-i, and ANLM. In fact, the computational time of the proposed algorithm is not much longer than that of bilateral filtering, whereas the computational times of other algorithms are approximately 2 to 11 times longer than that of bilateral filtering. All algorithms were run in Matlab version R2012b on a Windows 7 PC with 32 GB of memory and 3.4 GHz Intel Core i7 CPU. We should point out that the computation times reported for K-SVD and the proposed algorithm in Table 3.2 do not include the time for the dictionary learning stage. As we will describe later in this chapter, it is not always necessary to learn a new dictionary for every new set of projections and, in many cases, a dictionary learned on projections of one object can be effectively used for denoising of the projections of a different object.

Figure 3.3 shows a slice of the digital brain phantom in the images reconstructed with different algorithms. Compared with other sinogram denoising

algorithms (i.e., K-SVD, BM4D-s, and bilateral filtering) and ANLM, the proposed algorithm has resulted in a better image quality. The visual quality is not very different between the proposed algorithm and BM4D-i.

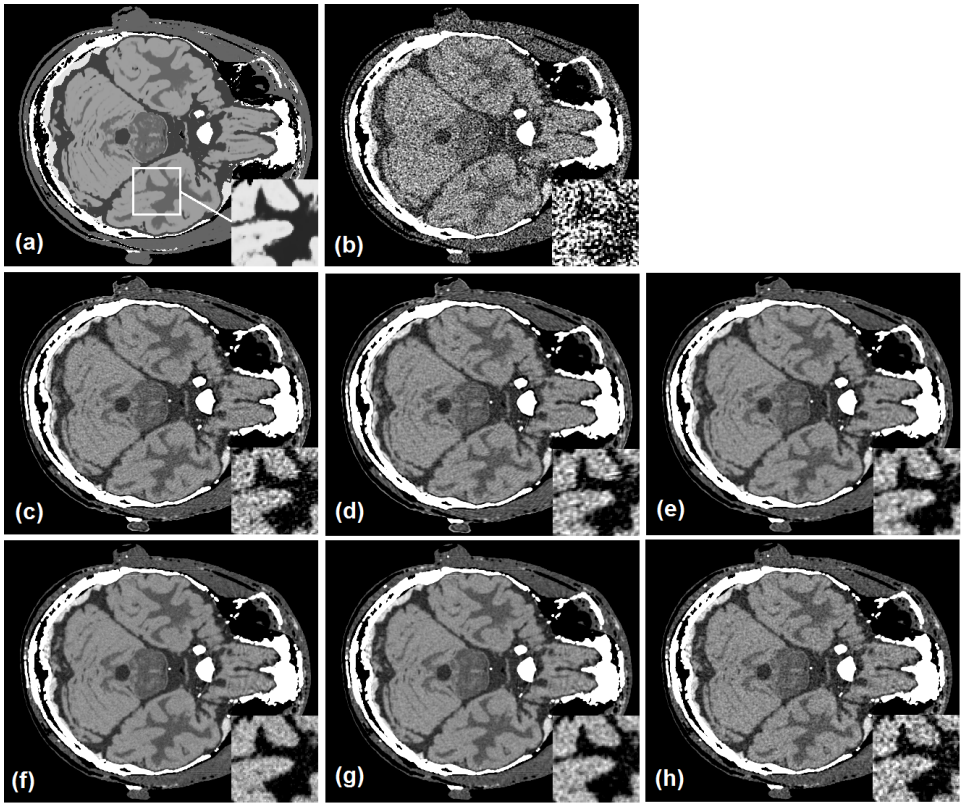

Figure 3.3: A slice of the digital brain phantom. (a) reference image, (b) FDK-reconstructed from noisy projections, (c) bilateral filtering, (d) K-SVD, (e) BM4D-s, (f) the proposed algorithm, (g) BM4D-i, (h) ANLM.

In order to further evaluate the proposed algorithm in terms of the spatial resolution, we performed a second simulation experiment. The goal of this experiment was to determine how the spatial resolution for different algorithms is affected by the object contrast. We rely on the estimation of the modulation transfer function (MTF) following an approach similar to that in [209]. In this experiment, we simulated noisy projections from the MTF bead phantom, which we generated using the CONRAD software [208]. As shown in Figure 3.4(a), this digital phantom includes three small

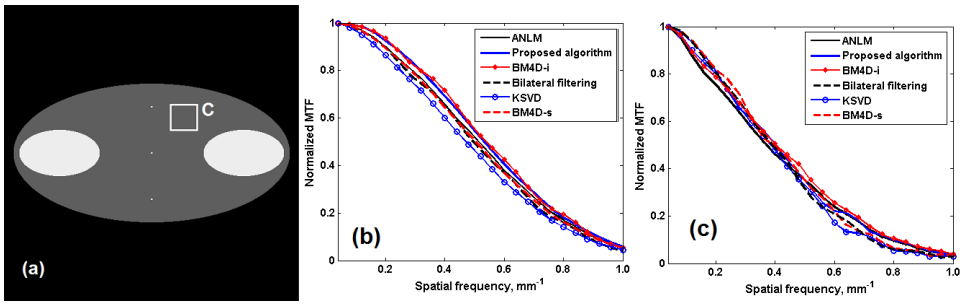

Figure 3.4: (a) The central slice of the MTF bead phantom; the square C shows the location of the cube used to compute the noise in the reconstructed images to ensure equal noise for all algorithms, (b) estimated MTF for the high-contrast phantom, (c) estimated MTF for the low-contrast phantom.

beads and two large high-attenuation inserts. We generated two versions of this phantom. In both versions, the phantom disk was assumed to have a linear attenuation coefficient of 1. However, in one of the phantoms the beads had a linear attenuation coefficient of 3, whereas in the other they had a linear attenuation coefficient of 1.4. We will refer to these two phantoms as high-contrast and low-contrast MTF bead phantoms, respectively. If we regard the phantom disk as the background, the contrast of the beads in the low-contrast phantom was 1/5 that in the high-contrast phantom. The phantom size was 512^3 voxels and the projections were each 600×600 pixels in size. We simulated 720 equally-spaced noisy projections from each phantom by assuming the number of incident photons to be $N_0^i = 2 \times 10^3$ and the standard deviation of the additive Gaussian noise to be $\sigma = 50$. We estimated the MTF from the center bead in the reconstructed images.

In addition to contrast, another factor that affects the spatial resolution is the amount of smoothing (i.e., the denoising strength). In fact, noise, contrast, and spatial resolution are three inter-dependent criteria. In order to also account for the noise level, we adjusted the denoising strengths of the sinogram-denoising and image-domain denoising methods such that the noise in the images produced by different algorithms were approximately equal. Each of these algorithms have tuning parameters that allow for adjustment of the denoising strength. In this simulation experiment we considered a cube, the cross-section of which is shown in 3.4(a), and adjusted the denoising strengths of the sinogram-denoising and image-domain denoising methods such that the variance of the voxel values in this cube was

	Spatial resolution (high-contrast)	Spatial resolution (low-contrast)
Bilateral filtering	0.86	0.75
KSVD	0.85	0.74
BM4D-s	0.88	0.79
Proposed algorithm	0.88	0.80
BM4D-i	0.90	0.82
ANLM	0.90	0.81

Table 3.3: Spatial resolution (defined as the spatial frequency at which the normalized MTF reaches 0.10) in units of mm^{-1} in the images of the MTF bead phantom produced by different algorithms.

approximately equal in the images produced by different algorithms.

The estimated MTFs from the high-contrast and low-contrast phantom images produced by different algorithms are shown in Figure 3.4(b)-(c). In Table3.3 we have shown the spatial frequency at which the normalized MTF for different algorithms reached a value of 0.10. These results show that the images produced by the proposed algorithm have a higher spatial resolution than the images produced by the other three sinogram denoising algorithms. The two image-based denoising algorithms also achieve high spatial resolution and outperform the sinogram-denoising algorithms including our proposed algorithm. From the MTF plots for the low-contrast phantom shown in Figure 3.4(c), there is also a general difference between the sinogram-denoising and image-domain denoising methods. In general, the sinogram denoising methods have resulted in higher MTF at low spatial frequencies but the image-domain denoising methods have resulted in higher MTF at higher spatial frequencies.

3.3.2 Experiment with micro-CT scan of a rat

All of the real CT data that is used in this dissertation were collected with a Gamma Medica eXplore CT 120 micro-CT scanner. Therefore, here we provide some technical details about this scanner. This scanner has a flat panel detector located 449 mm from the source and 397 mm from the axis of rotation. The detector panel includes 3500×2272 detector elements. Using three different binning options of 1×1, 2×2, and 4×4, projections with three different sizes of 3500×2272, 1750×1136, 875×568, respectively, can be obtained. The distance between the centers of adjacent detector elements

is $0.02831mm$. Unless otherwise stated, the size of the reconstructed images is $880 \times 880 \times 650$ voxels, each voxel having a size $0.1 \times 0.1 \times 0.1\ mm^3$. Other scan settings are different for different experiments; therefore they will be stated for each experiment separately. Some further information about this scanner can be found in [203].

For the rat scan used in this section, tube voltage, tube current, and exposure time were equal to 70 kV, 32 mA, and 16 ms, respectively. This was the lowest possible setting in terms of mAs because the scanner did not operate under 0.5 mAs. The scan consisted of 720 projections between $0°$ and $360°$ at $0.5°$ intervals. The size of each projection was 875×568 pixels.

Because we do not have access to the true (i.e., noise-free) projections, we evaluate the performance of different algorithms in terms of the quality of the reconstructed images. For this purpose, we reconstructed a high-quality image of the rat using the full set of 720 projections. To create this image, we reconstructed an initial image using the FDK algorithm followed by 50 iterations of the MFISTA algorithm [17] to improve the quality of the FDK-reconstructed image. The resulting image had a very high quality and we will refer to it as "the reference image".

To compare different algorithms, we applied them on the same scan. As we did in the simulation studies described above, in our experiments with real data we applied the sinogram denoising algorithms (i.e., bilateral filtering, K-SVD, BM4D-s, and the proposed algorithm) on the noisy projections and used the denoised projections to reconstruct an image with the FDK algorithm. We applied the post-processing algorithms (i.e., BM4D-i and ANLM) on the FDK-reconstructed image from the noisy projections. The quality of the reconstructed images was assessed by computing the RMSE, where we define the error as the difference between the reconstructed image and the reference image, and the SSIM between the two images. In Table 3.4, we have summarized the results of the quantitative comparison between different algorithms. In addition to RMSE, SSIM, and the computational time, we have included two additional numbers that are indicators of spatial resolution and noise strength, which we have denoted by SR and NS and are computed as follows:

- As an indicator of spatial resolution (SR), we computed the maximum absolute value of the gradient (i.e., slope) along the line L marked in Figure 3.5(a). A larger gradient, corresponding to a sharper slope, indicates a higher spatial resolution.

- As a measure of noise strength (NS), we computed the standard deviation of the voxel values in a cube whose cross-section has been marked

in Figure 3.5(a) with the rectangle C. From the reference image, we identified this cube to be highly uniform.

Noise suppression and spatial resolution are usually two opposing objectives in denoising. A stronger denoising, in general, leads to a loss of spatial resolution and a successful denoising algorithm can be simply defined as one that reduces the noise with little degradation of spatial resolution. Therefore, the above two numbers are very useful indicators for comparing different algorithms. A very similar approach was suggested for examining the trade-off between noise removal and spatial resolution in CT images in [219].

	RMSE	SSIM	time (h)	SR	NS
Bilateral filtering	0.0123	0.783	0.42	0.160	0.138
KSVD	0.0121	0.792	2.2	0.156	0.124
BM4D-s	0.0116	0.808	2.8	0.165	0.119
Proposed algorithm	0.0106	0.814	1.0	0.165	0.116
BM4D-i	0.0110	0.810	3.9	0.160	0.112
ANLM	0.0118	0.803	4.6	0.162	0.123

Table 3.4: Quality criteria for the images produced by different algorithms from the noisy rat scan.

The numbers in Table 3.4 show that the propsoed algorithm has outperformed other sinogram denoising algorithms and image-domain denoising algorithms, while having a much shorter computational time. The proposed algorithm is, in most cases, better than other methods in terms of the spatial resolution and the noise strength. Bilteral filtering has a shorter computational time but it is less effective than the other algorithms. In terms of computational time, the proposed algorithm requires a much shorter time than other methods, except for bilateral filtering. BM4D-s and BM4D-i, which are close to the proposed algorithm in terms of the image quality criteria, take 3 to 4 times longer to complete.

Figure 3.5 shows a slice in the image of the rat produced using different algorithms. The entire slice has been displayed with a window of linear attenuation coefficient of $[0, 0.45]$. In order to better demonstrate the difference between the images, we have selected two ROIs and displayed them with a magnification of 150% and with much narrower windows of linear attenuation coefficients. The locations of these ROIs have been marked on

the slice of the reference image in Figure 3.5(a). The ROI shown on the top-left of each slice contains fat surrounded by soft tissue and is shown with a window of linear attenuation coefficient of $[0.14, 0.22]$. The ROI shown on the top-right of each slice contains bone surrounded by soft tissue and is displayed with a window of linear attenuation coefficient of $[0.18, 0.30]$. From this figure, the proposed algorithm seems to have resulted in a better image than other sinogram denoising algorithms and post-processing algorithms. This is more visible from the ROI displayed on the top left of each slice.

3.3.3 Experiment with micro-CT scan of a phantom

A physical phantom was scanned using the same micro-CT scanner described above. This is a quality assurance phantom that has been designed for comprehensive evaluation of the performance of micro-CT systems. It has various modules that allow for a complete evaluation of the quality of the reconstructed images in terms of spatial resolution, noise, geometric accuracy, linearity, etc. Therefore, in this dissertation this phantom is used very frequently and referred to as "the physical phantom". A detailed description of this phantom can be found in [92].

For evaluation of the algorithm proposed in this chapter, the physical phantom was scanned twice:

1. **Low-noise scan.** This scan consisted of 720 projections at $0.5°$ intervals between $0°$ and $360°$. The tube voltage, tube current, and exposure time were 70 kV, 40 mA, and 25 ms, respectively.

2. **High-noise scan.** This scan consisted of 720 projections between $0°$ and $360°$ at $0.5°$ intervals. The tube voltage, tube current, and exposure time were equal to 50 kV, 32 mA, and 16 ms, respectively. Moreover, a $0.2\,mm$ copper filter was used for this scan.

The phantom had no movement between the two scans. Therefore, the two scans are from the same object at exactly the same location. Note that it was not possible to perform two identical scans in the rat experiment described above because the internal organs of the rat moved ever so slightly during the experiments.

From the low-noise scan, we reconstructed a high-quality reference image in a way similar to our rat experiment above. Different algorithms were used to reconstruct the image of the phantom from the high-noise scan and the reconstructed image was compared with the reference image. In addition

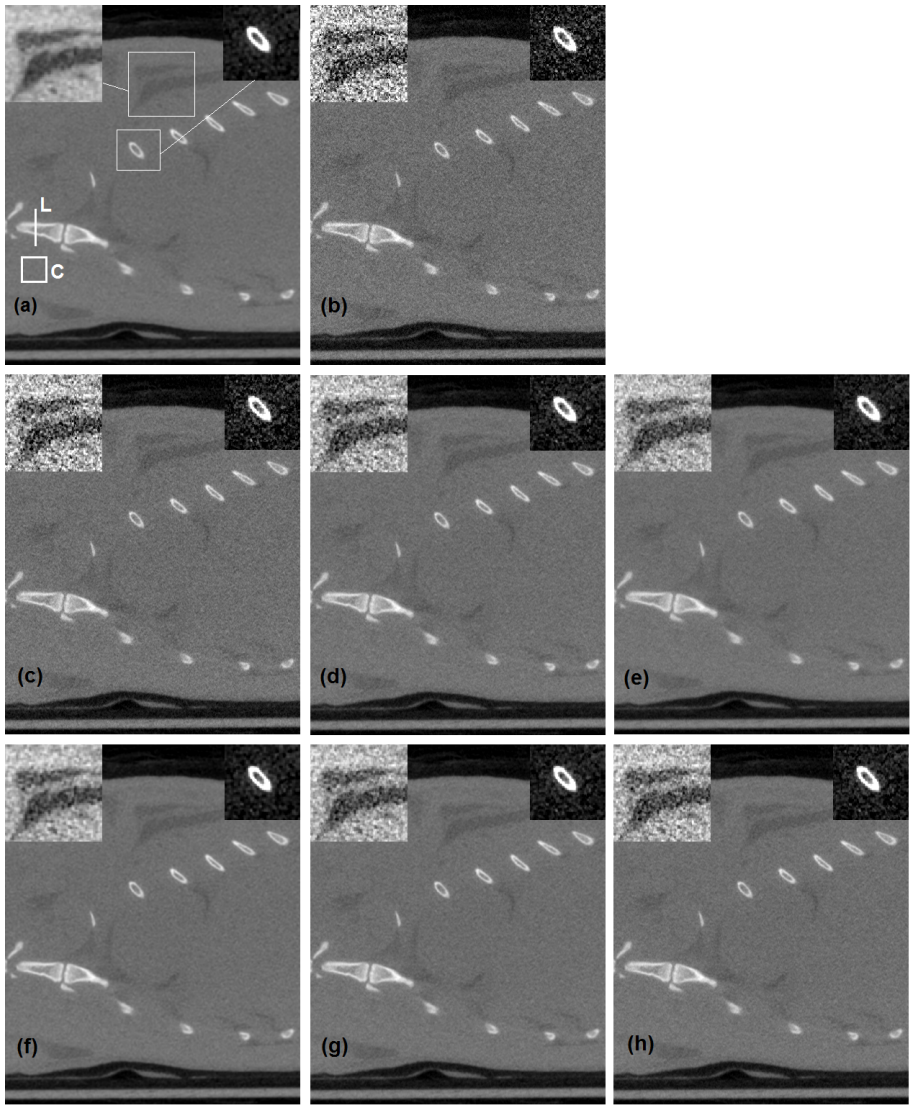

Figure 3.5: A slice of the image of the rat. (a) the reference image, (b) FDK-reconstructed from noisy projections, (c) bilateral filtering, (d) K-SVD, (e) BM4D-s, (f) the proposed algorithm, (g) BM4D-i, (h) ANLM.

to SSIM and RMSE, we computed two numbers as indicators of the noise strength (NS) and the spatial resolution (SR) as described below:

- The phantom has a uniform polycarbonate disk that has been included in the phantom for the purpose of estimating the noise level. We selected five 10^3-voxel cubes at different locations within the disk and computed the standard deviation of the voxel values in each cube. We use the average of these five standard deviations as an indicator of noise strength (NS).

- The phantom included a slanted edge that consisted of a plastic-air boundary that is specially designed for accurate estimation of the modulation transfer function (MTF). Although the MTF can also be estimated from a slit or a wire, this phantom provides an edge for MTF estimation because it is easier to fabricate a very smooth edge in a physical phantom. The slanted edge in this phantom had an angle of $5°$ relative to the image matrix that allowed for accurate estimation of MTF using methods such as that proposed in [39]. We estimate the MTF for the range of spatial frequencies between 0 and $5mm^{-1}$. As it is commonly done, we report the spatial frequency at which the normalized MTF reaches a value of 0.10 as a measure of spatial resolution (SR). We will also show the full MTF curves.

Figure 3.6 shows two fine coils inside the phantom in the images reconstructed by different algorithms. These coils are very useful for visual inspection of the spatial resolution and noise in the images. Compared with other sinogram denoising algorithms (i.e., bilateral filtering, K-SVD, and BM4D-s) and ANLM, the proposed algorithm seems to have resulted in a better image.

A more objective comparison of different algorithms can be performed using the quantitative criteria that we described above, which we have summarized in Table 3.5. These numbers clearly show that compared with the other sinogram denoising algorithms and ANLM, the proposed algorithm has resulted in a better image. BM4D-i has also performed well and is slightly better than the proposed algorithm but it takes 4 times longer to complete. In terms of computational time, the proposed algorithm is again much faster than the other algorithms, except bilateral filtering.

The values of spatial resolution (SR) shown in Table 3.5 represent the spatial frequency at which the normalized MTF reached a value of 0.10. However, this is not a complete characterization of the spatial resolution. A more detailed comparison of different algorithms can be done by examining

	RMSE	SSIM	time (h)	SR	NS
Bilateral filtering	0.0145	0.786	0.42	3.81	0.0148
KSVD	0.0142	0.789	2.5	3.78	0.0144
BM4D-s	0.0129	0.800	3.0	3.83	0.0139
Proposed algorithm	0.0127	0.820	0.9	3.83	0.0135
BM4D-i	0.0124	0.822	4.0	3.83	0.0132
ANLM	0.0132	0.797	5.4	3.81	0.0143

Table 3.5: Comparison between different algorithms on the scan of the physical phantom.

the full MTF curves. Moreover, as can also be seen in Table 3.5, the noise strengths are different for different algorithms. As we mentioned above, spatial resolution is influenced by the denoising strength. Therefore, we also estimated the MTF for different algorithms with noise matching. Specifically, we adjusted the denoising strengths of different algorithms such that the noise strength (NS, computed as described above) was approximately equal to 0.0130 for all algorithms. We then estimated the MTF for the images produced by different algorithms.

The estimated MTFs are shown in Figure 3.7. We have shown the estimated MTFs in part (a) of this figure. Because the number of algorithms compared is large and some of the MTF curves are very close, in part (b) of this figure we have shown the difference between the MTF of the image produced by different algorithms and the MTF of the image reconstructed from noisy projections with the FDK algorithm. We have done this only to be able to better see the difference between the MTFs for different algorithms. Overall, the plots of MTF in this figure show that the images produced by BM4D-s and the proposed algorithm have higher spatial resolution than other algorithms, especially in the frequency range $[1mm^{-1}, 2mm^{-1}]$.

3.4 Discussion

In summary, our results with simulated and real CT projections show that the proposed algorithm achieves state-of-the-art results. It outperforms or at least matches some of the best projection-domain and image-domain denoising algorithms in terms of the visual and objective quality of the produced image and computational time. Our simulation study with the digital brain phantom showed that the projections denoised with the proposed algorithm

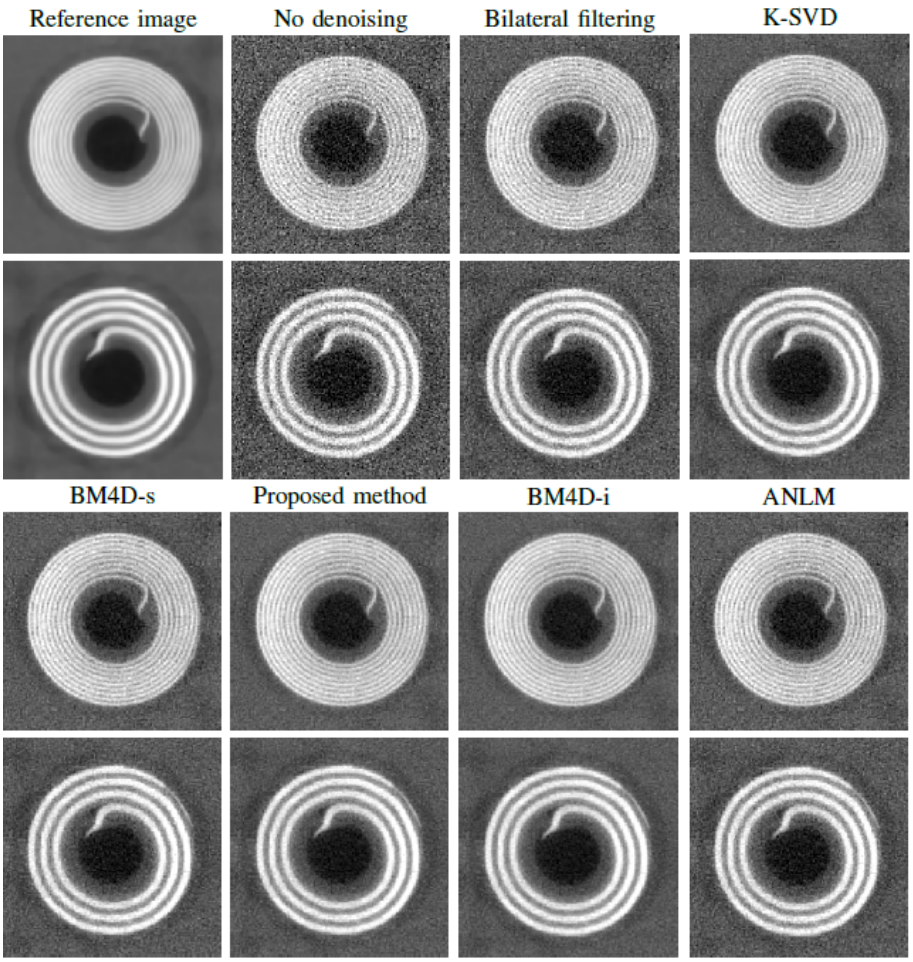

Figure 3.6: Images of two fine coils inside the physical phantom in the reference image and the images produced from the high-noise scan using different algorithms.

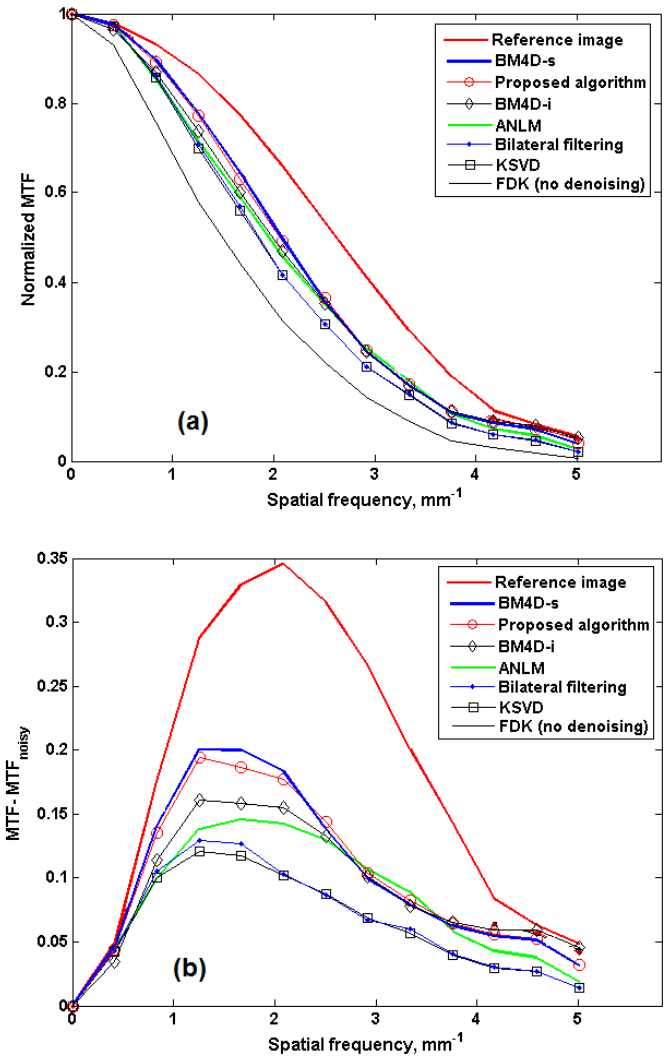

Figure 3.7: (a) The estimated noise-matched MTF for the images of the physical phantom produced by different algorithms, and (b) the difference between the MTF for different algorithms and the MTF for the FDK-reconstructed image without denoising.

were closer to the true projections, compared with projections denoised with the other sinogram-denoising algorithms. Our simulation experiments as well as our experiments with real data showed that in terms of the visual and objective quality of the produced image, the proposed algorithm achieved better results than the other methods. Denoising in the image domain with the BM4D algorithm often produced results that were close to the images produced by our proposed sinogram-denoising algorithm, but BM4D is much more computationally intensive than our proposed algorithm.

The proposed algorithm exploits the abundant self-similarity in the projections by grouping similar blocks extracted from the stacked projections. To denoise groups of similar blocks, the proposed algorithm makes the assumption that blocks in each such group should have a joint-sparse representation in a well-designed dictionary, which can be learned from training data. It is important to note that, at the level of small blocks of size 8^3 considered by the proposed algorithm, the type of patterns that appear in the stacked projections does not depend strongly on the object being imaged. This is important because it can mean that a dictionary learned from projections of a certain object can be used for denoising the projections of a different object. This can lead to large savings in computation because learning a dictionary is by far more computationally demanding than applying the dictionary for denoising. For natural images, in general, the dominant types of features in different images can be quite different and, for example, a dictionary learned on a cartoon-like or smooth image may not be optimal for denoising an image that contains fine textures. Therefore, for dictionary-based denoising of natural images, it is usually better to learn the dictionary from a set of similar training images or even from the noisy image itself. For CT projections, however, because the local nature of CT projections is not dependent on the scanned object, we expect that a dictionary learned from the projections of one object could be used for denoising the projections of another object.

In order to determine if this is in fact the case, we used the dictionary learned from the projections of the rat (from Section 3.3.2) to denoise the projections of the physical phantom (from Section 3.3.3). And vice versa, we used the dictionary learned on the projections of the physical phantom to denoise the projections of the rat. In both cases, the results obtained were almost exactly the same as those presented in the Results section above. In particular, none of the values in Tables 3.4 and 3.5 changed by more than 2%. In our opinion, this indicates that the dictionary learned from projections of one object can be used for denoising of the projections of an entirely different object. Of course, if the scan geometry changes, for

example if the cone angle is increased or decreased significantly, the learned dictionary may no longer be effective and a new dictionary must be learned. But these settings do not usually change on a commercial scanner.

As a further test of generalizability of the learned dictionaries in the proposed algorithm, we performed another experiment. In this experiment, the abdominal part of another rat was scanned. The rat in Section 3.3.2 was scanned in the chest region (as can be seen in Figure 3.5) and the tube voltage, tube current, and exposure time were equal to 70 kV, 32 mA, and 16 ms, respectively. In this new scan, the new rat was scanned in the abdominal region using a tube voltage of 50 kV. Furthermore, a $0.2\,mm$ copper filter was used to create a noisier scan than the scan in Section 3.3.2. We applied the proposed algorithm with the dictionary learned from the scan of the rat in Section 3.3.2 on this new scan. We also applied the other algorithms on this scan. A slice of the images produced by different algorithms from this scan has been shown in Figure 3.8. In the same figure, we have also shown a segment of a profile in this slice. The proposed algorithm has resulted in very effective denoising that is better than or comparable with the other methods. Our experience shows that the dictionary learned on the scan of one object can be used for effective denoising of the scan of another object, unless the scan settings such as angular spacing between successive projections changes drastically. This can mean large savings in computational time. As an example, denoising of the scan of the rat in Section 3.3.2 required approximately $1h$, as shown in Table 3.4, while learning the dictionary for the same scan takes longer than this. The dictionary learning time depends on the number of training signals, the initial dictionary used, and the dictionary size. With a good initialization, as we explained in Section 3.2.2, a dictionary can be trained in approximately $3 - 4h$.

The size of the dictionary has a direct effect on the computational demand and the denoising performance. A larger dictionary is usually more expressive and, in general, can lead to a sparser representation and higher performance. On the other hand, the computational cost of applying the dictionary is directly related to its size. As we mentioned above, in all of the experiments that we reported in this chapter we used a dictionary of size 1024, which means that the number of atoms was twice the signal dimensionality. A natural question is what is the effect of dictionary size in this application. In other words, one would like to know if the performance of the proposed algorithm can be improved by increasing the dictionary size, or whether the size of the dictionary may be reduced without a loss in performance.

In our experience, larger dictionaries do not lead to a significant im-

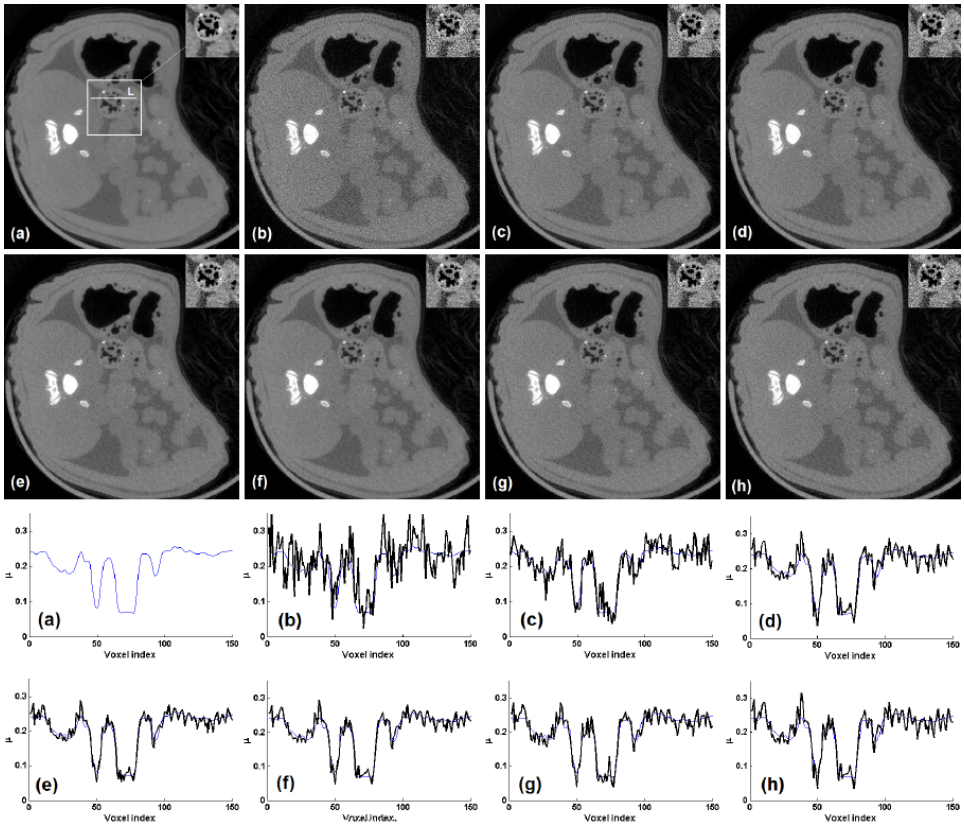

Figure 3.8: (Top) A slice in the image of the second rat produced by different algorithms, and (Bottom) a small profile segment through this slice. The location of the profile segment has been marked on the slice of the reference image with the line segment "L". (a) reference image, (b) FDK-reconstructed from noisy projections, (c) bilateral filtering, (d) KSVD, (e) BM4D-s, (f) the proposed algorithm, (g) BM4D-i, (h) ANLM.

provement in the denoising performance, while significantly increasing the computational load. Reducing the dictionary size, on the other hand, leads to a slight performance loss. As an example, in Figure 3.9 we show the effect of the size of the dictionary in our experiment with the rat data (Section 3.3.2). In this figure, we plot the RMSE, SSIM, and computational time for several different dictionary sizes. The actual values reported in Table 3.4 were for a dictionary of size 1024. In Figure 3.9 we plot the normalized values of RMSE, SSIM, and the computational time by dividing the values for various dictionary sizes by the values for a dictionary size of 1024. Note that a low RMSE and computational time and a high SSIM are desirable. From this figure, for dictionary sizes of 768 and 512 the quality of the reconstructed image is only slightly lower compared to a dictionary size of 1024. Increasing the dictionary size to 1536 and 2048 does not lead to a noticeable improvement in RMSE and SSIM, while significantly increasing the computational time.

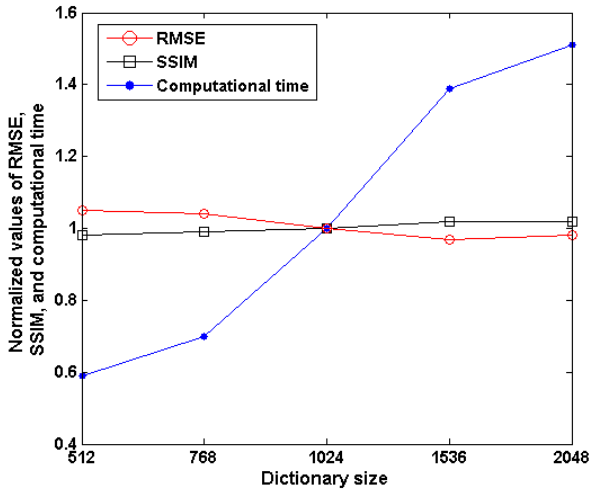

Figure 3.9: Effect of the dictionary size on the denoising performance and computational time for denoising of the projections of the rat (Section 3.3.2).

The observation that the dictionary size can be reduced to the signal dimensionality (i.e., 512) without a substantial degradation in the performance loss is surprising. In almost all image-processing applications of learned dictionaries, the dictionary size is at least twice the signal dimensionality. Figure 3.10(a) shows some typical atoms of the dictionary learned from the projections of the rat described in Section 3.3.2. Because the learned atoms

are actually $8 \times 8 \times 8$ cubes, we have randomly selected 50 of these atoms and have shown all of their 8 slices in Figure 3.10(a). In Figure 3.10(b) we have randomly selected 50 atoms learned on the reconstructed image of the rat. A visual comparison of these atoms from the two dictionaries quickly reveals their fundamental difference. Even though there exist similar-looking atoms in the two dictionaries, one can easily see at a glance that the atoms learned on the projections are simpler and smoother. This stems from the fact that the type of patterns that occur in CT projections are much more limited. This is also why we could limit the number of dictionary atoms to the signal dimensionality (512), while for natural images, including medical images, the number of atoms in the dictionary is usually chosen to be at least twice the signal dimensionality. Further reducing the dictionary size below 512 leads to a sharp deterioration of the performance of the proposed algorithm.

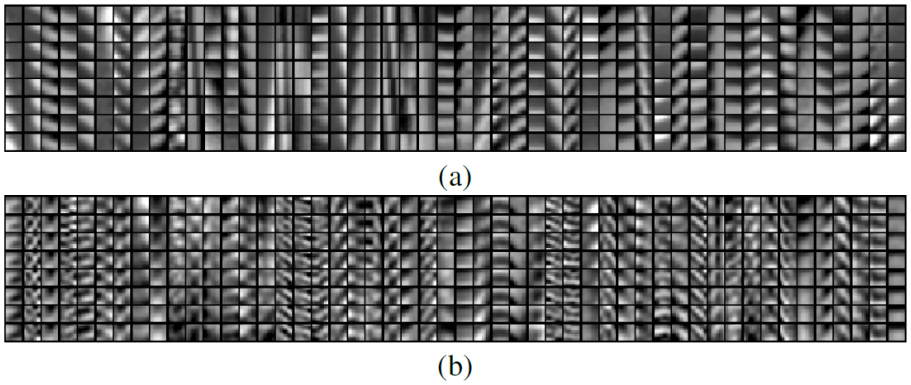

Figure 3.10: (a) A full depiction of 50 randomly selected atoms from the dictionary learned from the projections of the rat; each column shows the 8 slices of a single atom. (b) The same, for the dictionary learned from the reconstructed image of the rat.

Chapter 4

Sinogram Denoising using Total Variation

4.1 Introduction

In this chapter, we use a Poisson noise model for the projection measurements and propose denoising algorithms based on total variation (TV) regularization. Unlike our approach in Chapter 3, in this Chapter we will denoise each projection view separately.

We denote the true and noisy projections with u and v, respectively, and assume that they are of size $m \times n$. Individual pixels of a projection, for example for u, are denoted as $u(i,j),\ i = 1$ to $m,\ j = 1$ to n. For an arbitrary pixel location (from the probability mass function of a variable with Poisson distribution):

$$P(v(i,j)|u(i,j)) = \frac{e^{-u(i,j)}u(i,j)^{v(i,j)}}{v(i,j)!} \tag{4.1}$$

Assuming the pixel values are independent, for the entire image we will have:

$$P(v|u) = \prod_{i,j} \frac{e^{-u(i,j)}u(i,j)^{v(i,j)}}{v(i,j)!} \tag{4.2}$$

We ignore the denominator, which is independent of u. Since we want to find a functional to minimize, we consider the negative logarithm of the numerator:

$$-\log(P(v|u) \propto \sum_{i,j} u(i,j) - v(i,j)\log(u(i,j)) \tag{4.3}$$

With this measurement consistency term, total variation denoising can be performed by minimizing this cost function:

$$E_\lambda(u) = \int_\Omega (u - v\log u) + \lambda \int_\Omega |\nabla u| \tag{4.4}$$

where ∇u is the gradient of u. As we mentioned in Section 2.5, because this regularizer is based on the ℓ_1-norm of the gradient, it is very effective in preserving image edges while suppressing the noise. Therefore, it has been tremendously successful in reconstruction, deconvolution, and denoising of piecewise-constant images. In recent years, there has been a growing body of research on improving the capabilities of this model [48]. Perhaps the most significant enhancements have been achieved by including higher-order differentials in the model [47, 286].

The use of higher-order derivatives leads to superior results on images that contain piecewise-smooth features. On piecewise-smooth images, the basic TV formulation leads to artificial blocky features known as staircase artifacts. This is because with the basic TV model, piecewise-constant solutions are preferred. Including higher-order differentials, on the other hand, will encourage piecewise-smooth solutions. This can be very important for projection measurements in CT. Even if the imaged object (e.g., the human body) may be modeled as piecewise-constant, its projections will not be piecewise-constant. This is easy to visualize and we show a simple example in Figure 4.1. This figure shows a 2D slice and a 1D profile from the low-contrast 3D Shepp-Logan phantom alongside a typical cone-beam projection of it. Even though the the phantom itself is strictly piecewise constant, this is not the case for its projection. It is well documented that the basic TV model does not achieve optimal performance on this type of images [47, 199].

In this chapter, we propose two approaches for tackling this problem. The first approach is to use higher-order differentials. The second approach is to apply the basic TV denoising in a locally adaptive fashion.

4.2 Approach 1- Employing higher-order derivatives in TV regularization

4.2.1 The proposed algorithm

Following the above discussion, we suggest a regularization function that includes the ℓ_1-norm of both the gradient and the Hessian of the image, leading to a cost function of the form:

$$E(u) = \int_\Omega (u - v \log u) + \lambda_1 \int_\Omega |\nabla u| + \lambda_2 \int_\Omega |\nabla^2 u| \tag{4.5}$$

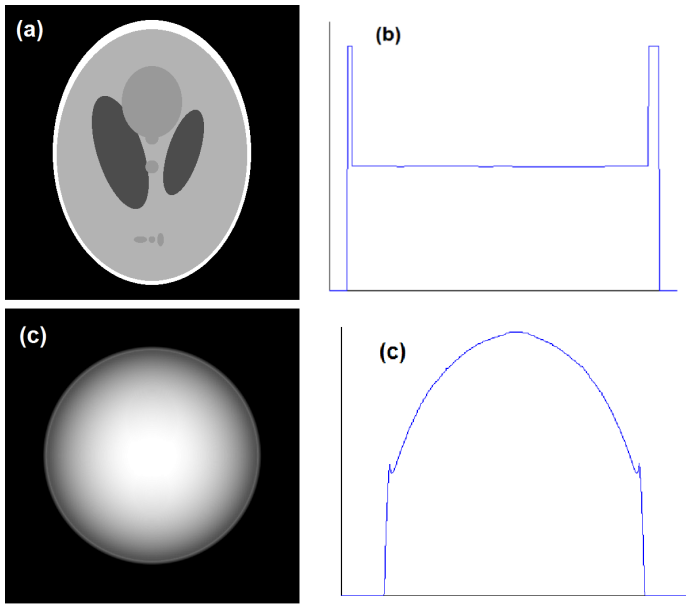

Figure 4.1: The central slice of the low-contrast 3D Shepp-Logan phantom (a), and a representative one-dimensional profile of it (b); a cone-beam projection of the same phantom (c), and a one-dimensional profile of the projection (d).

or in the discrete image domain:

$$\begin{aligned} E(u) = &\sum_{i,j}(u(i,j) - v(i,j)\log u(i,j)) \\ &+ \lambda_1 \sum_{i,j} |\nabla u(i,j)| + \lambda_2 \sum_{i,j} |\nabla^2 u(i,j)| \end{aligned} \tag{4.6}$$

The norms of the gradient and the Hessian in the discrete image domain are defined as follows:

$$|\nabla u(i,j)| = \left(D_x u(i,j)^2 + D_y u(i,j)^2\right)^{1/2} \tag{4.7}$$

$$|\nabla^2 u(i,j)| = \left(D_{xx} u(i,j)^2 + 2D_{xy} u(i,j)^2 + D_{yy} u(i,j)^2\right)^{1/2} \tag{4.8}$$

where D_x, D_y, D_{xx}, D_{xy}, and D_{yy} are the first and second-order difference

operators defined as follows:

$$\begin{aligned}
D_x u(i,j) &= u(i+1,j) - u(i,j) \\
D_y u(i,j) &= u(i,j+1) - u(i,j) \\
D_{xx} u(i,j) &= u(i+1,j) - 2u(i,j) + u(i-1,j) \\
D_{yy} u(i,j) &= u(i,j+1) - 2u(i,j) + u(i,j-1) \\
D_{xy} u(i,j) &= u(i+1,j+1) - u(i+1,j) - u(i,j+1) + u(i,j)
\end{aligned} \tag{4.9}$$

Here, we have provided the definitions for the interior pixels. For the boundary pixels we assume periodic boundary condition as in [248, 332]. Henceforth, we only work in the discrete domain but to simplify the expressions we drop the pixel indices and only show them when necessary.

To derive a minimization algorithm for the functional $E(u)$, we follow the split Bregman iterative framework [117] which is a very efficient algorithm for ℓ_1-regularized problems. Split Bregman method can be considered as a member of larger families of algorithms such as the alternating direction method of multipliers [33] or proximal methods [66]. In the split Bregman method, first the unconstrained optimization problem is converted into a constrained problem by introducing new variables. For $E(u)$ in (4.6), we write the corresponding constrained problem by introducing three new variables (f, g, and h):

$$\begin{aligned}
&\textbf{minimize } \sum (f - v\log f) + \lambda_1 \sum |g| + \lambda_2 \sum |h| \\
&\textbf{subject to } f = u,\ g = \nabla u,\ h = \nabla^2 u
\end{aligned} \tag{4.10}$$

This constrained problem can now be solved through the following Bregman iteration:

$$\begin{aligned}
&\textbf{Initialize: } u^0 = v,\ f^0 = v,\ g^0 = \nabla v,\ h^0 = \nabla^2 v,\ b_1^0 = b_2^0 = b_3^0 = 0 \\
&\textbf{while } \|u^k - u^{k-1}\|_2 > \epsilon \\
&\quad [u^{k+1}, f^{k+1}, g^{k+1}, h^{k+1}] = \underset{u,f,g,h}{\arg\min} \sum (f - v\log f) \\
&\qquad\qquad + \lambda_1 \sum |g| + \lambda_2 \sum |h| \\
&\qquad\qquad + \frac{\mu_1}{2} \sum (f - u - b_1^k)^2 \\
&\qquad\qquad + \frac{\mu_2}{2} \sum (g - \nabla u - b_2^k)^2 \\
&\qquad\qquad + \frac{\mu_3}{2} \sum (h - \nabla^2 u - b_3^k)^2
\end{aligned} \tag{4.11}$$

$$
\begin{aligned}
b_1^{k+1} &= b_1^k + u^{k+1} - f^{k+1} \\
b_2^{k+1} &= b_2^k + \nabla u^{k+1} - g^{k+1} \\
b_3^{k+1} &= b_3^k + \nabla^2 u^{k+1} - h^{k+1}
\end{aligned}
$$

end

where μ_i are the algorithm parameters and b_i are auxiliary variables. In the Bregman iterative approach, the updates of the auxiliary variables b_i replace the updates of the subgradients of the objective function. We should also note that if u is an m-by-n image, i.e., $u \in \mathbb{R}^{m\times n}$, then $f\,, b_1 \in \mathbb{R}^{m\times n}$, $\nabla u\,, g\,, b_2 \in (\mathbb{R}^{m\times n})^2$, and $\nabla^2 u\,, h\,, b_3 \in (\mathbb{R}^{m\times n})^4$. We denote the components of g and h as $g = [g_x, g_y]$ and $h = [h_{xx}, h_{xy}, h_{yx}, h_{yy}]$ and similarly for ∇u, $\nabla^2 u$, b_2, and b_3.

Although it may seem that the above modification has made the problem harder, the efficiency of the split Bregman scheme lies in the fact that the minimization problem can now be split into smaller problems that can be solved much more easily. Therefore, the large minimization problem in the above algorithm is solved by iteratively minimizing with respect to each of the four variables (u, f, g, and h), resulting in the following algorithm:

Initialize: $u^0 = v,\ f^0 = v,\ g^0 = \nabla v,\ h^0 = \nabla^2 v,\ b_1^0 = b_2^0 = b_3^0 = 0$

while $\|u^k - u^{k-1}\|_2 > \epsilon$

For $i = 1 : N$

$$
\begin{aligned}
f^{k+1} &= \arg\min_f \sum (f - v\log f) + \frac{\mu_1}{2}\sum (f - u^k - b_1^k)^2 \\
u^{k+1} &= \arg\min_u \frac{\mu_1}{2}\sum (f^{k+1} - u - b_1^k)^2 \\
&\qquad + \frac{\mu_2}{2}\sum (g^k - \nabla u - b_2^k)^2 \\
&\qquad + \frac{\mu_3}{2}\sum (h^k - \nabla^2 u - b_3^k)^2 \\
g^{k+1} &= \arg\min_g \lambda_1 \sum |g| + \frac{\mu_2}{2}\sum (g - \nabla u^{k+1} - b_2^k)^2 \\
h^{k+1} &= \arg\min_h \lambda_2 \sum |h| + \frac{\mu_3}{2}\sum (h - \nabla^2 u^{k+1} - b_3^k)^2
\end{aligned}
\tag{4.12}
$$

end

$b_1^{k+1} = b_1^k + u^{k+1} - f^{k+1}$

$$b_2^{k+1} = b_2^k + \nabla u^{k+1} - g^{k+1}$$
$$b_3^{k+1} = b_3^k + \nabla^2 u^{k+1} - h^{k+1}$$

end

To avoid complicating the notation, we have not introduced additional indices for the variable updates in the For loop. In fact, for many problems only one iteration of this loop is sufficient for fast convergence of the overall algorithm [117]. The efficiency of the split Bregman scheme entirely depends on how fast the sub-problems can be solved. In the following, we will show that for our problem, the four sub-problems can be solved very efficiently.

Minimization with respect to f: Returning to the notation with pixel indices, this sub-problem is:

$$\begin{aligned} f^{k+1} = \arg\min_f \sum_{i,j} (f(i,j) - v(i,j)\log f(i,j)) \\ + \frac{\mu_1}{2} \sum_{i,j} (f(i,j) - u^k(i,j) - b_1^k(i,j))^2 \end{aligned} \tag{4.13}$$

This expression can be written as a sum of scalar minimization problems in terms of individual pixel values. Since the function to be minimized is convex in terms of $f(i,j)$, the solution can be found by setting the derivative to zero. Using basic calculus and the knowledge that $f \geq 0$ we can show that the following formula gives the exact solution to this problem.

$$\begin{aligned} f^{k+1}(i,j) = & \frac{B}{2} + \sqrt{\left(\frac{B}{2}\right)^2 + \frac{v(i,j)}{\mu_1}} \\ & \text{where} \qquad B = u^k(i,j) + b_1^k(i,j) - \frac{1}{\mu_1} \end{aligned} \tag{4.14}$$

Minimization with respect to u: This subproblem can be solved through its optimality condition which can be written as [117, 248]:

$$\begin{aligned} \left[\mu_1 I - \mu_2(D_x^T D_x + D_y^T D_y) + \mu_3(D_{xx}^T D_{xx} + 2D_{xy}^T D_{xy} + D_{yy}^T D_{yy})\right] u^{k+1} = \\ \mu_1(f^{k+1} - b_1^k) - \mu_2(D_x^T(g_x^k - b_{2x}^k) + D_y^T(g_y^k - b_{2y}^k)) \\ + \mu_3(D_{xx}^T(h_{xx}^k - b_{3xx}^k) + 2D_{xy}^T(h_{xy}^k - b_{3xy}^k) + D_{yy}^T(h_{yy}^k - b_{3yy}^k)) \end{aligned}$$

where D_x^T, D_y^T, D_{xx}^T, D_{xy}^T, and D_{yy}^T are the backward-difference operators corresponding to the forward-difference operators defined in (4.9). For the pixels in the interior of the image, these operators are defined as:

$$\begin{aligned}
D_x^T u(i,j) &= u(i,j) - u(i-1,j) \\
D_y^T u(i,j) &= u(i,j) - u(i,j-1) \\
D_{xx}^T u(i,j) &= D_{xx} u(i,j) = u(i+1,j) - 2u(i,j) + u(i-1,j) \\
D_{yy}^T u(i,j) &= D_{yy} u(i,j) = u(i,j+1) - 2u(i,j) + u(i,j-1) \\
D_{xy}^T u(i,j) &= u(i,j) - u(i-1,j) - u(i,j-1) + u(i-1,j-1)
\end{aligned} \tag{4.15}$$

Despite its long expression, the above equation is a system of linear equations in u^{k+1}. Due to the structure of the forward and backward difference matrices, the system matrix is diagonally dominant. When the size of the image is not small, an efficient algorithm for finding a good approximate solution is the Gauss-Seidel method [118, 248].

Minimization with respect to g: This problem reads:

$$g^{k+1} = \arg\min_g \lambda_1 \sum_{i,j} |g(i,j)| + \frac{\mu_2}{2} \sum_{i,j} (g(i,j) - \nabla u^{k+1}(i,j) - b_2^k(i,j))^2$$

This problem is also equivalent to a set of scalar problems in terms of individual pixel values. Its solution is a simple extension of the soft thresholding operation [249, 328]:

$$\begin{aligned}
g_x^{k+1}(i,j) &= \max\left(|B(i,j)| - \frac{\lambda_1}{\mu_2}, 0\right) \frac{B_x(i,j)}{|B(i,j)|} \\
g_y^{k+1}(i,j) &= \max\left(|B(i,j)| - \frac{\lambda_1}{\mu_2}, 0\right) \frac{B_y(i,j)}{|B(i,j)|} \\
\text{where } B(i,j) &= [B_x(i,j), B_y(i,j)] \\
&= [b_{2x}^k(i,j) + D_x u^{k+1}(i,j), b_{2y}^k(i,j) + D_y u^{k+1}(i,j)]
\end{aligned} \tag{4.16}$$

Minimization with respect to h: This problem is very similar to minimization with respect to g above. Its solution is similarly a generalization

of the soft thresholding operation [249]:

$$
\begin{aligned}
h_{xx}^{k+1}(i,j) &= \max\left(|C(i,j)| - \frac{\lambda_2}{\mu_3}, 0\right)\frac{C_{xx}(i,j)}{|C(i,j)|} \\
h_{xy}^{k+1}(i,j) &= \max\left(|C(i,j)| - \frac{\lambda_2}{\mu_3}, 0\right)\frac{C_{xy}(i,j)}{|C(i,j)|} \\
h_{yx}^{k+1}(i,j) &= \max\left(|C(i,j)| - \frac{\lambda_2}{\mu_3}, 0\right)\frac{C_{yx}(i,j)}{|C(i,j)|} \\
h_{yy}^{k+1}(i,j) &= \max\left(|C(i,j)| - \frac{\lambda_2}{\mu_3}, 0\right)\frac{C_{yy}(i,j)}{|C(i,j)|} \\
\text{where } C(i,j) &= [C_{xx}(i,j), C_{xy}(i,j), C_{yx}(i,j), C_{yy}(i,j)] \\
&= [b_{3xx}^k(i,j) + D_{xx}^{k+1}u(i,j), b_{3xy}^k(i,j) + D_{xy}^{k+1}u(i,j), \\
&\quad b_{3yx}^k(i,j) + D_{yx}^{k+1}u(i,j), b_{3yy}^k(i,j) + D_{yy}^{k+1}u(i,j)]
\end{aligned}
\tag{4.17}
$$

If we set $\lambda_2 = 0$ in Equations (4.5) or (4.6), we get a simplified model with standard (i.e., first-order) TV regularization. This will greatly simplify the algorithm because the variables h and b_3 will also be removed. We will refer to this simplified model as "TV sinogram denoising" and to the full model described above as "(TV + TV^2) sinogram denoising" and will present the results for both models. This will allow us to see whether or not, and to what extent, the more complex model with two regularizers improves the results.

To evaluate the proposed denoising algorithm, we applied it on sets of simulated and real cone-beam sinograms. We compare our proposed algorithm with the bilateral filtering algorithm that we described in Section 3.3.

4.2.2 Simulation experiment

Noisy cone-beam projections were simulated from a 3D low-contrast Shepp-Logan phantom according to the model in Equation (1.1). We used two different values of $N_0^i = 1000$ and $N_0^i = 100$ to simulate two sets of projections with different levels of noise. For each ray, the expected number of detected photons (N_d^i) is given by Equation (1.1). The actual detected photon count was simulated as a Poisson random variable with mean equal to N_d^i. We will refer to the scans with $N_0^i = 1000$ and $N_0^i = 100$ as low-noise and high-noise, respectively. The phantom size was 256^3 voxels and the projections were each 300×300 pixels in size. Each simulated scan consisted of 720 projections between 0° and 360°.

Because for this simulated experiment we have access to the true projections, we can quantitatively compare the denoised projections with the true projections. To this end, we computed the value of two criteria: (1) the root-mean-square of the error (RMSE), where the error is defined as the difference between the denoised projection and the true projection (i.e., without the Poisson noise), and (2) the mutual information (MI) between the denoised projections ($\hat{u}$) and the true projections (u^*) computed as [259]:

$$\mathrm{MI}(u^*, \hat{u}) = \sum_{i=1}^{h} \sum_{j=1}^{h} q(u_i^*, \hat{u}_j) \log \left(\frac{q(u_i^*, \hat{u}_j)}{p(u_i^*) p(\hat{u}_j)} \right) \tag{4.18}$$

Here, p and q represent the marginal and joint probability distribution functions, respectively. We used histograms of u^* and $\hat{u}$ for estimating these probability densities and h is the number of bins in the histograms. We normalized the computed $\mathrm{MI}(u^*, \hat{u})$ by dividing it by $\mathrm{MI}(u^*, u^*)$.

The results of this comparison are presented in Table 4.1. As we mentioned above, "TV sinogram denoising" means setting $\lambda_2 = 0$ and adjusting λ_1 in our model. We will discuss the role of the parameter values and some possible approaches to selecting proper values later in this chapter. The results shown for TV sinogram denoising in Table 4.1 were obtained with $\lambda_1 = 2$, $\mu_1 = 10$, and $\mu_2 = 0.1$. The results shown for (TV + TV^2) sinogram denoising in Table 4.1 were obtained with $\lambda_1 = 2$, $\lambda_2 = 0.1$, $\mu_1 = 10$, $\mu_2 = 0.5$, and $\mu_3 = 0.01$. For bilateral filtering, the results are shown for the choice of the bandwidth of P_2 being equal to 2.2, which gave us the lowest RMSE. The numbers in this table indicate that the proposed TV sinogram denoising methods, especially the (TV + TV^2) sinogram denoising algorithm, outperform the method based on bilateral filtering.

		Bilateral filtering	TV denoising	(TV + TV^2) denoising
$N_0 = 100$	RMSE	0.0420	0.0364	0.0308
	MI	0.277	0.351	0.375
$N_0 = 1000$	RMSE	0.0192	0.0174	0.0163
	MI	0.408	0.425	0.442

Table 4.1: Comparison of different denoising algorithms in terms of the RMSE and MI of the denoised projections on the data simulated from the low-contrast Shepp-Logan phantom.

In order to determine the effect of sinogram denoising on the quality of the reconstructed image, we used the FDK algorithm to reconstruct the image of the phantom from noisy and denoised projections. In Figure 4.2 we have shown the central slices of the reconstructed images. This figure clearly shows that the image reconstructed from the projections denoised with the (TV + TV^2)-model has a better quality than the bilateral filtering and TV denoising, especially for the high-noise case.

For a quantitative comparison, we computed the RMSE, where error is defined as the difference between the reconstructed image and the true phantom image, and the structural similarity index (SSIM) between the two images as given in Equation (3.6). We have summarized the results of this quantitative comparison in Table 4.2. The numbers in this table indicate that denoising of the projections using the (TV + TV^2)-model results in a better image. The difference is more significant for the high-noise case.

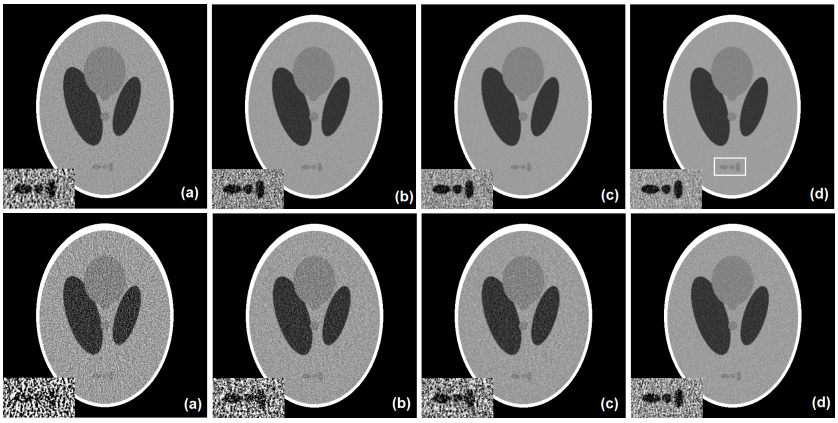

Figure 4.2: The central slice of the low-contrast Shepp-Logan phantom reconstructed from (a) noisy projections, and from projections denoised using (b) bilateral filtering, (c) TV sinogram denoising, and (d) (TV + TV^2) sinogram denoising. The top row is for reconstruction from the low-noise projections and the bottom row is for reconstruction from the high-noise projections. The location of the ROI that has been displayed on the bottom left of each slice has been marked by a rectangle in part (d) of this figure.

4.2.3 Experiments with real micro-CT data

The scanner described in Section 3.3.2 was used to scan the physical phantom. The phantom was scanned twice. The first scan consisted of 720

		No denoising	Bilateral filtering	TV denoising	(TV + TV^2) denoising
$N_0 = 100$	RMSE	0.360	0.148	0.124	0.111
	SSIM	0.220	0.396	0.451	0.480
$N_0 = 1000$	RMSE	0.210	0.091	0.083	0.077
	SSIM	0.404	0.634	0.672	0.690

Table 4.2: RMSE and SSIM for the images of the low-contrast Shepp-Logan phantom reconstructed from noisy projections and from projections denoised with different denoising algorithms.

projections between 0° and 360° at 0.5° intervals. For this scan, the tube voltage, tube current, and exposure time were chosen to be, respectively, 70 kV, 40 mA, and 25 ms. We used the full set of 720 projections from this scan to reconstruct a high-quality image. This image was created by first using the FDK algorithm to reconstruct an initial image and then applying 50 iterations of the MFISTA algorithm [17] to further improve its quality. We will refer to this image as "the reference image" and will use it as the ground-truth for evaluating the denoising algorithms. The second scan of the phantom consisted of 360 projections between 0° and 360° at 1° intervals. For this scan, the tube voltage, tube current, and exposure time were chosen to be 50 kV, 32 mA, and 16 mAs, respectively. This was the lowest possible setting in terms of mAs as the scanner did not operate under 0.5 mAs. Moreover, a 0.1-mm copper filter was used to further reduce the radiation dose, further increasing the noise level. The resulting scan was very noisy. Since we do not have the true projections, in this experiment we evaluate the denoising algorithms in the image domain.

In order to evaluate the quality of the reconstructed images, we compared them with the reference image by computing the RMSE and SSIM. The results are summarized in Table 4.3. In addition, we used two of the modules in the phantom to evaluate the spatial resolution and the noise level in the reconstructed images. We used the plastic-air edge for estimating the modulation transfer function (MTF). We used the method proposed in [39] to estimate the MTF over the range of spatial frequencies $[0, 5\,mm^{-1}]$. We report the spatial frequency at which the normalized MTF reached a value of 0.1 as the indicator of the spatial resolution. We use the standard deviation of the voxel values in a uniform polycarbonate disk in the phantom as an indicator of noise level. The values in Table 4.3 clearly show that the pro-

posed TV denoising model is better than bilateral filtering. Moreover, the full (TV + TV^2) model has led to a higher image quality than the simpler TV-denoising model.

	No denoising	Bilateral filtering	TV denoising	(TV + TV^2) denoising
RMSE	0.0393	0.0255	0.0228	0.0210
SSIM	0.460	0.646	0.696	0.707
Spatial resolution	3.55	3.58	3.61	3.61
Noise level	0.0383	0.0191	0.0172	0.0160

Table 4.3: Comparison of different sinogram denoising algorithms in terms of the quality of the reconstructed image of the physical phantom.

For visual comparison, in Figure 4.3 we have shown a slice of the phantom reconstructed from noisy and denoised projections. This figure agrees with quantitative comparison in Table 4.3. Our proposed TV-based denoising seems to have resulted in a higher quality image than bilateral filtering. Moreover, the (TV + TV^2)-model has produced a slightly better image than the TV-model. To show the difference between the images reconstructed from projections denoised with different algorithms more clearly, in Figure 4.4 we have shown a profile in the reconstructed images. The location of this profile has been marked with a white vertical line on the slice of the reference image in Figure 4.3(a). From the profiles in Figure 4.4, the image reconstructed from the projections denoised using the (TV+TV^2)-model is much closer to the reference image, which agrees with the quantitative evaluations presented in Table 4.3.

The values of the image quality metrics presented in Table 4.3 and the images shown in Figures 4.3 and 4.4 were obtained using one particular set of parameter values for each algorithm. However, they do not present a complete comparison of the performance of different denoising algorithms. Comparing different denoising algorithms requires a more detailed look at the trade-off between noise and spatial resolution. In Figure 4.5 we have shown plots of the noise level versus spatial resolution for a range of parameter values for different algorithms. Noise level and spatial resolution were computed as described above using the slanted edge module and the uniform disk module in the phantom. For bilateral filtering, we have shown the plot for the bandwidth of P_2 in the range $[0.7, 2.8]$. For the proposed TV-based algorithm we have presented three curves, each for one different

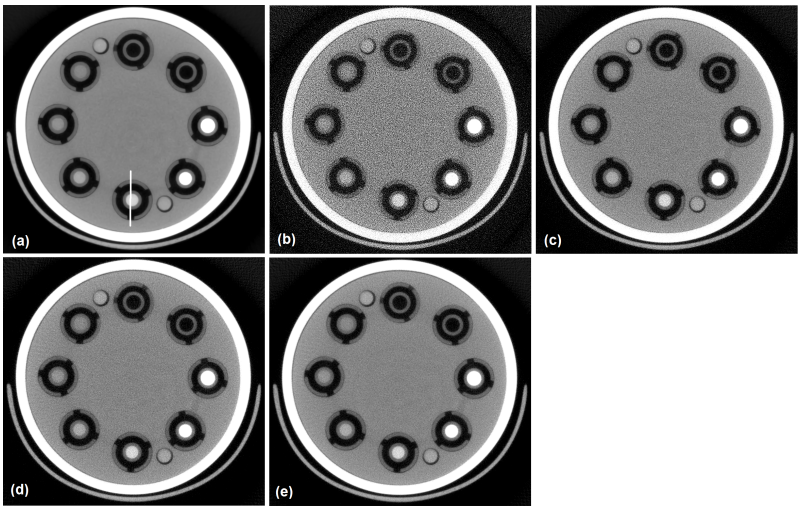

Figure 4.3: A slice of the image of the physical phantom reconstructed from noisy and denoised projections. (a) the reference image, (b) reconstructed from noisy projections, and reconstructed from projections denoised using (c) bilateral filtering, (d) the TV-model, and (e) the (TV+TV^2)-model.

value of the regularization parameter $\lambda_2 \in \{0, 0.2, 1\}$. The curve for $\lambda_2 = 0$ corresponds to the "TV-denoising", i.e. regularization only in terms of the gradient. For each value of λ_2, we applied the proposed algorithm for ten values of λ_1 in the range $[0.5, 10]$. Note that a high spatial resolution and a low noise level are desirable.

From this figure, it is clear that the proposed TV-based denoising algorithm outperforms the bilateral filtering. For the range of values of λ_1 and λ_2 that we tried in this experiment, the value of λ_1 seems to more strongly affect the behavior of the proposed algorithm. The Hessian regularization has a very positive effect. When $\lambda_2 > 0$, i.e., when the Hessian regularization term exists, the proposed algorithm can achieve better results in terms of noise level and spatial resolution, i.e., lower noise level and higher spatial resolution. Moreover, when $\lambda_2 > 0$, the performance is more stable with regard to changes in λ_1. This can be seen by comparing the three curves that have been shown for the TV-based algorithm. The curve corresponding to $\lambda_2 = 0$ is influenced more strongly by the change in λ_1, whereas when $\lambda_2 > 0$ the algorithm is less sensitive to change in λ_1.

For a complete characterization of the spatial resolution of the images reconstructed from the projections denoised by different algorithms, we plot

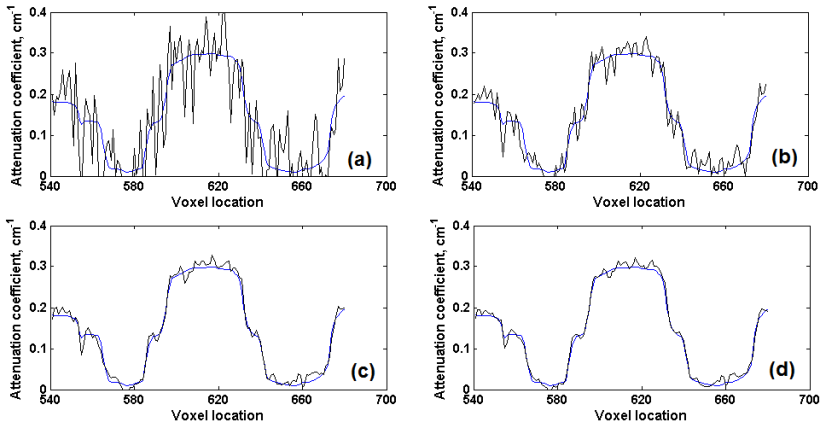

Figure 4.4: A profile in the reconstructed images of the physical phantom reconstructed from (a) noisy projections, and from projections denoised with (b) bilateral filtering, (c) the TV-model, and (d) the (TV+TV^2)-model. The blue line in each figure shows the profile in the reference image. The location of this profiles has been marked in Figure 4.3(a).

the estimated MTFs in Figure 4.6. Our approach to estimating the MTFs that are shown in this figure are different from the spatial resolutions shown in Table 4.3 in an important way. As can be seen in Table 4.3, the three algorithms are different in terms of both spatial resolution and noise level. However, as we mentioned above, noise and spatial resolution are interdependent. Therefore, estimation of the MTF curves in Figure 4.6 was done while matching the noise level of the reconstructed image for different algorithms. We adjusted the tuning parameters of the different algorithms (i.e., σ_i for bilateral filtering and λ_1 and λ_2 for the proposed algorithm) such that the noise level was the same in the reconstructed images for all three algorithms. We then estimated the MTF for different algorithms. From the estimated MTFs in Figure 4.6 it can be seen that the TV-based denoising leads to a higher MTF for all spatial frequencies, especially for spatial frequencies above $2mm^{-1}$. All three algorithms are very close up to the spatial frequency of approximately $1mm^{-1}$. The MTF for the two TV-based algorithms are close, but the (TV + TV^2)-model results in a sightly higher MTF at higher spatial frequencies.

The same scanner was used to scan two dead rats. The first scan had relatively less noise, whereas the second scan was much noisier. We will refer to these two scans as low-noise and high-noise rat scans. Each of the

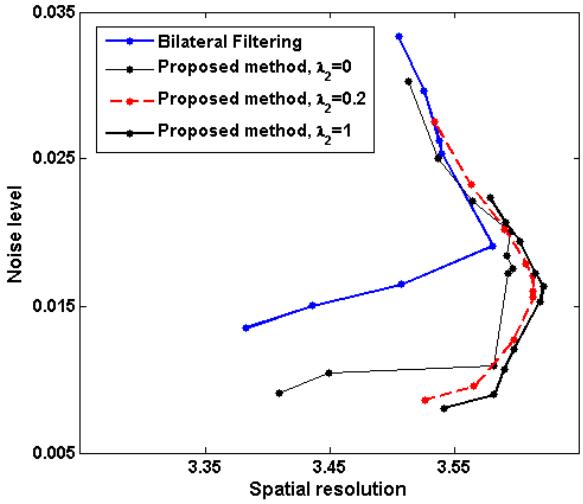

Figure 4.5: Plots of noise level versus spatial resolution for a range of parameter values for bilateral filtering and the proposed TV-based denoising algorithm.

two scans consisted of 720 projections between 0° and 360° at 0.5° intervals. For both scans, the tube voltage, tube current, and exposure time were set to 50 kV, 32 mA, and 16 ms, respectively. However, for the high-noise scan we used a 0.2-mm copper filter. For both the low-noise and the high-noise rat scans, we used all 720 projections to reconstruct a high-quality reference image using the same procedure as that described for the physical phantom above. We will use this image as the ground-truth for evaluating the denoising algorithms. For both the low-noise and the high-noise scans, we applied the denoising algorithms on a subset of 360 projections of the same scan and reconstructed the image of the rat using the FDK algorithm.

Table 4.4 shows a summary of the quantitative comparison between different sinogram denoising algorithms in terms of RMSE and SSIM of the reconstructed images of the low-noise rat scan. Furthermore, for a visual comparison, we have shown a typical slice of the reconstructed images in Figure 4.7. Similar to the above experiments, compared with bilateral filtering, the TV-based algorithms have produced better results.

In Figure 4.8, we have shown a slice from the reconstructed images of the high-noise rat scan. All three sinogram denoising algorithms have resulted in a significant improvement in the visual quality of the reconstructed image from this high-noise scan. Similar to the above experiments, the proposed

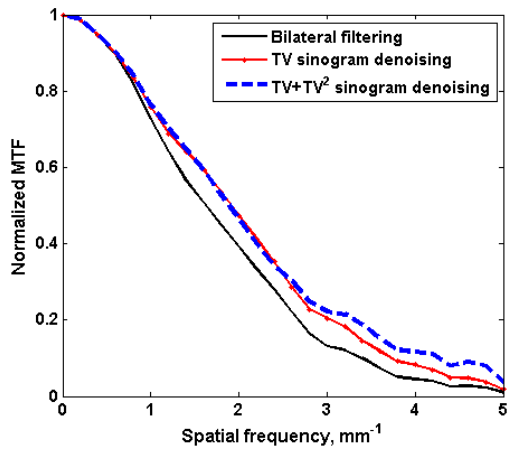

Figure 4.6: The estimated noise-matched MTF for different sinogram denoising algorithms.

	No denoising	Bilateral filtering	TV denoising	(TV + TV^2) denoising
RMSE	0.0220	0.0173	0.0140	0.0126
SSIM	0.622	0.685	0.711	0.740

Table 4.4: Comparison of different sinogram denoising algorithms in terms of the quality of the reconstructed image of the low-noise rat scan.

TV-based algorithm has worked better than bilateral filtering. Moreover, the (TV + TV^2) model seems to have more effectively suppressed the noise without blurring the image features.

However, the images shown in Figure 4.8 have been reconstructed using a particular set of parameters for each algorithm. For a better comparison between different denoising algorithms, we examine the trade-off between the noise suppression and image sharpness. For this purpose, as a measure of the noise level we computed the contrast-to-noise-ratio (CNR). We selected the cubes b and s as shown on the slice of the reference image in Figure 4.8(a). We estimated the CNR using the following equation:

$$\mathrm{CNR} = \frac{|\mu_\mathrm{s} - \mu_\mathrm{b}|}{(\sigma_\mathrm{s} + \sigma_\mathrm{b})/2} \tag{4.19}$$

where μ and σ denote, respectively, the mean and the standard deviation

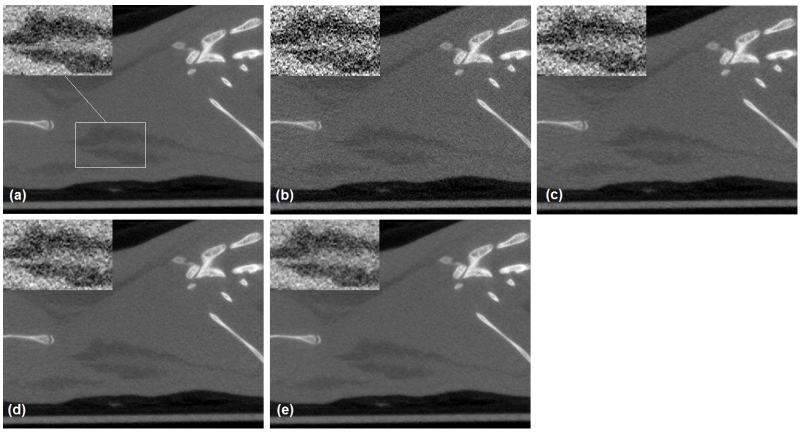

Figure 4.7: A slice of the images reconstructed from the low-noise rat scan; (a) the reference image, (b) reconstructed from noisy projections, and reconstructed from projections denoised with (c) bilateral filtering, (d) the TV model, and (e) the (TV + TV^2)-model.

of the voxels in each cube. As a measure of image sharpness, we computed the maximum slope along the line "L" shown in Figure 4.8(a). This line lies on an edge between soft tissue and fat. If the spatial resolution is high, the slope of this edge will also be high. On the other hand, if the image is over-smoothed this slope becomes small. We will denote the computed slope along the line "L" with SL and use it as a measure of image sharpness. A similar approach was used to quantify the spatial resolution in [219]. Therefore, a small SL indicates that the image is over-smoothed and spatial resolution is low. On the other hand, a large value of SL indicates that the image is sharp and spatial resolution is high.

Figure 4.9 shows the plots of SL versus CNR for different parameter values for bilateral filtering and TV-based denoising algorithms. Note that a high CNR and a high SL are desirable. For bilateral filtering, we changed the bandwidth of P_2 in the range $[0.7, 2.8]$. For the proposed TV-based algorithm, we found that parameter values $\lambda_1 = 4$ and $\lambda_2 = 0.4$ lead to good results. Therefore, in order to investigate the role of these two parameters, we first kept $\lambda_2 = 0.4$ constant and changed λ_1 in the range $[0, 10]$. Then, we kept $\lambda_1 = 4$ constant and changed λ_2 in the range $[0, 2.0]$. Therefore, for the proposed TV-based algorithm we have two curves that show the effect of tuning λ_1 and λ_2. It is clear from the proposed cost function in Equation (4.5) that λ_1 and λ_2 determine the strength of regularization in terms of the

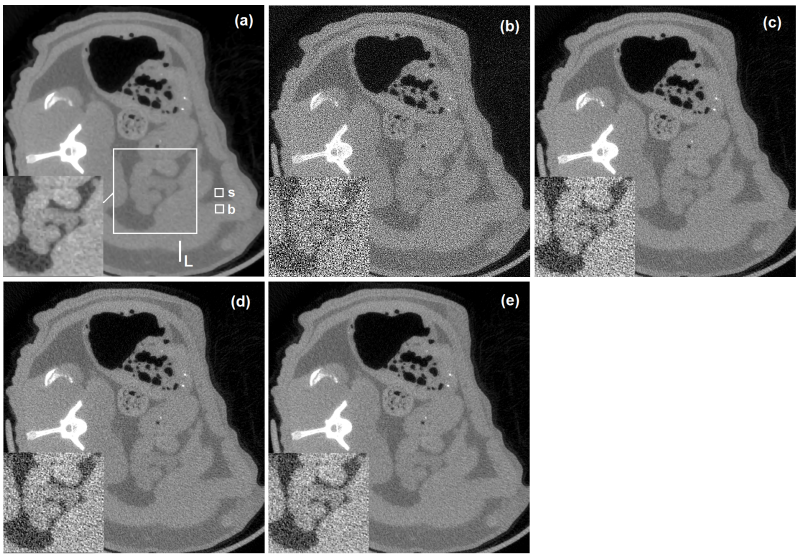

Figure 4.8: A slice of the images reconstructed from the high-noise rat scan; (a) the reference image, (b) reconstructed from noisy projections, and reconstructed from projections denoised with (c) bilateral filtering, (d) the TV model, (e) the (TV + TV^2)-model.

gradient and the Hessian, respectively. The plots in Figure 4.9 show that the proposed TV-based denoising algorithm can achieve higher SL and CNR compared with bilateral filtering. It is also clear that both regularization terms play an important role in the performance of the proposed algorithm because when either λ_1 or λ_2 decrease to zero, one or both of the criteria decrease. Especially, when $\lambda_2 \to 0$, both SL and CNR decrease. This clearly indicates the importance of the regularization in terms of the Hessian.

4.2.4 Discussion

Overall, the results of our experiments with simulated and real data show that the proposed algorithm is highly effective in suppressing the noise in the projection measurements in cone-beam CT. This is evident from the quantitative evaluation presented in Table 4.1. The effect of the noise suppression on the quality of the FDK-reconstructed images is substantial, as can be seen from Figures 4.2 - 4.9 and from the objective image quality metrics in Tables 4.2 - 4.4. Our results suggest that the proposed TV-based algorithm is superior to the method based on bilateral filtering both in terms of spatial

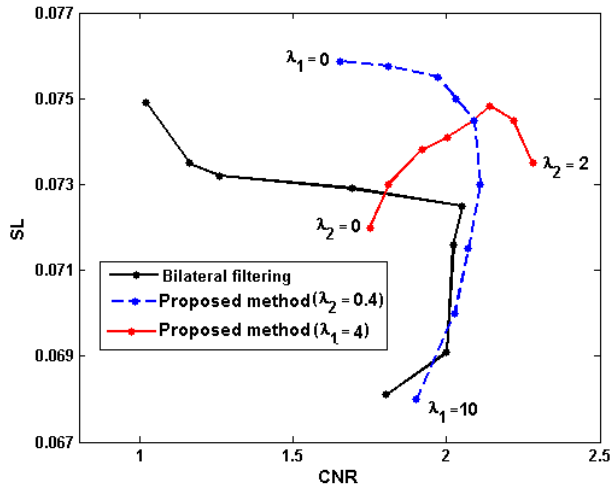

Figure 4.9: The computed SL (a measure of image sharpness or spatial resolution) versus CNR for a range of parameter values for different sinogram denoising algorithms.

resolution and noise suppression. The difference between the proposed TV sinogram denoising and the (TV + TV^2) sinogram denoising models was also significant on both the simulated data and on the real data. This shows a clear gain in the denoising performance by including the regularization in terms of the Hessian. We should note, however, that this gain in performance comes at the cost of more computation. For example, with our Matlab implementation on a Windows 7 PC with a 3.4 GHz Intel Core i7 CPU, denoising of one projection from the rat scan takes approximately 33 seconds with the TV model and 46 seconds with the (TV + TV^2) model.

The choice of algorithm parameters can significantly influence its performance, as shown in our results in Figures 4.5 and 4.9. There are two sets of parameters in the proposed algorithm. The regularization parameters, λ_1 and λ_2, control the degree of regularization. They must be selected according to the desired quality of the final image. The Bregman parameters, μ_1, μ_2, and μ_3 mostly influence the convergence of the algorithm [117, 248]. The simplified model with only first-order TV regularization has three parameters, λ_1, μ_1, and μ_2, which makes it easier to find good parameter values. For the full model, too, one can first ignore the second-order differential terms and find proper values for λ_1, μ_1, and μ_2, and then proceed to find good values for λ_2 and μ_3.

A well-known method for selecting λ_1 was proposed by Chambolle [46]. This method was proposed for the case of additive Gaussian noise. In our experience this method works for the case of Poisson noise as well. For this method to work, one needs to have a prior estimate of the noise level in the image. If the average noise variance in the noisy image v is σ^2, one starts with an arbitrary value of the regularizarion parameter $\lambda = \lambda^0$ and updates λ as follows:

Repeat until convergence

$$u^k = \arg\min_u \Psi(u, v) + \lambda^k \int_\Omega |\nabla u|$$

$$\lambda^{k+1} = \lambda^k \frac{(N\sigma)}{\Psi(u^k, v)}$$

end

The same strategy has been suggested for finding λ_1 for Poisson denoising in [112]. That paper has also developed an empirical equation relating the regularization parameter λ_1 and the Poisson noise variance, which has the form: $\lambda_1 = 1/(72.4/\sigma + 97.7/\sigma^2)$, where σ^2 is the noise variance. This equation can be very useful for choosing a good initial value for λ_1, which can then be improved using Chambolle's method mentioned above. For the simplified model, we found that choosing $\mu_1 \approx 10\,\lambda_1$ and $\mu_2 \approx 0.1\,\lambda_1$ lead to good results. For the full model, we had good results with $\lambda_2 < \lambda_1$, usually $\lambda_2 \approx \lambda_1/10$, and $\mu_3 \approx \lambda_2/10$, which are not very different from the values suggested for the inpainting problem in [249].

4.3 Approach 2- Locally adaptive regularization

4.3.1 The proposed algorithm

This approach is based on a comparison between the optimality conditions for the TV denoising models for Gaussian and Poisson noises [175]:

$$\begin{cases} \text{Gaussian} & E_\lambda(u) = \frac{1}{2}\|u - v\|_2^2 + \lambda \int_\Omega |\nabla u| \qquad (u - v) + \lambda\, p = 0 \\ \text{Poisson} & E_\lambda(u) = \int_\Omega (u - v \log u) + \lambda \int_\Omega |\nabla u| \quad (u - v) + (\lambda u)\, p = 0 \end{cases}$$

where p is a sub-gradient of $\int_\Omega |\nabla u|$. The only difference between the two equations is the dependence of the regularization parameter on u in the Poisson case. This suggests that a stronger smoothing must be applied where the signal has larger values. This outcome agrees with what we expect

since under the Poisson distribution the noise variance is proportional to the signal intensity.

Minimization of the cost function in (4.4) is not a straightforward problem. One approach is to first replace $|\nabla u|$ with $\sqrt{|\nabla u|^2 + \epsilon}$ for a small $\epsilon > 0$ and then to apply a gradient descent iteration [175]. Another approach, suggested in [282], is to use a Taylor's expansion of the data fidelity term and to minimize this approximate model. Here, we use an algorithm that was developed to solve the original ROF denoising problem for Gaussian noise. However, we denoise each sinogram pixel separately by minimizing $E_\lambda(u)$ in a small neighborhood around that pixel, and with a regularization parameter inspired by the optimality condition described above.

As we mentioned in the previous section, a heavily researched approach to reducing the staircase effect is to replace the ℓ_1 norm of the gradient with the ℓ_1 norm of higher-order differential operators. A less sophisticated approach, but one that has a trivial implementation, is to perform the total variation minimization locally. This approach has also been shown to alleviate the staircase effect [195]. Moreover, with a local minimization strategy, if the size of the neighborhood considered in minimization is small enough, one can safely assume that the sinogram intensity and noise level are approximately constant. Therefore, a solution based on the ROF's original model will be a good approximation to the solution of the model based on Poisson noise. This way, we can utilize efficient existing algorithms for the ROF model while avoiding the staircase artifacts.

Since our approach is based on Chambolle's famous algorithm [46], we briefly describe this algorithm here. This algorithm minimizes the following cost function, which is the same as the TV denoising model for Gaussian noise that we described in Section 2.5.

$$E_\lambda(u) = \frac{1}{2}\|u - v\|_2^2 + \lambda \int_\Omega |\nabla u| \tag{4.20}$$

If we denote by X and Y the space of the image u and its gradient, ∇u, respectively, then an alternative definition of total variation of u is:

$$\sum_{i,j} |\nabla u|_{i,j} = \sup\{\langle p, \nabla u\rangle_Y : p \in Y, |p_{i,j}| \leq 1\} \tag{4.21}$$

Chambolle introduced the discrete divergence operator as the dual of the gradient operator, i.e. $\langle p, \nabla u\rangle_Y = \langle -\mathrm{div}\, p, u\rangle_X$. In the discrete image domain:

$$(\mathrm{div}\, p)_{i,j} = (p^1_{i,j} - p^1_{i-1,j}) + (p^2_{i,j} - p^2_{i,j-1}) \tag{4.22}$$

Because of the duality of the gradient and divergence operators, total variation can also be written as:

$$\sum_{i,j} |\nabla u|_{i,j} = \sup_{z \in K} \langle z, u \rangle_X \quad K = \{\text{div } p : p \in Y, \, |p_{i,j}| \leq 1\} \tag{4.23}$$

The minimizer of the cost function in (4.20) is then obtained by projecting v onto the set λK:

$$u = v - \pi_{\lambda K}(v) \tag{4.24}$$

which is equivalent to minimizing the Euclidian distance between v and $\lambda \operatorname{div} p$, and this can be achieved via the following iteration for computing p:

$$p^0 = 0; \quad p_{i,j}^{n+1} = \frac{p_{i,j}^n + \tau(\nabla(\operatorname{div} p^n - v/\lambda))_{i,j}}{1 + \tau|(\nabla(\operatorname{div} p^n - v/\lambda))_{i,j}|} \tag{4.25}$$

where $\tau > 0$ is the step size. For a small enough step size, $\tau \leq 1/8$, the algorithm is guaranteed to converge [46].

Instead of a global solution, we minimize the cost function (4.4) in a small neighborhood of each pixel. To this end, let us denote by ω the set of indices that define the desired neighborhood around the current pixel. For example, for a square neighborhood of size $(2m+1) \times (2m+1)$ pixels: $\omega = \{(i,j) : i,j = -m : m\}$. We also consider a normalized Gaussian weighting function on this neighborhood:

$$W(i,j) = \exp\left(-\frac{(i^2 + j^2)}{h^2}\right) \tag{4.26}$$

The local problem will then become that of minimizing the following cost function:

$$E_{\lambda,W}(u') = \frac{1}{2}\|u' - v_\omega\|_W^2 + \lambda' \int_\omega |\nabla u'| \tag{4.27}$$

where $\|.\|_W^2$ denotes the weighted norm with weights W and v_ω and u' are images restricted to the window ω around the current pixel. The solution of this local optimization problem will be similar to Chambolle's algorithm described above [195]. The only difference is in the update formula for p:

$$p^0 = 0; \quad p_{i,j}^{n+1} = \frac{p_{i,j}^n + \tau(\nabla(D^{-1}\operatorname{div} p^n - v/\lambda'))_{i,j}}{1 + \tau|(\nabla(D^{-1}\operatorname{div} p^n - v/\lambda'))_{i,j}|} \tag{4.28}$$

where D is a diagonal matrix whose diagonal elements are the values of W.

The regularization parameter, λ', must be chosen according to (4.20). The simplest approach is to set $\lambda' = \lambda v(i,j)$, where λ is a global regularization parameter and $v(i,j)$ is the value of the current pixel in the noisy image. Since $v(i,j)$ is noisy, a better choice is to use a weighted local average as the estimate of the intensity of the true image at the current pixel (note that the maximum-likelihood estimate of the mean of a Poisson process from a set of observations is the arithmetic mean of the observations). Therefore, we suggest the following choice for the local regularization parameter.

$$\begin{aligned} \lambda' = \lambda \frac{\sum_{-a \le i', j' \le a} W'(i',j') v(i-i', j-j')}{\sum_{-a \le i', j' \le a} W'(i',j')} \\ \text{where} \qquad W'(i,j) = \exp\left(-\frac{(i^2+j^2)}{h'^2}\right) \end{aligned} \tag{4.29}$$

There are several parameters in the proposed algorithm. The global regularization parameter λ controls the strength of the denoising. It should be set based on the desired level of smoothing. Parameter m sets the size of the neighborhood considered around each pixel, which in this study was chosen to be a square window of size $(2m+1) \times (2m+1)$. Numerical experiments in [195] have shown that in total variation denoising, the influence map of a pixel is usually limited to a radius of approximately 10 pixels for typical values of the regularization parameter. Therefore, a good value for m would be around 10, which is the value we used for all experiments reported in this chapter. The width of the Gaussian weighting function W is adjusted through h. We used $h = 2m$ which we found empirically to work well. Similarly, a and h' in (4.29) determine the size of the window and the weights used for determining the local regularization parameter. These have to be set based on the noise level in the image; larger values should be chosen when noise is stronger. We used $a = 4$ and $h' = 2a$.

A simple implementation of the proposed algorithm can be computationally intensive because it will involve solving a minimization problem, though very small, for every individual pixel in the sinogram. This will be a major drawback because a big advantage of sinogram denoising methods, compared to iterative image reconstruction methods, is the shorter computational time. To reduce the computational time, after minimizing the local cost function (4.27) around the current pixel, we will replace the value of all pixels in the window of size $(2a+1) \times (2a+1)$ around the current pixel, instead of just the center pixel, and then shift the window by $(2a+1)$. Our extensive numerical experiments with simulated and real projections showed

that with this approach the results will be almost identical to the case where only one pixel is denoised at a time. This is the approach that we followed in all experiments reported in this section.

We evaluated the proposed denoising algorithm on simulated projections and two sets of real low-dose projections of the physical phantom and a rat obtained using the micro-CT scanner. We compared the performance of our proposed algorithm with two other methods:

1. The bilateral filtering algorithm described in Section 3.3. We applied the bilateral filtering for several values of the bandwidth of P_2, which we denote with σ, in the range $[0.7, 2.8]$ and applied the proposed algorithm for several values of the regularization parameter, λ.

2. A nonlocal principal component analysis (NL-PCA) algorithm proposed in [281]. In this method, patches of the image are first clustered using the K-Means algorithm. For all patches in a cluster a Poisson PCA (also known as exponential PCA) is performed to denoise them. The PCA problem is solved using the Newton's method. The denoised patches are returned to their original locations and averaged (to account for the patch overlaps) in order to form the denoised image. Patch-based methods are computationally very intensive. Therefore, with this algorithm we used parameter settings that resulted in a reasonable computational time.

4.3.2 Simulation experiment

We simulated 360 noisy cone-beam projections, from 0° to 359° from a 3D Shepp-Logan phantom according to the model in (1.1). We used two values of $N_0^i = 500$ and 2000 to simulate two sets of projections, which we will call high-noise and low-noise, respectively. The phantom size was $512 \times 512 \times 512$ voxels and the projections were each 700×700 pixels in size.

Figure 4.10 shows one-dimensional profiles of the noisy and denoised projections. The plots in this figure show that the proposed TV-based denoising significantly removes the noise and seems to be superior to bilateral filtering and NL-PCA.

For quantitative comparison, we computed the Root Mean Square of the Error (RMSE), where error is defined as the difference between the denoised and the true (i.e., noise-free) projections, and the mutual information (MI). Figure 4.11 shows the plots of RMSE and MI. For the proposed algorithm, we have plotted these values for 10 logarithmically-spaced values of λ in the range $[0.01, 1]$, which we found to give the best denoising results. For

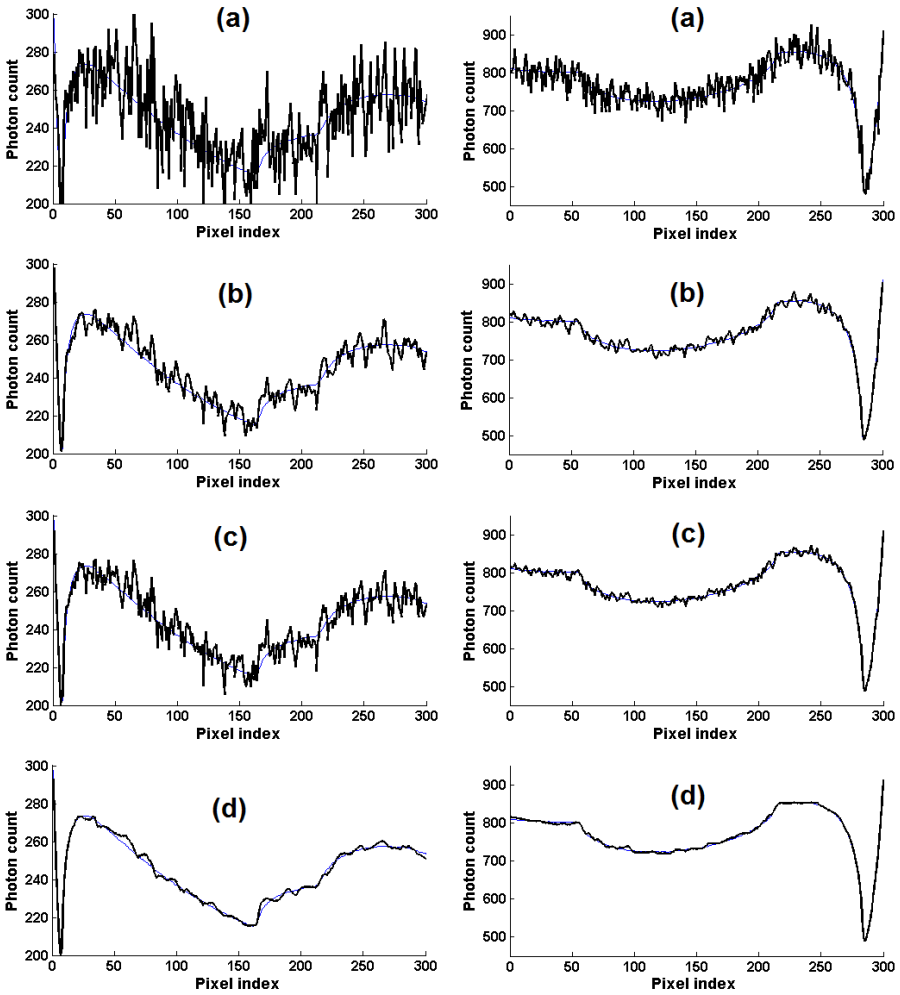

Figure 4.10: Two typical one-dimensional profiles of the noisy and denoised projections simulated from the Shepp-Logan phantom. The thin blue curve in each plot shows the corresponding noise-free projection. The left column is for the high-noise case and the right column is for the low-noise case. (a) the noisy sinogram, and denoised using (b) bilateral filtering, (c) NL-PCA, and (d) the proposed TV-based algorithm.

bilateral filtering, following [219], we have plotted these values for 10 linearly-spaced values of σ in the range $[0.5, 3.2]$. From these plots it is clear that the proposed algorithm has achieved significantly better denoising results than bilateral filtering and NL-PCA. Best results with the proposed algorithm are achieved with λ values around 0.1 and the denoising is too strong for $\lambda > 1$. For bilateral filtering, we found that best denoising results were usually obtained for values of σ close to 3.0 and the performance did not improve or slightly deteriorated when σ was increased beyond 3.2. The solid squares on these plots show the optimum value of the corresponding parameter (i.e., lowest RMSE or highest MI). The phantom profiles shown in Figure 4.10 for the proposed algorithm and bilateral filtering were obtained with the parameter values that resulted in the lowest RMSE.

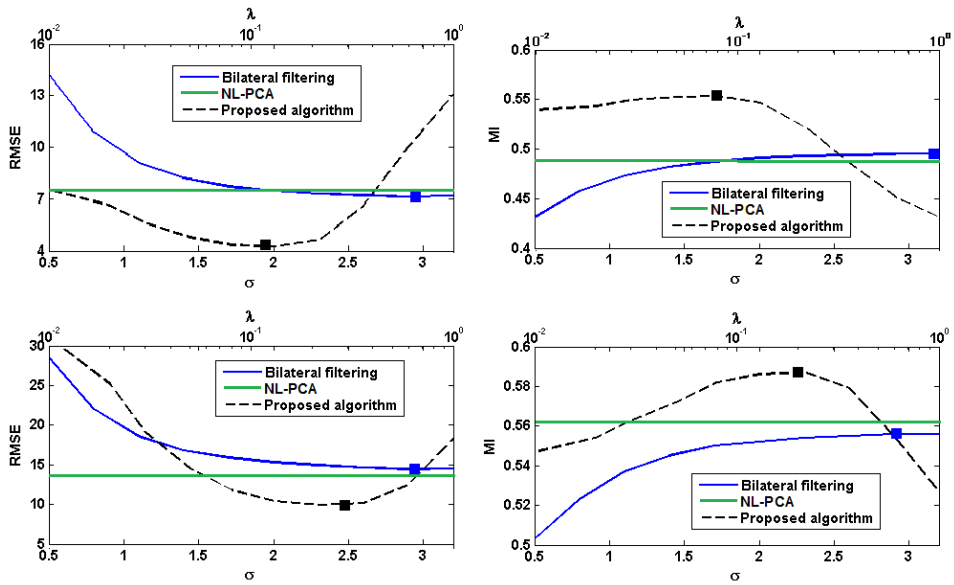

Figure 4.11: Comparison between different denoising algorithms in terms of RMSE and MI for the high-noise projections (top row) and low-noise projections (bottom row) simulated from the Shepp-Logan phantom. Values for the bilateral filtering algorithm are plotted as a function of σ (the bottom horizontal axis), whereas the values for the proposed algorithm are plotted as a function of the regularization parameter λ (the top horizontal axis). The solid squares indicate the points of optimum.

4.3.3 Experiment with real micro-CT data

Cone-beam projections were acquired from the physical phantom using the Gamma Medica micro-CT scanner. Two scans of the phantom were generated:

1. **Low-noise scan.** Consisting of 720 projections of size 875×568 pixels between $0°$ and $360°$ at $0.5°$ intervals. The tube voltage, tube current, and exposure time were 70 kV, 40 mA, and 25 ms, respectively.

2. **High-noise scan.** Consisting of 240 projections of size 875×568 pixels between $0°$ and $360°$ at $1.5°$ intervals. The tube voltage, tube current, and exposure time were 50 kV, 32 mA, and 16 mAs, respectively.

We used the low-noise scan to reconstruct a high-quality reference image of the phantom using the FDK algorithm. To evaluate the denoising algorithms, we applied them on the high-noise projections, reconstructed the image of the phantom from the denoised projections using the FDK algorithm, and compared the reconstructed image with the reference image. Similar to the experiment with the simulated projections, we performed the denoising for 10 linearly-spaced values of σ in the range $[0.5, 3.2]$ for bilateral filtering. Similarly, we ran the proposed algorithm with 10 logarithmically-spaced values of λ in the range $[0.001, 0.1]$. In order to assess the overall quality of the reconstructed images, we computed the RMSE and SSIM. The plots of RMSE and SSIM are shown in Figure 4.12. Compared with both bilateral filtering and NL-PCA, the image reconstructed from projections denoised using the proposed algorithm has a significantly lower RSME and higher SSIM. Best results in terms of SSIM with the proposed algorithm are obtained with $\lambda = 0.0129$ and for bilateral filtering algorithm with $\sigma = 2.6$.

Figure 4.13 shows two of the fine coils in the images of the phantom reconstructed from noisy and denoised projections. These coils have thicknesses of 500 μm and 200 μm, corresponding to spatial resolutions of 1 and 2.5 line pairs per mm, respectively. The image shown for the proposed algorithm corresponds to $\lambda = 0.0129$ and the image shown for bilateral filtering corresponds to $\sigma = 2.6$. As we mentioned above, these parameter values led to highest SSIM. The images show a marked improvement in the image quality via sinogram denoising. It also seems that the proposed algorithm leads to a smoother image without affecting the spatial resolution. In Figure 4.14 we have shown a profile through the center of the 500-μm coil for the images reconstructed from noisy and denoised projections and also the difference between them and the reference image for a closer comparison. It is

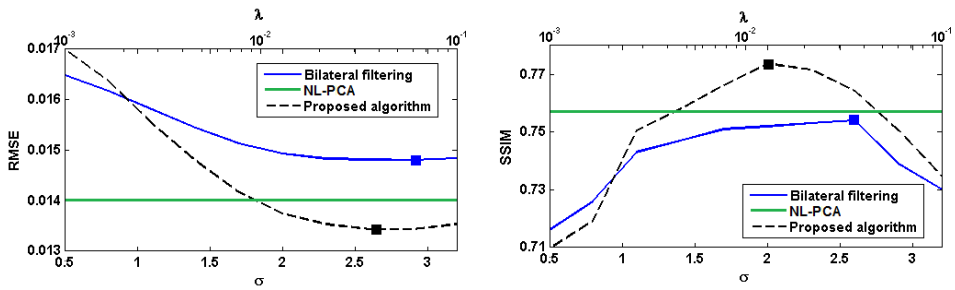

Figure 4.12: Performance comparison between different sinogram denoising algorithms in terms of RMSE and SSIM on the scan of the physical phantom. Values for the bilateral filtering algorithm are plotted as a function of σ (the bottom horizontal axis), whereas the values for the proposed algorithm are plotted as a function of the regularization parameter λ (the top horizontal axis). The solid squares indicate the points of optimum.

clear from these profiles that the image reconstructed from the projections denoised using the proposed algorithm are closer to the reference image.

In order to compare the denoising algorithms in terms of the trade-off between noise and spatial resolution, we followed an approach similar to that in Section 4.2. Specifically, we computed the following two numbers as measures of spatial resolution and noise level in the reconstructed image of the phantom:

Measure of spatial resolution. We estimated the MTF as described in Section 4.2.3 and used the spatial frequency at which the normalized MTF reached a value of 0.10 as a measure of spatial resolution.

Measure of noise level. We selected five cubes in the uniform polycarbonate disk in the phantom, each $10 \times 10 \times 10$ voxels, at different locations within this disk and computed the standard deviation of the voxel values in each cube. We use the average standard deviation of voxel values in these cubes as a measure of noise level.

In Figure 4.15, we have shown plots of these two values for the three denoising algorithms. Note that a high spatial resolution and a low noise level are desirable. Therefore, all three denoising algorithms have improved the quality of the reconstructed image for the range of parameter values used (except for $\lambda = 0.1$ with the proposed algorithm). Moreover, the proposed

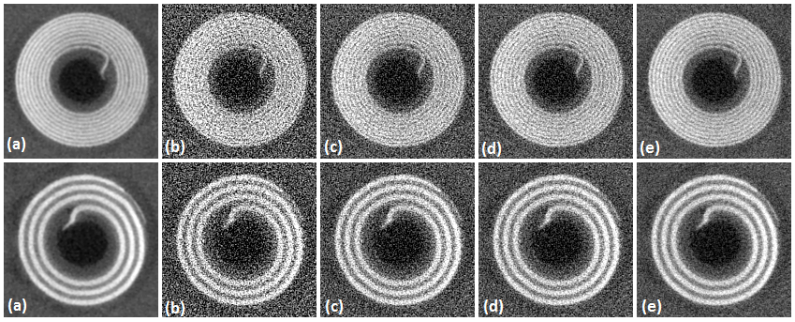

Figure 4.13: The 200-μm (top row) and 500-μm (bottom row) coils in the images reconstructed from noisy and denoised projections of the physical phantom; (a) the reference image, (b) without denoising, (c) bilateral filtering, (d) NL-PCA, and (e) the proposed algorithm.

algorithm has achieved better results than bilateral filtering and NL-PCA. Specifically, for $\lambda \in [0.0077, 0.0359]$ the proposed algorithm has achieved both higher spatial resolution and lower noise than bilateral filtering (for any parameter value) and NL-PCA.

In Figure 4.15, we have also shown plots of the MTF obtained with the three denoising algorithms. All three sinogram denoising algorithms have led to an improvement in the spatial resolution in the reconstructed image. The proposed algorithm has resulted in a higher MTF than bilateral filtering and NL-PCA for all spatial frequencies.

A rat was scanned using the micro-CT scanner. Because the internal organs of the rat constantly moved, it was not possible to create two identical scans with different noise levels as we did for the phantom. Therefore, the rat was scanned only once. The scan consisted of 720 projections of size 875×568 pixels between $0°$ and $360°$ at $0.5°$ intervals with the tube voltage, tube current, and exposure time equal to 70 kV, 32 mA, and 16 ms, respectively. To create a high-quality reference image from the full set of 720 projections, we first reconstructed an initial image using the FDK algorithm. Then, we used 50 iterations of MFISTA algorithm [17] to improve the quality of the FDK-reconstructed image. We applied the denoising algorithms on a subset of 240 projections of the same scan (projections at $1.5°$ intervals) and reconstructed the image of the rat using the FDK algorithm.

Similar to the physical phantom experiment, we use RMSE and SSIM as a measure of the overall closeness of the reconstructed images to the reference image. Figure 4.16 shows these criteria for the three sinogram

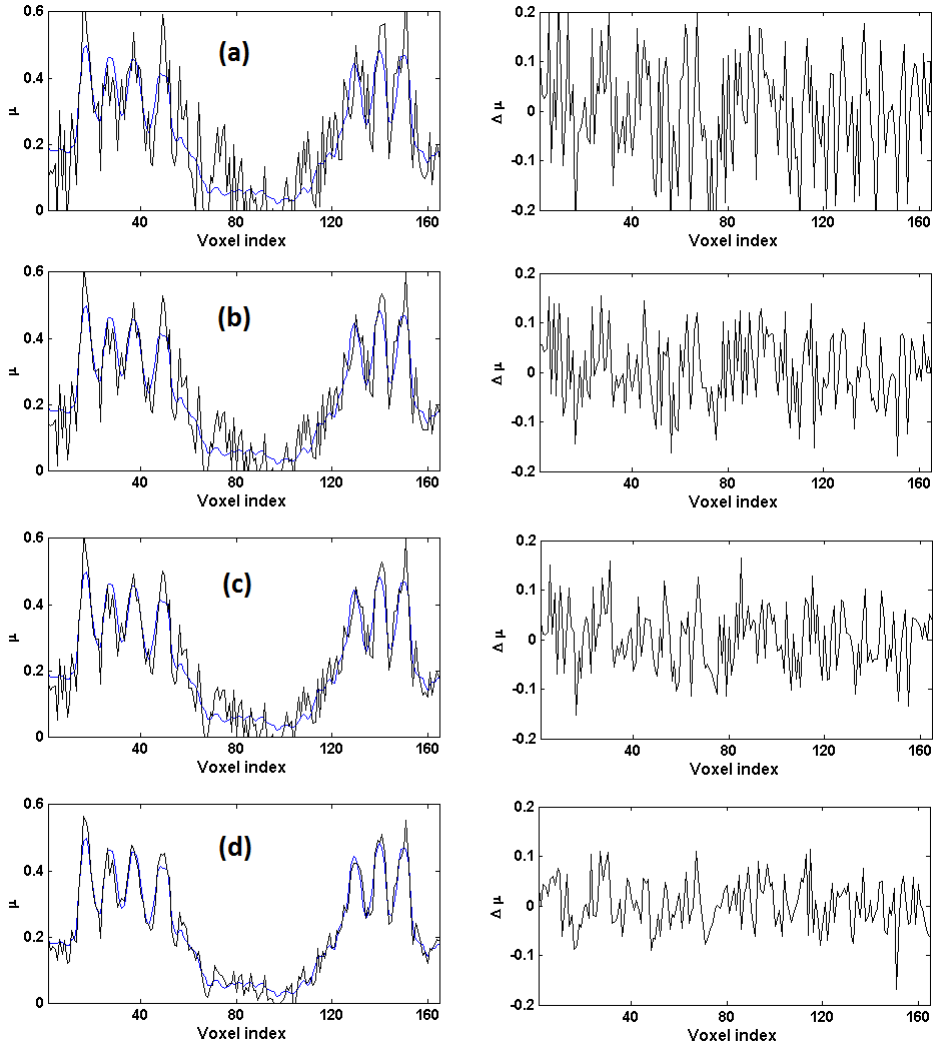

Figure 4.14: The left column shows a profile through the 500-μm coil in the images of the physical phantom reconstructed from noisy and denoised projections: (a) without denoising, (b) bilateral filtering, (c) NL-PCA, and (d) the proposed algorithm. In these plots, the blue curve is the profile of the reference image. The right column shows the difference between the profiles shown in the left column and the profile of the reference image.

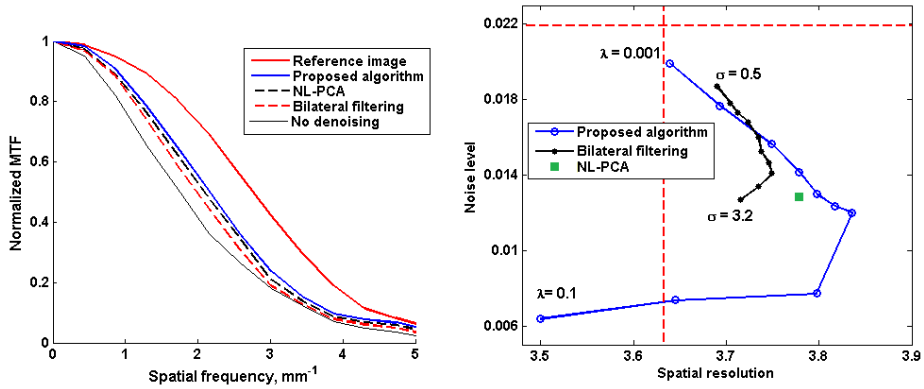

Figure 4.15: Left: plots of the normalized MTF obtained by different sinogram denoising algorithms. Right: plots of noise level versus spatial resolution for different denoising algorithms. The dashed lines in this plot show the corresponding values for the image reconstructed without sinogram denoising.

denoising algorithms. From this figure, denoising of the projections with the proposed algorithm has lead to superior results in terms of RMSE and SSIM compared to bilateral filtering and NL-PCA.

For a visual comparison, Figure 4.17 shows a 2D slice of the reconstructed images. For the proposed algorithm and bilateral filtering, the images shown in this figure were obtained using the parameter values that resulted in the lowest SSIM, i.e., $\lambda = 0.0129$ and $\sigma = 2.6$ (see Figure 4.16).

The window of the linear attenuation coefficient, μ, used to display the whole slices is $[0, 0.55]$. To allow a better visual comparison, we have selected two regions of interest (ROI) within this slice and have shown them in zoomed-in views and with narrower μ-windows. The ROI shown on the top left of each slice contains fat surrounded with soft tissue; this ROI is shown with a magnification factor of 1.5 and with a μ-window of $[0.15, 0.20]$. The ROI shown on the top right of each slice contains bone surrounded with soft tissue; this ROI is shown with a magnification factor of 2.0 and with a μ-window of $[0.18, 0.50]$. These images show a strong positive effect for sinogram denoising in terms of the visual quality of the reconstructed image. Moreover, denoising with the proposed algorithm seems to have resulted in a higher-quality image, especially in the soft-tissue ROI.

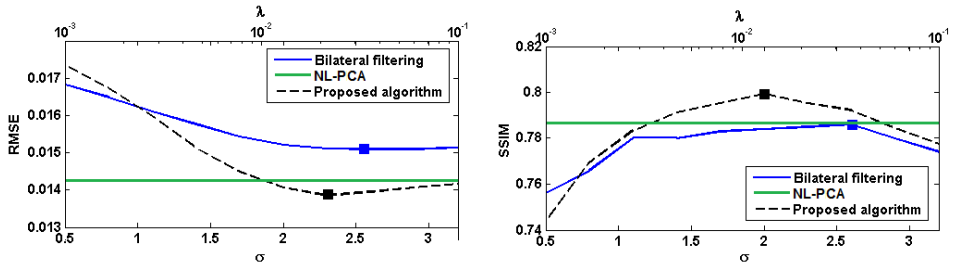

Figure 4.16: Comparison of different sinogram denoising algorithms on the rat scan.

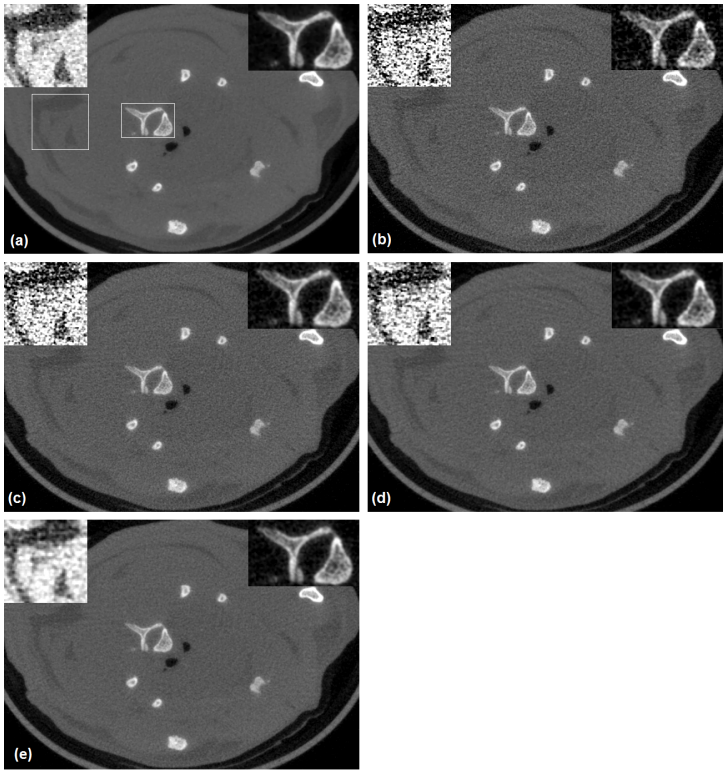

Figure 4.17: A slice of the image of the rat reconstructed from noisy and denoised projections: (a) the reference image, (b) without denoising, (c) bilateral filtering, (d) NL-PCA, and (e) the proposed algorithm. The locations of the selected ROIs have been marked on the reference image (a).

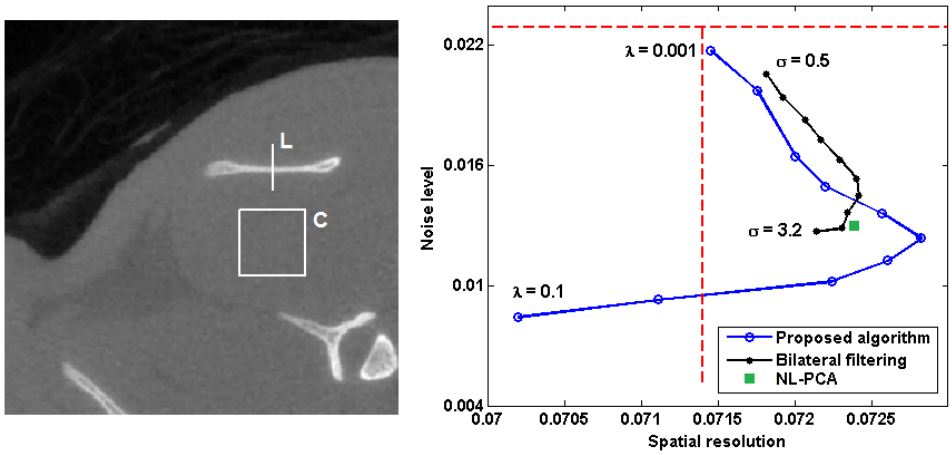

Figure 4.18: Left: the ROI used to compute the noise level and spatial resolution in the reconstructed images of the rat; the noise level was computed as the standard deviation of voxel values in the cube C and the spatial resolution was computed as the maximum gradient along the line L. Right: plots of noise level versus spatial resolution for the three denoising algorithms. The dashed horizontal and vertical lines in this plot show the corresponding values for the image reconstructed without sinogram denoising.

In order to compare the denoising algorithms in terms of the trade-off between noise suppression and spatial resolution, we selected an ROI shown in Figure 4.18 and computed the following measures of noise level and spatial resolution:

Measure of spatial resolution. We compute the maximum absolute value of the gradient (i.e., slope) along the line L marked in the ROI shown in Figure 4.18 as a measure of spatial resolution.

Measure of noise level. We consider a cube of size $50 \times 50 \times 50$ voxels, the cross-section of which is shown in the displayed ROI. From the reference image, we identified this cube as being highly uniform. Therefore, we computed the standard deviation of the voxel values in this cube as a measure of noise level.

The results are plotted in Figure 4.18. This plot is very similar to the plot shown for the physical phantom experiment in Figure 4.15. The main observations are that all three sinogram denoising algorithms have improved

the quality of the reconstructed image in terms of spatial resolution and noise level, and that the proposed algorithm can outperform the bilateral filtering algorithm and NL-PCA with the right selection of the regularization parameter. Specifically, with $\lambda \in [0.0129, 0.0359]$, the proposed algorithm has resulted in lower noise and better spatial resolution than bilateral filtering (with any choice of σ) and NL-PCA.

4.3.4 Discussion

Overall, the results of our experiments show that the proposed algorithm performs better than bilateral filtering. We should emphasize that it is likely that NL-PCA can outperform our proposed TV-based denoising algorithm, albeit at a much higher computational cost. In this study, for NL-PCA we did not use the parameter values that the authors of [281] had suggested. Instead, we chose parameter values that resulted in a relatively short computational time. For example, the authors of [281] suggest patch sizes of 20×20 pixels, but we used patches of size 8×8 pixels.

In order to compare the computational time of the proposed algorithm with that of bilateral filtering and NL-PCA, we considered the denoising of 240 projections of the rat scan. As we mentioned above, each projection in this scan was 875×568 pixels. The proposed TV-based algorithm implemented in Matlab version R2012b and executed on a Windows 7 PC with 16 GB of memory and 3.4 GHz Intel Core i7 CPU needed approximately 6 minutes to denoise all 240 projections. In comparison, bilateral filtering and NL-PCA needed 8.5 minutes and 42 minutes, respectively, for the same denoising task.

In general, our experience shows that the patch-based algorithm proposed in Chapter 3 can achieve better results than the two TV-based algorithms proposed in this chapter. Another advantage of the patch-based denoising algorithm proposed in Chapter 3 is that it is less sensitive to the choice of its parameters. Both algorithms proposed in this chapter will perform poorly if their parameter(s) are not selected properly. Compared with the algorithm proposed in this section that has only one parameter, choosing the parameter values is much harder for the algorithm proposed in Section 4.2 because the number of parameters is larger and their interactions is complex. The advantage of the TV-based denoising algorithms, compared with the patch-based algorithms, is that they are usually much faster. Most of the computational time required by the dictionary-based methods includes the time needed for learning the dictionary. The results of Chapter 3 show that if the scan geometry does not change, there is no

need to learn a new dictionary. However, if the scan geometry or the angular spacing between successive projections change, a new dictionary will have to be learned. Some other patch-based methods (such as NL-PCA algorithm) learn the dictionary from the noisy sinogram. Therefore, such methods are expected to be very computationally intensive.

Chapter 5

Sinogram Interpolation

5.1 Introduction

As we mentioned previously in this dissertation, in practice there are two basic ways to reduce the radiation dose used in CT imaging: (1) lowering the x-ray photon current, and/or (2) reducing the number of projection measurements taken. However, if filtered-backprojection methods are used to reconstruct an image from such noisy and/or undersampled measurements, the quality of the produced image will be very low. In this chapter, we propose a sinogram interpolation algorithm for cone-beam CT. Our algorithm exploits both smoothness and self-similarity of the sinogram. We apply the proposed algorithm on simulated and real cone-beam CT projections and compare it with an algorithm that is based on learned dictionaries.

A schematic of the cone-beam CT and a sample sinogram of a brain phantom are shown in Figure 5.1(a). The variation of the photon flux incident on the detectors follows a Poisson distribution. As we mentioned in Section 1.2, after the logarithm transformation the noise in the projection measurements follows approximately a Gaussian distribution with signal-dependent variance. Specifically, let us denote the true line integral of the attenuation coefficient along the line from the x-ray source to the detector indexed with (i, j) in the kth projection with $y_t(i, j, k)$. In other words, i and j indicate the detector location on the detector plane and k indicates the rotation angle angle, θ. Then the noisy measurement $y_n(i, j, k) \sim \mathcal{N}(y_t(i, j, k), \sigma^2_{ijk})$, where $\sigma^2_{ijk} \propto \exp(y_t(i, j, k))$. In this chapter we use this Gaussian noise model because our experience shows that our proposed interpolation algorithm works well with this noise model, even on very low dose scans.

We assume that only a portion of the desired projections have been directly measured and the rest are to be estimated (i.e., interpolated). Although the algorithm that we propose can be applied on very general cases, in order to simplify the presentation of the algorithm, here we consider the case depicted in Figure 5.1(b). Specifically, we assume that only half of the n_θ desired projection views have been measured and the remaining

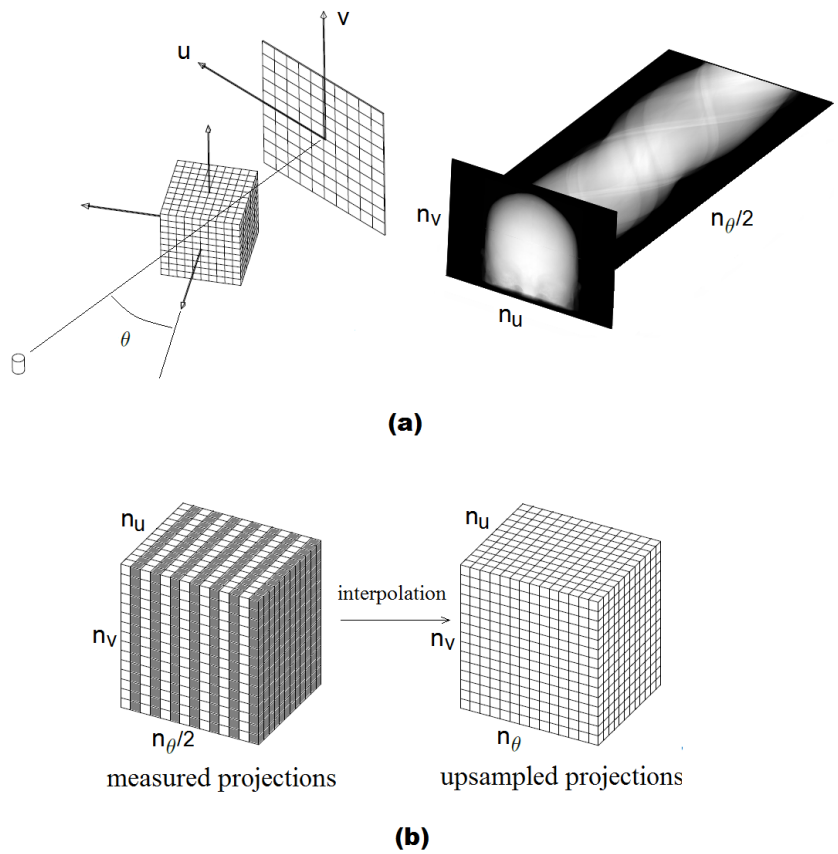

Figure 5.1: (a) A schematic of cone-beam CT and sinogram of a brain phantom, (b) A schematic representation of the sinogram interpolation problem.

half are to be estimated. In other words, the measured noisy sinogram is $y_n \in \mathbb{R}^{n_u \times n_v \times n_\theta/2}$ and we would like to estimate the interpolated sinogram $y \in \mathbb{R}^{n_u \times n_v \times n_\theta}$. As we will explain later, even though our main goal is sinogram interpolation to estimate the missing projection views, the proposed algorithm also has an excellent denoising effect. Therefore, we estimate the full set of n_θ projection views, not only the missing $n_\theta/2$ views.

Given a noisy sinogram y_n, we propose to estimate the true sinogram y by minimizing the following cost function:

$$J(y) = \|My - y_n\|_W^2 + \lambda_s R_s(y) + \lambda_h R_h(y) \tag{5.1}$$

The first term in J is obtained simply by maximizing the log-likelihood of

the measurements. In this term, M is a binary mask matrix that removes from y those projection views that have not been measured and W is a diagonal weight matrix whose diagonal elements are inversely proportional to the measurement variance. The regularization functions R_s and R_h will be explained in Subsections 5.2.1 and 5.2.2, respectively. R_s is a regularization function in terms of nonlocal similarities, R_h is a regularization function in terms of smoothness, and λ_s and λ_h are regularization parameters.

5.2 The proposed algorithm

5.2.1 Regularization in terms of sinogram self-similarity

In Section 2.2, we reviewed some of the applications of image processing algorithms that are based on nonlocal patch similarities. Inspired by the great success of these algorithms and because this type of self-similarity is very abundant in sinogram (even more than in natural images, as can be seen in the sample sinogram in Figure 5.1(a)), we suggest a similar form for R_s:

$$R_s(y) = \|y - y^*\|_2^2 \quad \text{where:}$$
$$y^*(i,j,k) = \sum_{(i',j',k')} \frac{G_a(z[i',j',k'] - y[i,j,k])}{\sum\limits_{(i',j',k')} G_a(z[i',j',k'] - y[i,j,k])} z(i',j',k') \tag{5.2}$$

Computation of y^* has a few differences with the basic NLM in Equation (2.14). Firstly, we work with 3D blocks instead of 2D patches. We stack the 2D projections to form a 3D image, as shown in Figure 5.1(b), and work with small blocks of this image. This will allow us to exploit both the correlation between adjacent pixels within a projection as well as the correlation between pixels in adjacent projection views. Therefore, in the above equation, $y[i,j,k]$ is a small block centered on pixel $y(i,j,k)$. Secondly, unlike Equation (2.14) where patch similarities in the same image are exploited, in Equation (5.2) we use a second image, as shown in Figure 5.2. This second image, denoted with z in Equation (5.2) and Figure 5.2 is built by stacking the projections of the scan of a similar object. For instance, this can be a previous scan of the same patient or of a different patient. We will explain our justification for this choice below. Thirdly, for computing $y^*(i,j,k)$, we first find a small number of blocks in z that are very similar to $y[i,j,k]$ and use those blocks only, instead of all blocks in the image. In other words, in Equation (5.2), both summations are over $(i',j',k') \in \Omega_{i,j,k}$,

where $\Omega_{i,j,k}$ denotes the indices of these blocks. This is a necessary compromise to keep the computational time reasonable and it is followed by all practical implementations of nonlocal patch-based methods.

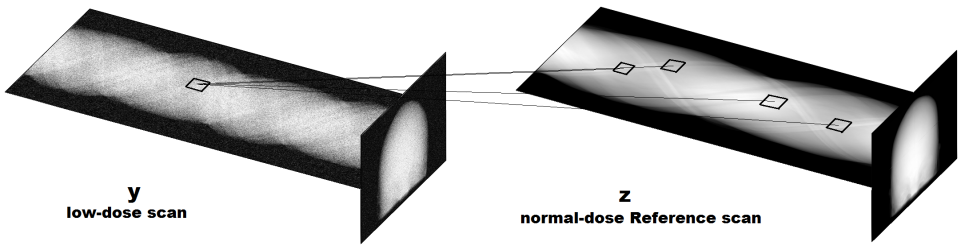

Figure 5.2: Block matching between the noisy scan to be restored y and the high-dose reference scan z.

Computation of R_s requires that for each pixel $y(i,j,k)$ we consider a small block around it, $y[i,j,k]$, and find a set of blocks sufficiently similar to $y[i,j,k]$. Even for medium-size 2D images, this can be very computationally costly and many algorithms have been proposed for reducing the computational load. Here, we use the Generalized PatchMatch algorithm [12, 13] for this purpose.

The Generalized PatchMatch algorithm is an iterative stochastic algorithm for finding a set of k similar patches in x_{ref} for every patch in x. Therefore, the goal of this algorithm is not to find the set of k most similar patches, but only to find a set of k very similar patches. Let us denote the set of indices of the k similar patches to $x[i]$ in x_{ref} with $\mathcal{S}_i$. The algorithm starts by random assignment; i.e., for each patch $x[i]$, $\mathcal{S}_i$ is chosen to be k random patches in x_{ref}. This random assignment is then iteratively improved via a set of very effective heuristics, called *propagation*, *random search*, and *enrichment*. We describe these steps very briefly here; implementation details can be found in [12, 13]. In propagation, similar patches are shared among neighboring pixels. In other words, the algorithm seeks to improve $\mathcal{S}_i$ by examining $\mathcal{S}_{\mathcal{N}_i}$, where $\mathcal{N}_i$ is the set of immediate neighbors of $x(i)$. In random search, the algorithm seeks to improve $\mathcal{S}_i$ by examining a window around the patches that have been identified as good matches. In other words, the algorithm examines $\{j + \mathcal{M},\ \forall j \in \mathcal{S}_i\}$. Here, $\mathcal{M}$ is a search window, whose size is exponentially reduced with more iterations. In enrichment, good matches are propagated in the "patch space". In other words, the algorithm seeks to improve $\mathcal{S}_i$ by examining $\{\mathcal{S}_j,\ \forall j \in \mathcal{S}_i\}$ or $\{\mathcal{S}_j,\ \forall i \in \mathcal{S}_j\}$, called forward enrichment and inverse enrichment, respectively. In forward enrichment the idea is that a patch that has been identified

as similar to $x[i]$ is likely to have, in its own set of similar patches, more similar patches to $x[i]$. The intuition behind inverse enrichment is similar.

We follow this algorithm exactly as in [13], except that in the first iteration we do not use a random neighborhood. Instead, to find blocks similar to $y[i,j,k]$, we search a neighborhood around $z(i,j,k)$ because if y and z are similar scans (e.g., scans of the same body part of the same patient or different patients) similar blocks are likely to exist in similar spatial locations in the two scans. Our experience shows that with a proper choice of z, this approach works much better than finding similar blocks from the same image. We should also note that in finding similar blocks and in computing the block differences in (5.2), we only include the pixels from projections that have been directly measured.

5.2.2 Regularization in terms of sinogram smoothness

An important characteristic of sinogram is its smoothness. Inspired by the success of the denoising algorithm that we proposed in Section 4.2, we model this smoothness via the ℓ_1 norm of the Hessian of the sinogram. We suggest the following form for R_h:

$$R_h(y) = \sum_{i,j,k} \left(|\nabla^2_{uv} y(i,j,k)| + |\nabla^2_{v\theta} y(i,j,k)| + |\nabla^2_{\theta u} y(i,j,k)| \right) \tag{5.3}$$

In other words, we compute the 2D Hessians in the three orthogonal planes and add their ℓ_1 norms. We compute $|\nabla^2_{uv} y(i,j,k)|$ as:

$$|\nabla^2_{uv} y(i,j,k)| = \left(D_{uu} y(i,j,k)^2 + 2D_{uv} y(i,j,k)^2 + D_{vv} y(i,j,k)^2 \right)^{1/2} \tag{5.4}$$

and similarly for $|\nabla^2_{v\theta} y(i,j,k)|$ and $|\nabla^2_{\theta u} y(i,j,k)|$. The forward difference are defined as in Equation (4.9). For pixels at the boundaries, we use periodic boundaries as suggested in [248].

Therefore, the proposed cost function has the form below, where we have dropped the pixel indices to simplify the expressions.

$$\begin{aligned} J(y) = \sum_{i,j,k} \Bigg(& \|My - y_n\|^2_W + \lambda_s \|y - y^*\|^2_2 \\ & + \lambda_h \Big(|\nabla^2_{uv} y| + |\nabla^2_{v\theta} y| + |\nabla^2_{\theta u} y| \Big) \Bigg) \end{aligned} \tag{5.5}$$

In summary, the first term encourages consistency with the portion of the sinogram that has been measured. The second term ($R_s(y)$) is the term that actually performs the interpolation. It does so by matching blocks from a similar scan. The last term ($R_h(y)$) promotes smoothness by penalizing the ℓ_1 norm of the Hessian.

5.2.3 Optimization algorithm

An estimate of the interpolated (and denoised) projections is obtained as a minimizer of $J(y)$. To perform this minimization, we use the split Bregman iterative algorithm, which has much similarities with the algorithm used for sinogram denoising in Section 4.2. The split Bregman method first converts the unconstrained optimization problem of minimizing $J(y)$ into a constrained problem:

$$\begin{aligned} \textbf{minimize } & \sum \left(\|Mf - y_n\|_W^2 + \lambda_s \|f - y^*\|_2^2\right) \\ & \qquad + \lambda_h \sum |g_1| + \lambda_h \sum |g_2| + \lambda_h \sum |g_3| \\ \textbf{subject to: } & f = y,\ g_1 = \nabla^2_{uv} y,\ g_2 = \nabla^2_{v\theta} y,\ g_3 = \nabla^2_{\theta u} y \end{aligned} \tag{5.6}$$

This constrained optimization problem is solved via Bregman iteration:

$$\begin{aligned}
&\textbf{Initialize: } y^0 = y_n,\ f^0 = y_n,\ g_1^0 = \nabla^2_{uv} y_n,\ g_2^0 = \nabla^2_{v\theta} y_n,\ g_3^0 = \nabla^2_{\theta u} y_n, \\
&\qquad\qquad b_1^0 = 0,\ b_2^0 = 0,\ b_3^0 = 0,\ b_4^0 = 0 \\
&\textbf{while } ||y^k - y^{k-1}||_2^2 > \epsilon \\
&\quad [y^{k+1}, f^{k+1}, g_1^{k+1}, g_2^{k+1}, g_3^{k+1}] = \\
&\qquad \underset{u,f,g_1,g_2,g_3}{\arg\min} \sum \left(\|Mf - y_n\|_W^2 + \lambda_s \|f - y^*\|_2^2\right) \\
&\qquad\qquad + \lambda_h \sum |g_1| + \lambda_h \sum |g_2| + \lambda_h \sum |g_3| \\
&\qquad\qquad + \frac{\mu_1}{2} \sum (f - y - b_1^k)^2 + \frac{\mu_2}{2} \sum (g_1 - \nabla^2_{uv} y - b_2^k)^2 \\
&\qquad\qquad + \frac{\mu_2}{2} \sum (g_2 - \nabla^2_{v\theta} y - b_3^k)^2 + \frac{\mu_2}{2} \sum (g_3 - \nabla^2_{\theta u} y - b_4^k)^2 \\
&\quad b_1^{k+1} = b_1^k + y^{k+1} - f^{k+1} \\
&\quad b_2^{k+1} = b_2^k + \nabla^2_{uv} y^{k+1} - g_1^{k+1} \\
&\quad b_3^{k+1} = b_3^k + \nabla^2_{v\theta} y^{k+1} - g_2^{k+1} \\
&\quad b_4^{k+1} = b_4^k + \nabla^2_{\theta u} y^{k+1} - g_3^{k+1} \\
&\textbf{end}
\end{aligned}$$

The advantage of this reformulation is that the above minimization problem can be split into five smaller problems, one for each of the five variables that can be solved more easily. Minimization with respect to f has a simple closed-form solution. Minimizations with respect to g_1, g_2, and g_3 have simple soft-thresholding solutions similar to that shown in Equation (4.17). Minimization with respect to y is more difficult. We solve this optimization approximately by considering each of the three Hessian terms in turn. Each of these three sub-problems will involve a linear system which can be approximately solved using the Gauss-Seidel method.

The regularization parameters λ_s and λ_h influence the recovered solution, whereas μ_1 and μ_2 influence the convergence speed. We do not discuss the effects of these parameters as their effects are very similar to the denoising problem discussed in Section 4.2 and also in [248, 249]. In our experiments we used $\lambda_h = 0.0001$, $\mu_1 = \mu_2 = 0.001$, and $\lambda_s = 1$.

5.3 Results and discussion

We applied the proposed algorithm on simulated and real cone-beam CT projections. As mentioned previously in this chapter, unlike the iterative reconstruction methods that aim at reconstructing a high-quality image from undersampled projections, our goal is to estimate the "missing" projections. Therefore, in all of the experiments reported in this chapter, for image reconstruction we used the FDK algorithm. We compare our algorithm with the dictionary-based sinogram interpolation method proposed in [188], which has been shown to be better than spline interpolation.

5.3.1 Experiment with simulated data

We first applied our algorithm on scans simulated from a brain phantom, which we obtained from the BrainWeb database [63]. We simulated n_θ projections from this phantom, for two values of $n_\theta = 1440$ and 960. For each n_θ, we first reconstructed the image of the phantom from the full set of n_θ projections and from $n_\theta/2$ projections; we denote these images with x_{n_θ} and $x_{n_\theta/2}$, respectively. We then applied the proposed algorithm and the dictionary-based interpolation algorithm to interpolate the subset of $n_\theta/2$ projections to generate n_θ projections and reconstructed the image of the phantom from the interpolated projections. We will denote these images with $x^{\text{proposed}}_{n_\theta/2}$ and $x^{\text{dict.}}_{n_\theta/2}$. We simulated two levels of noise in the projections with different number of incident photons: $N_0 = 10^6$ and $N_0 = 5 \times 10^4$. We

will refer to these simulations as low-noise and high-noise, respectively. For both simulations, we assumed the detector electronic noise to be additive Gaussian with a standard deviation of 40. As we mentioned in Section 5.2.1, for the block-matching required for computation of R_s, we use another scan (denoted with z in Equation (5.2) and Figure 5.2), which can be the scan of another object or a previous scan of the same object. In this experiment, we used the simulated scan of a different brain phantom from the same database.

We compared the reconstructed images with the true phantom image by computing the root-mean-square of error (RMSE) and the Structural Similarity (SSIM) index. The results of this comparison are presented in Table 5.1. Sinogram interpolation with the proposed algorithm has resulted in a large improvement in the objective quality of the reconstructed image. The improvement is more substantial in the case of high-noise projections. This is because, as we mentioned above, both regularization terms R_s and R_h have excellent denoising effects. Therefore, the proposed algorithm results in an automatic denoising. The proposed algorithm has also outperformed the interpolation algorithm based on learned dictionaries. The objective quality of $x_{n_\theta/2}^{\text{proposed}}$ is very close to x_{n_θ} in the low-noise case and better than x_{n_θ} in the high-noise case.

			x_{n_θ}	$x_{n_\theta/2}$	$x_{n_\theta/2}^{\text{proposed}}$	$x_{n_\theta/2}^{\text{dict.}}$
$n_\theta = 1440$	Low-noise	RMSE	0.104	0.140	0.111	0.128
		SSIM	0.745	0.683	0.726	0.705
	High-noise	RMSE	0.124	0.157	0.121	0.136
		SSIM	0.710	0.642	0.715	0.686
$n_\theta = 960$	Low-noise	RMSE	0.127	0.154	0.130	0.138
		SSIM	0.708	0.640	0.691	0.671
	High-noise	RMSE	0.143	0.164	0.136	0.144
		SSIM	0.688	0.625	0.690	0.662

Table 5.1: Objective quality of the reconstructed images of the brain phantom.

Figure 5.3 shows a slice in the reconstructed images of the phantom. Interpolation of the sinogram with the proposed algorithm has resulted in a substantial improvement in the visual quality of the reconstructed image. Not only artifacts have been significantly reduced, noise has also been decreased substantially.

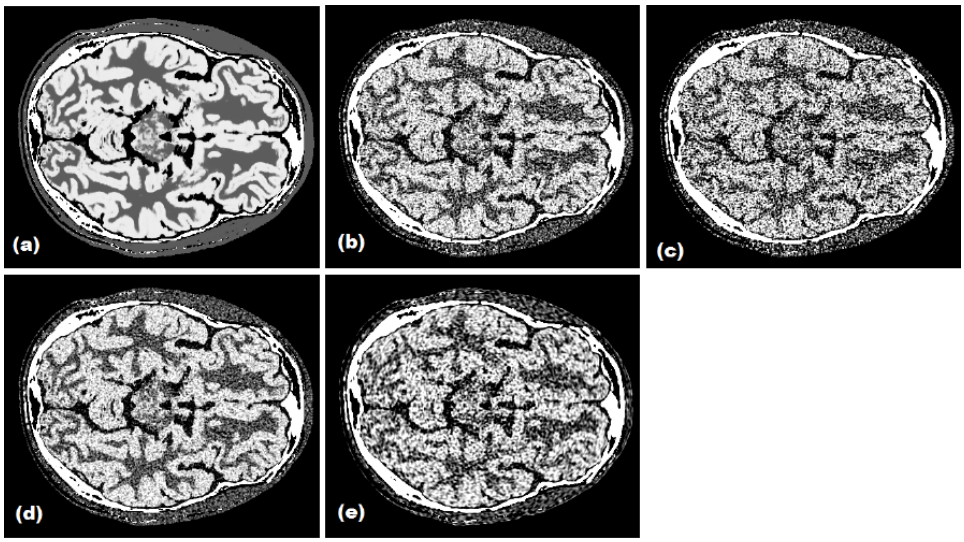

Figure 5.3: A slice in the images of the brain phantom reconstructed from the high-noise projections with $n_\theta = 1440$. (a) the true phantom, (b) x_{n_θ}, (c) $x_{n_\theta/2}$, (d) $x^{\text{proposed}}_{n_\theta/2}$, (e) $x^{\text{dict.}}_{n_\theta/2}$.

5.3.2 Experiment with real CT data

We used micro-CT scans of a rat for this experiment. The scan consisted of 720 projections. We reconstructed a high-quality "reference" image of the rat from all 720 projections using 25 iterations of MFISTA algorithm. Then, we selected subsets of n_θ projections from this scan for two values of $n_\theta = 360$ and 180. For each n_θ, we reconstructed the image of the rat from n_θ and $n_\theta/2$ projections. We then interpolated the subset of $n_\theta/2$ projections using the proposed algorithm and the dictionary-based interpolation algorithm to obtain n_θ projections and reconstructed the image of the rat from the interpolated projections.

We will refer to the scan described above as normal-dose scan because it was obtained at normal dose used in routine imaging. The same rat was scanned at much reduced dose (by reducing the mAs setting to half of that in routine imaging and using additional copper filtration) and the same analysis as above was performed. We will refer to this scan as the low-dose scan. For the block-matching in the computation of R_s we used the scan of a different rat.

To assess the quality of the reconstructed images, we computed the

			x_{n_θ}	$x_{n_\theta/2}$	$x^{\text{proposed}}_{n_\theta/2}$	$x^{\text{dict.}}_{n_\theta/2}$
$n_\theta = 720$	Normal-dose	RMSE	0.0173	0.0222	0.0185	0.0191
		SSIM	0.654	0.611	0.640	0.636
		CNR	13.8	11.2	13.6	12.3
	Low-dose	RMSE	0.0194	0.0236	0.0189	0.0199
		SSIM	0.630	0.597	0.637	0.623
		CNR	12.3	10.5	13.0	12.1
$n_\theta = 360$	Normal-dose	RMSE	0.0200	0.0238	0.0208	0.0216
		SSIM	0.632	0.582	0.624	0.617
		CNR	13.0	10.9	13.0	12.1
	Low-dose	RMSE	0.0223	0.0250	0.0215	0.0229
		SSIM	0.612	0.556	0.620	0.586
		CNR	11.7	10.2	12.2	11.5

Table 5.2: Objective quality of the reconstructed images of the rat.

RMSE and SSIM with respect to the reference image as well as the contrast-to-noise ratio (CNR). The results of this evaluation have been summarized in Table 5.2. There is a substantial improvement in image quality metrics as a result of sinogram interpolation with the proposed algorithm. The proposed algorithm has led to much better image quality than the dictionary-based sinogram interpolation. The gain in image quality is more pronounced in the low-dose case. In fact, the objective quality of $x^{\text{proposed}}_{n_\theta/2}$ is even better than x_{n_θ} on the low-dose scan.

For a visual comparison, Figure 5.4 shows a slice from the reconstructed images of the rat from the low-dose scan with $n_\theta = 240$. There is a remarkable improvement in image quality as a result of sinogram interpolation with the proposed algorithm. The visual quality of $x^{\text{proposed}}_{n_\theta/2}$ seems to be better than both $x^{\text{dict.}}_{n_\theta/2}$ and x_{n_θ}.

Overall, our experiments show that the proposed sinogram interpolation algorithm can lead to a large improvement in the quality of the reconstructed image. Due to the additional denoising effect of both regularization functions used by the proposed algorithm, this improvement is more significant when applied on low-dose scans. This means that the proposed algorithm is especially well-suited for sinogram restoration in low-dose CT. Our other experiments, not reported here because of space limitations, also show that the proposed algorithm can be used to effectively interpolate the sinogram in more general cases, for example when the angular spacing of the missing

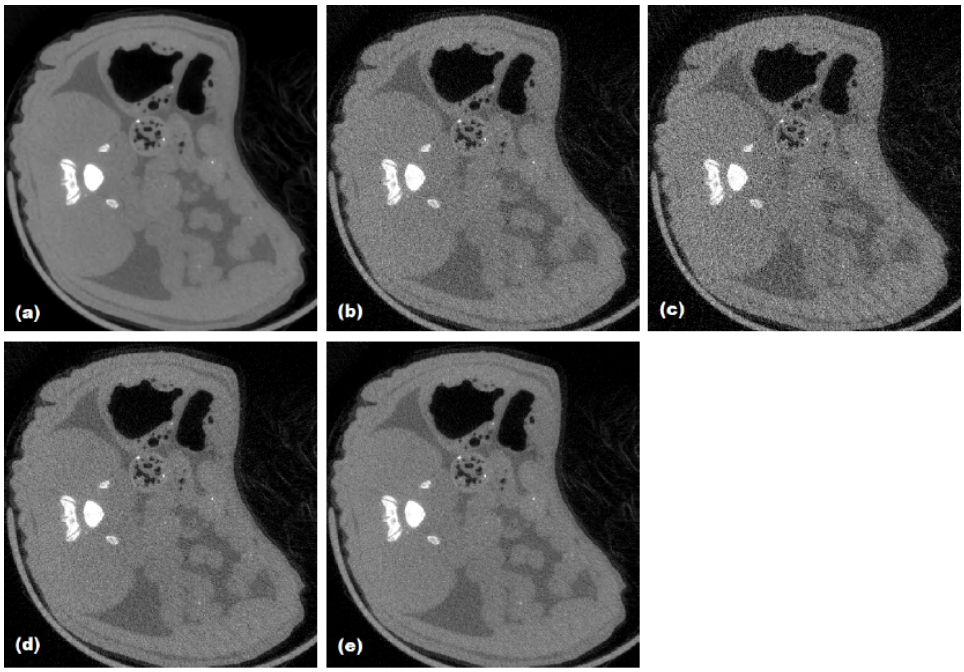

Figure 5.4: A slice in the images reconstructed from the low-dose scan of the rat with $n_\theta = 240$. (a) the reference image, (b) x_{n_θ}, (c) $x_{n_\theta/2}$, (d) $x^{\text{proposed}}_{n_\theta/2}$, (e) $x^{\text{dict.}}_{n_\theta/2}$.

projection views is non-uniform or when some detector measurements are corrupted.

Chapter 6

Reducing Streak Artifacts using Coupled Dictionaries

6.1 Introduction

As mentioned previously in this dissertation, a simple approach to reducing the radiation dose is to reduce the number of projections. Unfortunately, the images reconstructed by FBP-based methods from such few-view scans will contain large amounts of streaking artifacts. This chapter proposes a novel technique for suppressing these artifacts. Artifacts in CT may arise from different causes and would therefore have different shapes and structures [14, 231]. This chapter focuses on streak artifacts that occur in images that are reconstructed by the FBP-based methods from a small number of projections. In particular, the FDK algorithm that is commonly used for image reconstruction in cone-beam CT requires several hundred projections in order to produce artifact-free images. Reducing the number of projections can result in severe streak artifacts. This chapter proposes an algorithm that suppresses these artifacts without blurring or distorting the genuine image features.

The proposed method is based on learning two dictionaries for sparse representation of small blocks extracted from 3D CT images; one dictionary is for artifact-full images and the other is for high-quality artifact-free images. The proposed method employs a linear map that relates the representation coefficients of the artifact-full and the artifact-free blocks in these two dictionaries. This linear map is also learned from the training data. The central idea is to use the representation coefficients of small blocks in an artifact-full image to find the representation coefficients of the corresponding artifact-free blocks, thereby recovering the artifact-free image.

As mentioned in Section 2.6.3, there are studies that have used patch-based methods to suppress artifacts in CT. Some of these studies depend on the existence of a high-quality prior image of the same patient or a rich database of scans from a large number of patients [340, 341]. In many

situations, however, one does not have access to a previous scan of the same patient or a rich database of images. The method proposed in this chapter does not rely on any such prior image or database. In order to compare our algorithm with the previously-proposed algorithms, we consider the Artifact Suppressed Dictionary Learning (ASDL) method proposed in [57], which also does not require a prior image. We have briefly described ASDL in Section 2.6.3.

6.2 Methods

6.2.1 The proposed approach

The proposed algorithm learns two separate dictionaries, one for artifact-full images and another for high-quality images devoid of artifacts. These two dictionaries are denoted, respectively, by D^a (for artifact) and D^c (for clean). We also denote an artifact-full image (reconstructed from a small number of projection views) by x^a and its corresponding high-quality image (reconstructed from a very large number of projections) by x^c. In the training stage of the proposed algorithm, we extract small blocks from each of these images and stack the vectorized versions of these blocks to create two matrices, which we denote by X^a and X^c. The ith column of each of these matrices are denoted by X_i^a and X_i^c. Given an artifact-full block, X_i^a, we would like to recover its artifact-free version, X_i^c. Our approach here is to use the sparse representation of X_i^a in D^a to recover the sparse representation of X_i^c in D^c. In other words, given an artifact-full block, X_i^a, we first find the sparse representation vector Γ_i^a such that $X_i^a \cong D^a\Gamma_i^a$. From Γ_i^a, we estimate Γ_i^c, the sparse code of the corresponding artifact-free block in its dictionary. Finally, the estimate of the artifact-free block will be $\widehat{X_i^c} = D^c\Gamma_i^c$.

A very important decision is the choice of the relation between the sparse representation vectors, Γ_i^a and Γ_i^c. The simplest choice is an equality relation, i.e., $\Gamma_i^a = \Gamma_i^c$. This relation was suggested for image super-resolution in [344, 345]. However, this relation is very restrictive and a more relaxed relation can provide a greater flexibility, hopefully leading to better results. For super-resolution, for instance, as we explained in Section 2.1.3, improved results have been reported by using more relaxed relations. Therefore, in this study we use a linear relation between Γ_i^a and Γ_i^c because such a model has yielded very good results in image super-resolution, multi-modal retrieval, and cross-domain image synthesis and recognition [132, 327, 362]. In other words, we assume that the vector of representation coefficients of an artifact-

free block can be obtained from the vector of representation coefficients of the corresponding artifact-full block using the equation $\Gamma_i^c = P\Gamma_i^a$, where P is a linear map (i.e., a matrix) that is also estimated from the training data.

Throughout this chapter, N and n denote the number of projections that we use to reconstruct the artifact-free and the artifact-full images, respectively. In the experimental evaluations, $N = 720$ and $n \approx 100$. For image reconstruction we use the FDK algorithm. The goal of the proposed algorithm is to suppress the artifacts in an image reconstructed from n projections so that it looks similar to the reference image reconstructed from N projections.

Assuming for a moment that we have learned the dictionaries (D^a and D^c) and the linear map (P), we now explain how an artifact-suppressed image is produced by the proposed algorithm. The proposed algorithm is shown as a schematic in Figure 6.1. As can be seen from this figure, the algorithm starts by dividing the given set of n projections into two subsets of odd and even projections, each containing $n/2$ projections, and reconstructs two images using these two subsets of projections. The rationale behind this approach is to better exploit the correlation in the projections. The images reconstructed from each of these two subsets of $n/2$ projections will contain more artifacts than an image reconstructed from n projections. However, while the artifacts in these two images will be quite different, the genuine image features will be shared between the two images. Therefore, we expect that reconstructing two separate images using odd and even projections should result in better results in terms of the quality of the final recovered image. Our experience shows that this is indeed the case. It should be noted that with the FDK algorithm the computational cost of reconstructing two images each from $n/2$ projections is the same as that of reconstructing one image from n projections. We should also note that (as it is commonly done in this field) we work with mean-subtracted images. Later, the mean image is added back to the final reconstructed image.

From each of the two artifact-full images, we extract small overlapping blocks. In all of our experiments we used 8^3-voxel blocks with an overlap of 5 voxels in each direction between neighboring blocks. The ith pair of blocks extracted from the two artifact-full images are vectorized and stacked in tandem to form one vector, X_i^a. The sparse representation of this vector in D^a is computed such that $X_i^a \cong D^a\Gamma_i^a$. Γ_i^a is then multiplied by the matrix P to find the sparse code, Γ_i^c, of the corresponding artifact-suppressed block. Finally, the artifact-suppressed block is estimated as $\widehat{X_i^c} = D^c\Gamma_i^c$. This process is repeated for all pairs of overlapping blocks extracted from

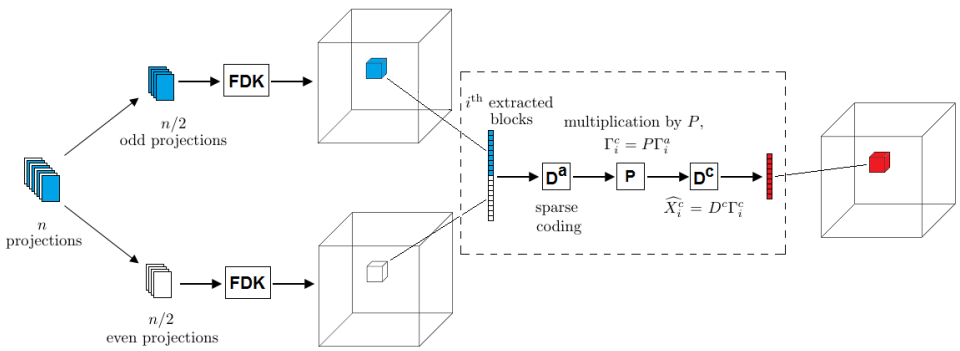

Figure 6.1: A schematic representation of the proposed algorithm. The steps shown in the dashed box are performed for all extracted overlapping blocks.

the artifact-full images. The artifact-suppressed blocks are placed in their correct locations (which is the same location where the artifact-full blocks were extracted from) in the destination image and averaged to obtain the final estimate of the artifact-suppressed image, $\widehat{x^c}$. The averaging is to take into account the fact that each voxel, except for the ones at the corners of the image, participates in more than one block because we use overlapping blocks. Mathematically, if R_i is the binary matrix that places $\widehat{X_i^c}$ in its proper location, then $\widehat{x^c}$ is computed as:

$$\widehat{x^c} = \left(\sum_{i=1}^{K} R_i \widehat{X_i^c}\right) \oslash \left(\sum_{i=1}^{K} R_i \mathbf{1}\right) \tag{6.1}$$

where K is the total number of blocks, $\oslash$ indicates element-wise division, and $\mathbf{1} \in \mathbb{R}^{512}$ is a vector of ones.

6.2.2 The dictionary learning algorithm

In the above, we assumed that the two dictionaries, D^c and D^a, and the linear map, P, were known and we described the steps taken to remove artifacts from an artifact-full image. In this section, we present algorithms for learning D^c, D^a, and P. The training data needed by the proposed algorithm includes a set of artifact-full blocks and their corresponding artifact-free blocks. To generate this data, we scan an appropriate object with a high angular sampling rate so that the scan contains a very large number (N) of projections. We reconstruct our artifact-free image using all N projections.

We then choose $n \ll N$ of these projections, divide them into odd and even projections and reconstruct two artifact-full images from each subset of $n/2$ projections. Then, we extract random blocks from the artifact-free image, vectorize and stack them as columns of a matrix X^c. From exactly the same location as the blocks used to create X^c, we extract blocks from the two artifact-full images reconstructed from $n/2$ projections and stack them together to create a matrix X^a. We refer to columns of X^a and X^c as "the training signals". As we mentioned above, each pair of blocks extracted from the same location in the artifact-full images are stacked in tandem as a single column of X^a. Therefore, X^c and X^a have the same number of columns, while each column of X^a is twice as long as each column of X^c. Specifically in our experiments, $X^c \in \mathbb{R}^{512\times K}$ and $X^a \in \mathbb{R}^{1024\times K}$, where K is the number of extracted blocks. We will provide more detail on our training and test data in Section 6.3.

Having generated our training data, X^a and X^c, we suggest to learn the dictionaries D^c and D^a and the linear map P by solving the following optimization problem, which is very similar to the formulations suggested for super-resolution and photo-sketch synthesis in [327]:

$$\begin{aligned} \underset{\{D^a,\Gamma^a,D^c,\Gamma^c,P\}}{\textbf{minimize}} \quad & \Big(\|X^a - D^a\Gamma^a\|_F^2 + \|X^c - D^c\Gamma^c\|_F^2 \\ & + \lambda^a\|\Gamma^a\|_1 + \lambda^c\|\Gamma^c\|_1 \\ & + \alpha\|\Gamma^c - P\Gamma^a\|_F^2 + \beta\|P\|_F^2 \Big) \\ \textbf{subject to:} \quad & \|D_i^a\|_2 \le 1 \quad \& \quad \|D_i^c\|_2 \le 1 \quad \forall i \end{aligned} \tag{6.2}$$

The first two terms in the objective function force the dictionaries D^a and D^c to accurately model the training signals in X^a and X^c, respectively. The third and fourth terms encourage sparsity of these representations. These four terms are reminiscent of the terms of the objective function in the basic dictionary learning algorithm. The fifth term enforces the linear relation between the vectors of sparse representation in the two dictionaries. The last term penalizes the norm of P in order to avoid overfitting and numerical instability.

The proposed objective function is not convex with respect to its five variables simultaneously. However, it is convex with respect to each of the variables if we fix the rest. Therefore, as it is common in dictionary learning, we follow an alternating minimization scheme to find a stationary point of this problem.

Updating the dictionaries D^a and D^c. With other variables being fixed, minimization with respect to the dictionary atoms can be written as the following two identical optimization problems.

$$\begin{aligned} &\underset{D^a}{\textbf{minimize}}\ \|X^a - D^a\Gamma^a\|_F^2 \quad \textbf{subject to:} \quad \|D_i^a\|_2 \leq 1 \quad \forall i \\ &\underset{D^c}{\textbf{minimize}}\ \|X^c - D^c\Gamma^c\|_F^2 \quad \textbf{subject to:} \quad \|D_i^c\|_2 \leq 1 \quad \forall i \end{aligned} \tag{6.3}$$

We use the efficient implementation of the K-SVD algorithm proposed in [275] to solve these problems.

Updating the sparse representation matrices Γ^a and Γ^c. With all other variables fixed, minimization with respect to Γ^a can be simplified as follows:

$$\begin{aligned} &\underset{\Gamma^a}{\textbf{minimize}}\ \|X^a - D^a\Gamma^a\|_F^2 + \alpha\|\Gamma^c - P\Gamma^a\|_F^2 + \lambda^a\|\Gamma^a\|_1 \\ \equiv &\underset{\Gamma^a}{\textbf{minimize}}\ \left\| \begin{bmatrix} X^a \\ \sqrt{\alpha}\Gamma^c \end{bmatrix} - \begin{bmatrix} D^a \\ \sqrt{\alpha}P \end{bmatrix} \Gamma^a \right\|_F^2 + \lambda^a\|\Gamma^a\|_1 \end{aligned} \tag{6.4}$$

Similarly, minimization with respect to Γ^c can be written as:

$$\begin{aligned} &\underset{\Gamma^c}{\textbf{minimize}}\ \|X^c - D^c\Gamma^c\|_F^2 + \alpha\|\Gamma^c - P\Gamma^a\|_F^2 + \lambda^c\|\Gamma^c\|_1 \\ \equiv &\underset{\Gamma^c}{\textbf{minimize}}\ \left\| \begin{bmatrix} X^c \\ \sqrt{\alpha}P\Gamma^a \end{bmatrix} - \begin{bmatrix} D^c \\ \sqrt{\alpha}I \end{bmatrix} \Gamma^c \right\|_F^2 + \lambda^c\|\Gamma^c\|_1 \end{aligned} \tag{6.5}$$

where I is the identity matrix of the right size.

The optimization problems in (6.4) and (6.5) are simple sparse coding problems. In our experience, greedy algorithms such as the orthogonal matching pursuit (OMP) [252] with a sparsity constraint work very well on these problems. Therefore, we used OMP with sparsity constraints $\|\Gamma^c\|_0 \leq T^c$ and $\|\Gamma^a\|_0 \leq T^a$ to solve these problems. In other words, we replace the ℓ_1-norm penalty with ℓ_0-norm constraints. We choose $T^c = 16$ for artifact-free blocks as suggested in the context of denoising in [274]. For artifact-full blocks we choose a higher sparsity level of $T^a = 20$ because the signals are longer and include artifacts too. This approach will also eliminate the need to tune the parameters λ^a and λ^c.

Updating the linear map P. Minimization with respect to P involves solving the following problem:

$$\underset{P}{\textbf{minimize}}\ \alpha||\Gamma^c - P\Gamma^a||_F^2 + \beta||P||_F^2 \tag{6.6}$$

which has the following closed-form solution:

$$P^* = \Gamma^c(\Gamma^a)^T\left(\Gamma^a(\Gamma^a)^T + \frac{\beta}{\alpha}I\right)^{-1} \tag{6.7}$$

Initialization and regularization parameter selection. We initialize D^a and D^c to the dictionaries learned with the basic dictionary-learning scheme, Equation (2.1), from artifact-full and artifact-free images, respectively. We used the K-SVD algorithm [2] to learn these initial dictionaries. Of course, for D^a we need to concatenate two such dictionaries to obtain the right size. This initialization will help our learning algorithm converge much faster than if we use an overcomplete DCT or wavelet basis as the initial dictionary. The number of atoms in each dictionary was chosen to be 1024. We initialize P to the identity matrix. Using OMP, we compute the sparse representations of the training signals X^a and X^c in the initial dictionaries D^a and D^c, respectively, and use them as the initial values of Γ^a and Γ^c. As mentioned above, our training strategy does not require knowing the values of λ^a and λ^c. We used $\alpha = \beta = 0.1$ for all experiments reported in this chapter. We found empirically that these values work very well in all our experiments. As mentioned above, the regularization term $\|P\|_F^2$ is meant to avoid numerical instability. As suggested in [69], if the amount of training data is not too small, numerical instabilities are unlikely to occur and β can be set to a very small number.

After the above initializations, we alternately minimize the objective function with respect to the five variables. For the stopping criterion, one can adopt a cross-validation approach. Specifically, in this approach part of the training data (X^a and X^c) can be set aside as the validation data. At the end of each iteration of the learning algorithm, the learned parameters (D^a, D^c, and P) are applied to reconstruct the artifact-free blocks in the validation data set from their corresponding artifact-full blocks using our algorithm as shown in Figure 6.1. The learning is stopped when an acceptable level of accuracy in reconstructing the artifact-free blocks in the validation data set is achieved. We should also mention that, instead of a straightforward application of the algorithm for reconstruction as shown in Figure 6.1, one can optimize the coefficients Γ^a and Γ^c simultaneously. The opti-

mization steps will be similar to those presented for training above, except that the parameters will be fixed and only Γ^a, Γ^c, and X^c are optimized. Alternatively, one can stop the learning algorithm after a pre-specified number of iterations, which is also very common in dictionary learning [2, 212]. In the experiments reported in this chapter we stopped the learning algorithm after 50 iterations, which we have found empirically to be sufficient.

6.3 Evaluation

The first scan consisted of $N = 720$ projections of a rat between $0°$ and $360°$ at $0.5°$ intervals. We used the full set of 720 projections to reconstruct a very high-quality image of the rat. For this purpose, we first reconstructed an image using the FDK algorithm, followed by 50 iterations of the MFISTA algorithm. The resulting image, which we refer to as the reference image, had a very high quality with no visible artifacts of any kind. To evaluate our proposed algorithm, we used a subset of $n = 120$ projections from this scan, i.e., projections at $3°$ intervals. As we mentioned above and showed in Figure 6.1, the proposed algorithm divides the projections into two halves, each containing $n/2 = 60$ projections (odd and even projections), and reconstructs separate images from each set of 60 projections using the FDK algorithm.

In this first experiment, we generated the training and test data from the same images. Each reconstructed image was $880 \times 880 \times 650$ voxels. We divided each of the images into two halves (i.e., into two $880 \times 880 \times 325$-voxel images). We used one half for training, i.e., for learning the dictionaries D^a and D^c and the matrix P. Then, we applied our method as shown in Figure 6.1 to suppress the artifacts in the other half of the image.

Figure 6.2 shows the effect of applying the proposed algorithm for artifact suppression. In this figure, we have shown a typical slice of the reference image (reconstructed from 720 projections) and artifact-full image (reconstructed from 120 projections), alongside the result of applying our algorithm and ASDL with 120 projections. For a comparison, we have also included the same slice in the image reconstructed with the FDK algorithm from 240 projections. We have also shown, in part (f) of this figure, the image obtained with total-variation (TV) denoising. We used Chambolle's famous algorithm for TV denoising.

The whole slice in Figure 6.2 is shown using a window of attenuation coefficient (μ-window) of $[0, 0.40]$. For a better visual comparison, we have selected two regions of interest (ROIs) and have displayed them with narrower

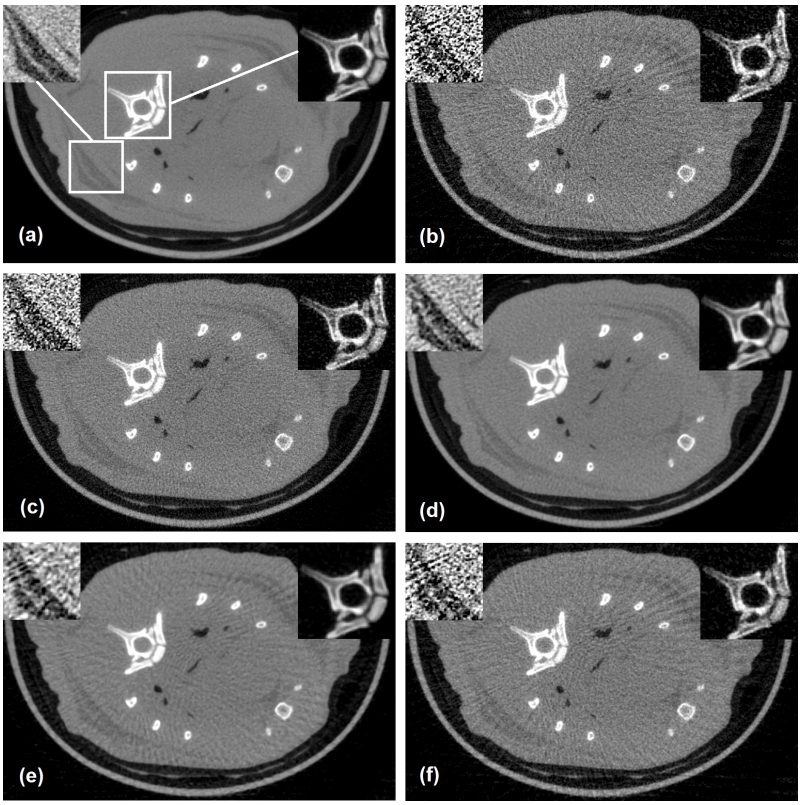

Figure 6.2: A typical slice of the images reconstructed from the first rat scan. (a) the reference image reconstructed from 720 projections, (b) FDK-reconstructed from 120 projections, (c) FDK-reconstructed from 240 projections, (d) the image produced from 120 projections using our proposed algorithm, (e) the image reconstructed from 120 projections using ASDL, (f) FDK-reconstructed from 120 projections followed by TV denoising. Two ROIs are shown magnified on the top-left and top-right of the images. The locations of these ROIs have been marked on the slice of the reference image.

μ-windows. One of these ROIs (shown on the top right of each image) contains bone surrounded with soft tissue. We display this ROI with a μ-window of $[0.2, 0.5]$. The second ROI contains fat surrounded with soft tissue, which we display with a μ-window of $[0.14, 0.23]$. Both ROIs are displayed with a magnification of 150%. This figure shows a marked improvement in the quality of the reconstructed image as a result of applying the proposed algorithm. The image reconstructed with the proposed algorithm (from 120 projections) has a much higher quality than the FDK-reconstructed image from 120 projections and appears to be better than the FDK-reconstructed image from 240 projections also. ASDL seems to have removed most of the noise in the image without suppressing much of the artifact. The image obtained via TV denoising has a low quality and all of the streak artifacts have remained, even though much of the noise has been removed. This is an expected result because denoising algorithms cannot distinguish between true image features and streak artifacts. The TV denoising method, for instance, is based on the prior assumption that the image has a sparse gradient, encouraging piece-wise constant solutions. Although this approach is effective for removing random noise, it cannot distinguish strong artifacts from genuine image features.

For a more meaningful evaluation, we applied the dictionary and linear map learned in the experiment described above to suppress the artifacts in the image of a different rat. Similar to the above experiment, in this new experiment the scan consisted of 720 projections. A reference artifact-free image was reconstructed using all 720 projections. The proposed algorithm and ASDL were applied to produce artifact-suppressed images from 120 projections. Figure 6.3 shows a typical slice in the images from this experiment. We have included the two intermediate FDK-reconstructed images (from 60 even and odd projections) that were used in the proposed algorithm in parts (g) and (h) of this figure. However, the image produced by the proposed algorithm should be compared with the FDK-reconstructed image from 120 projections, shown in part (b) of this figure. This figure shows that the proposed algorithm has successfully reduced the artifacts, significantly improving the image quality. This is more visible in the two ROIs that are re-displayed with narrow μ-windows. The μ-window for the whole slice is $[0, 0.40]$ and the μ-windows for the ROIs shown on the top-left and bottom-right are $[0.16, 0.21]$ and $[0.2, 0.45]$, respectively. The proposed algorithm has also produced an image that is markedly better than the image produced by ASDL and slightly better than the FDK-reconstructed image from 240 projections. ASDL has not been very successful in reducing the artifacts in this experiment.

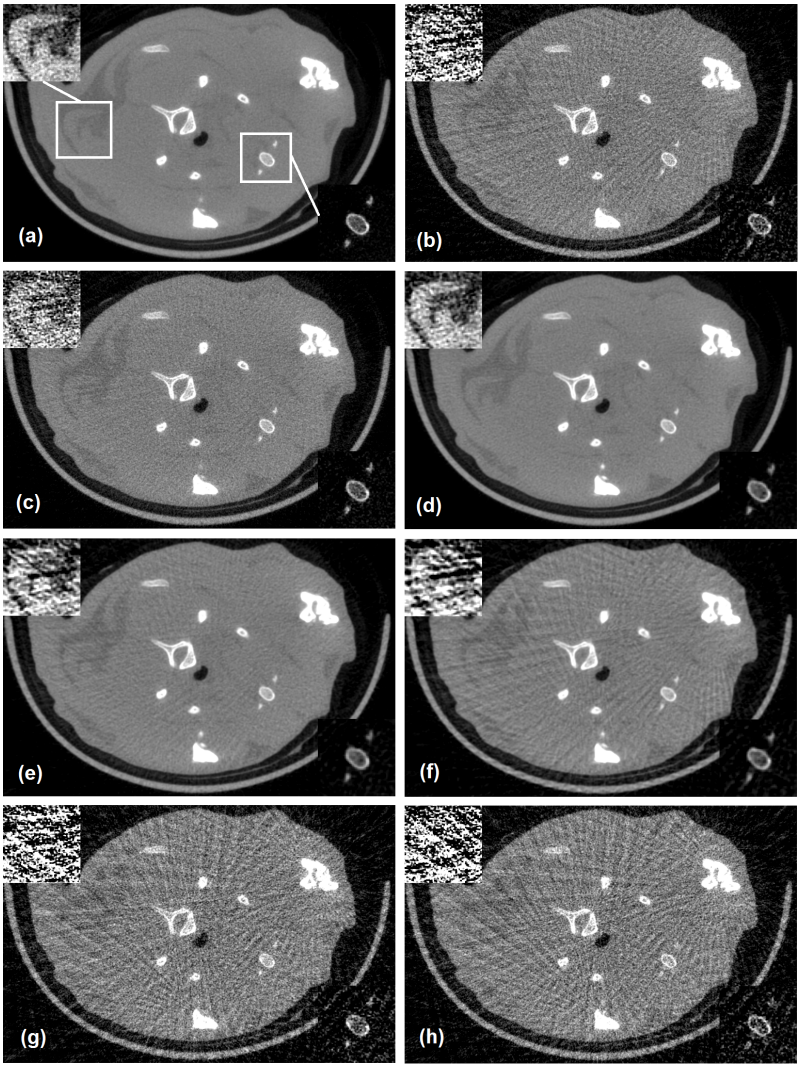

Figure 6.3: A slice from the reconstructed images of the second rat. (a) the reference image, (b) FDK-reconstructed from 120 projections, (c) FDK-reconstructed from 240 projections, (d) produced by the proposed algorithm from 120 projections, (e) produced by ASDL from 120 projections, (f) FDK-reconstructed image from 120 projections followed by K-SVD denoising, (g)-(h) the intermediate images in the proposed algorithm (each reconstructed with the FDK algorithm from 60 projections). The locations of the two ROIs have been marked on the slice of the reference image.

One of the reasons for the improved quality of the images produced by the proposed algorithm is the noise reduction. Even though we designed our algorithm for artifact reduction, it also leads to an automatic noise reduction, which is a free extra benefit. The reason for this automatic denoising is that the output image of the proposed algorithm is created as a sparse representation of blocks in the dictionary D^c. This is a successful denoising strategy, as we explained in Section 2.1.3. Therefore, one may question whether the improvement in the image quality by the proposed algorithm is mostly due to its denoising effect, and not artifact-suppression as we have claimed. To show that this is not the case, i.e., that our proposed algorithm is indeed an artifact-suppression algorithm, we applied the K-SVD denoising algorithm [100] to remove the noise in the image reconstructed with the FDK algorithm from 120 projections. The resulting image is shown in Figure 6.3(f). It is quite clear that dictionary-based denoising only removes the noise and leaves the artifacts untouched. Therefore, our proposed algorithm does indeed accomplish much more than a dictionary-based denoising algorithm and is a true artifact-suppression algorithm.

For a more objective evaluation of the proposed algorithm, we compute the root-mean-square of the error (RMSE), where we define the error as the difference between the reconstructed image and the reference image, and the structural similarity index (SSIM) between the reconstructed image and the reference image. The results are presented in Table 6.1. The proposed algorithm has clearly outperformed ASDL in terms of RMSE and SSIM. Moreover, compared with the FDK-reconstructed image from 240 projections, the artifact-suppressed image produced by our proposed algorithm from 120 projections is closer to the reference image in terms of both RMSE and SSIM.

	FDK-120	FDK-240	FDK-120 with K-SVD denoising	Proposed algorithm - 120	ASDL - 120
RMSE	0.0206	0.0119	0.0165	0.0104	0.0147
SSIM	0.653	0.856	0.702	0.860	0.764

Table 6.1: RMSE and SSIM for the FDK-reconstructed images of the second rat from 120 and 240 projections, FDK-reconstructed from 120 projections followed by denoising using the K-SVD algorithm, and the images produced from 120 projections by the proposed algorithm and ASDL.

In order to further evaluate the performance of the proposed algorithm,

we applied it on a scan of the physical phantom. In order to test the generalizability of the learned parameters (D^a, D^c, and P), we used those parameters that were learned from the rat scan described above to suppress the artifacts in this experiment with the phantom. Similar to the scans of the rat, the phantom scan consisted of 720 projections. All of the 720 projections were used to reconstruct a high-quality reference image. The proposed algorithm was then used to produce an artifact-suppressed image from 120 projections.

In Figure 6.4 we show parts of two selected slices in the image of the phantom. The μ-window used for displaying these figures is $[0, 0.50]$. It is clear from this figure that the proposed algorithm has significantly reduced the artifacts without degrading genuine image features. The right column in this figure shows two of the resolution coils that have been included in this phantom for the purpose of visual inspection of the spatial resolution. In the FDK-reconstructed image from 120 projections, and to a lower degree also in the FDK-reconstructed image from 240 projections, these coils have given rise to ring-shape artifacts. While the proposed algorithm has substantially reduced these and other artifacts, it has not blurred or degraded the fine image features but seems to have improved them too. Compared with our proposed algorithm, ASDL has been much less effective in removing these artifacts. On the other hand, TV-based denoising seems to have had no effect on the artifacts, although it has managed to reduce the noise.

Table 6.2 presents a quantitative evaluation of the quality of the images reconstructed by different algorithms in this experiment. In addition to RMSE and SSIM, we estimated the modulation transfer function (MTF) using an approach similar to that in Chapters 3 and 4. The values reported in Table 6.2 as spatial resolution are the spatial frequencies at which the normalized MTF reached 0.1 in different images.

	Reference image	FDK-120	FDK-240	Proposed algorithm-120	ASDL-120
RMSE	0.0	0.022	0.010	0.0094	0.0137
SSIM	1.0	0.623	0.892	0.900	0.812
S.R. (mm^{-1})	4.39	3.66	3.98	3.98	3.76

Table 6.2: Quantitative comparison of the quality of the images of the physical phantom reconstructed with the FDK algorithm from 120 and 240 projections and the artifact-suppressed images produced using the proposed algorithm and ASDL from 120 projections.

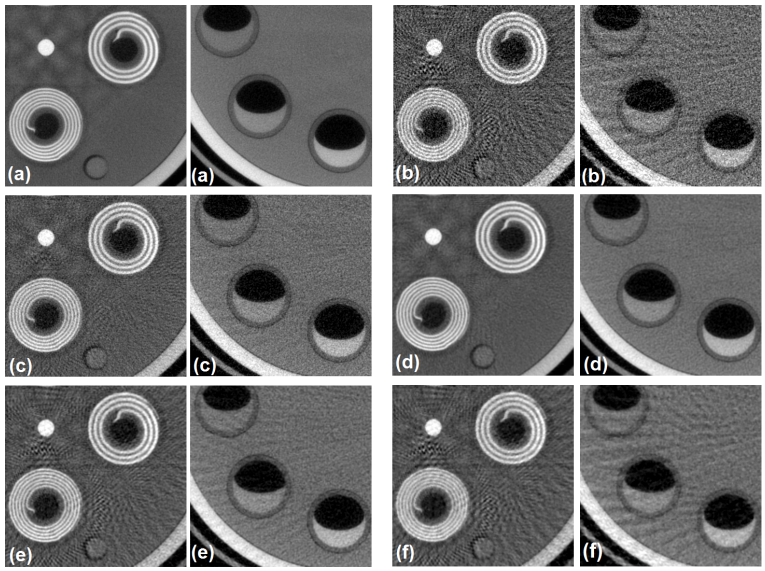

Figure 6.4: Parts of two typical slices in the image of the phantom; (a) the reference image, (b) FDK-reconstructed from 120 projections, (c) FDK-reconstructed from 240 projections, (d) produced by the proposed algorithm from 120 projections, (e) produced by ASDL from 120 projections, (f) FDK-reconstructed from 120 projections followed by TV denoising.

As we mentioned above, the results shown in Figure 6.4 and Table 6.2 were obtained by using the parameters (i.e., D^a, D^c, and P) from the rat experiment. Even though our results show that these parameters work very well on the phantom image, one may wonder if we could have obtained better results by learning a new set of parameters from a similar image. In order to determine if learning a new set of parameters from a similar image can improve the results, we performed a new experiment. In this experiment, the physical phantom was scanned one more time and new dictionaries (D^a and D^c) and a new matrix (P) were learned. We then applied our algorithm with these parameters on the scan of the phantom described above. The results of this experiment were very close to the results shown in Figure 6.4 and Table 6.2. In particular, the values of SSIM and spatial resolution shown for the proposed algorithm in Table 6.2 increased slightly to 0.905 and 4.00, while the value of RMSE remained almost the same. In our opinion, this means that the parameters learned from one image can be used when applying the algorithm on a very different image, even though slightly better results may be obtained if the parameters are

learned from a similar image. This is because our proposed algorithm is based on relating the sparse representations of artifact-full blocks with their corresponding artifact-free blocks. Because of the small size of the blocks, the learned parameters of the algorithm depend on the local shapes of these artifacts, which does not depend much on the shape of large-scale features of the image.

The above results indicate that the dictionaries and the linear map learned from one image are applicable to another image. However, in the above experiments we used the same number of projections (i.e., 120) for both the rat and the phantom. It is well known that artifacts become stronger and their angles change as the number of projections is reduced. Therefore, parameters learned from one image may not work well when used for artifact suppression in an image reconstructed from a different number of projections. Moreover, as the number of projections decreases, it will be more difficult to suppress the artifacts because they become much stronger. Even though our algorithm was able to effectively suppress the artifacts in the images reconstructed from 120 projections, it is expected that its performance should decrease as the number of projections is reduced further. Therefore, we face two important questions: 1) What is the limit of performance of the proposed algorithm in terms of the number of projections required? and 2) Is it possible to use the parameters learned from an experiment with n_1 projections for artifact suppression with a different number of projections, $n_2 \neq n_1$?

In order to answer these questions, we used the parameters learned from the scan of the first rat to suppress the artifacts in the image reconstructed from the scan of the second rat, but this time we used $n = 90$ projections. In other words, D^a, D^c, and P are learned on artifact-full images reconstructed from 120 projections, and then they are applied to suppress the artifacts in images reconstructed from 90 projections. The results of this experiment are shown in Figure 6.5 and Table 6.3 and denoted as "Proposed Algorithm - train:120 - test:90". The μ-windows used for Figure 6.5 are the same as those for Figure 6.3. For comparison, we also learned new dictionaries and a linear map, this time from images reconstructed from 90 projections of the scan of the first rat and used them to suppress the artifacts in the images reconstructed from 90 projections of the scan of the second rat. The results are also shown in Figure 6.5 and Table 6.3, denoted as "Proposed Algorithm - train:90 - test:90".

There are two main conclusions that can be drawn as answers to the two questions that we posed above. First, Figure 6.5 and Table 6.3 show a marked improvement in "train:90 - test:90" compared with "train:120 -

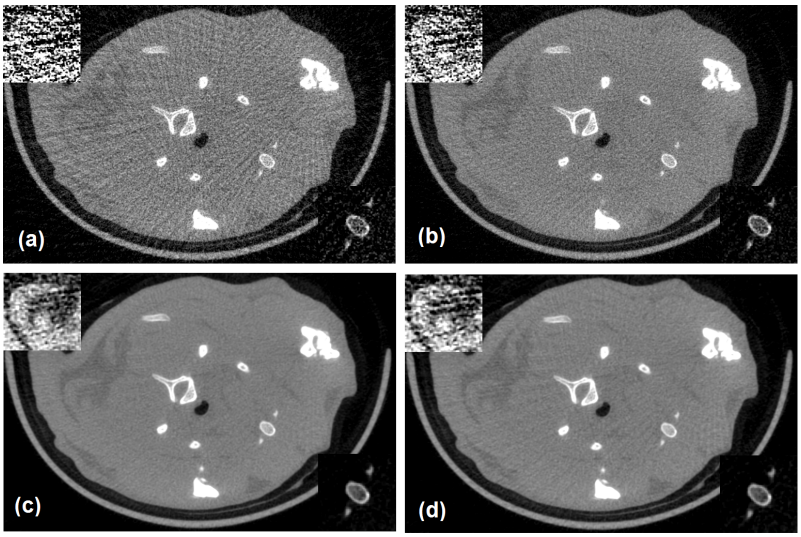

Figure 6.5: A slice from the reconstructed image of the second rat; (a) FDK-reconstructed from 90 projections, (b) FDK-reconstructed from 180 projections, (c) produced by the proposed algorithm from 90 projections using the parameters learned on the image of the first rat with 90 projections, (d) produced by the proposed algorithm from 90 projections using the parameters learned on the image of the first rat with 120 projections. (The reference image for this figure is the same as Figure 6.3 (a)).

test:90". Therefore, parameters learned for artifact reduction by the proposed algorithm are no longer optimal if the number of projections changes significantly. We think this is because the shape and the strength of the artifacts change significantly when the number of projections changes significantly. Secondly, the performance of our proposed artifact-suppression algorithm was reduced compared to our experiments with 120 projections that we reported earlier in this chapter. Figure 6.5(c), which corresponds to "train:90 - test:90" is still better than the FDK-reconstructed image with twice the number of projections shown in Figure 6.5(b). However, by comparing these results with those shown in Figures 6.3 and 6.4 and Tables 6.1 and 6.2, we see that the gain in the image quality is reduced. This is what we should expect because as the number of projections is reduced, the artifacts dominate the genuine image details and it should thus be more difficult for the algorithm to remove them.

As we mentioned in Section 2.1.3, for image super-resolution and other

	FDK-90	FDK-180	Proposed Algorithm - train:120 - test:90	Proposed Algorithm - train:90 - test:90
RMSE	0.0259	0.0167	0.0172	0.0160
SSIM	0.531	0.730	0.705	0.741

Table 6.3: Comparison between the quality of the images of the second rat reconstructed using the FDK algorithm with 90 and 180 projections and the artifact-suppressed images produced by the proposed algorithm with 90 projections.

applications, various different models have been suggested for relating the sparse representation coefficients of the source and the target images. Similarly, many different models can be used for relating the sparse representations of artifact-full and artifact-free blocks. An investigation of all possible models is beyond the limitations of this dissertation. However, one natural question is whether a simpler model than the linear model used in the proposed algorithm would work equally well. To answer this question, we applied our algorithm on the rat data by assuming that the sparse representation coefficients are identical. This is the same as the assumption used for image super-resolution in [345]. In terms of the notation used in this chapter, this approach means setting $P = I$ where I is the identity matrix and $\Gamma^c = \Gamma^a$. The result of this experiment is presented in Figure 6.6. For comparison, we have also shown the slice of the image reconstructed by the proposed algorithm (with linear mapping). It is clear that this simplified model is much less effective than the proposed model with the linear mapping. In fact, in terms of the RMSE and SSIM, the image produced by this simplified model is slightly worse than the image produced by ASDL.

6.4 Discussion

Our results show that the proposed algorithm is very effective in suppressing the streak artifacts that appear in CT images reconstructed from approximately 100 projections. In all of our experiments, the image reconstructed by the proposed algorithm had a better quality than the image reconstructed with the FDK algorithm from twice as many projections. The streak artifacts were largely removed in the images produced from 120 projections by the proposed algorithm. There was also no visible blurring or distortion

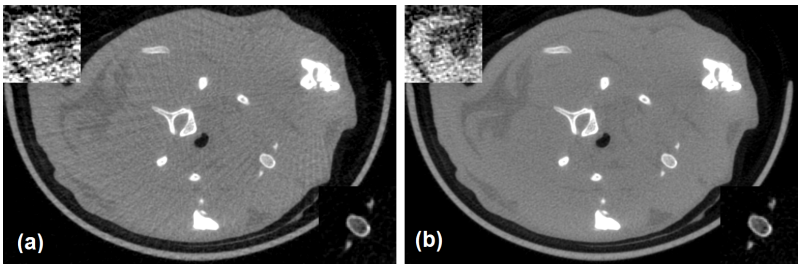

Figure 6.6: (a) Image reconstructed by assuming an identity relation between representation coefficients of artifact-full and artifact-free blocks. (b) Image reconstructed by assuming a linear relation; this image is the same as that in Figure 6.3(d).

of the true image features by the proposed algorithm. In addition to suppressing the artifacts, the proposed algorithm effectively reduces the noise also.

An important and promising behavior of the proposed algorithm is that its parameters, i.e., the two dictionaries and the linear map, do not have to be learned for every image, as long as the numbers of projections used in the training and reconstruction stages are close. This is a very valuable property because it means a substantial saving in the time and effort needed for training. This behavior is due to the design of the algorithm. Specifically, the algorithm is based on relating the sparse representations of small blocks of the artifact-full and artifact-free images in the learned dictionaries. The local structure of these artifacts does depend on the number of projections used to reconstruct the artifact-full images, but it is, to a large degree, independent of the shape of large image features. Therefore, the parameters learned from one set of training images work well when the algorithm is applied on a very different image as long as the number of projections does not change much.

The theoretical underpinnings of the dictionary learning algorithms have not quite matured yet. In fact, a complete theoretical analysis of dictionary learning is still considered as an open problem [98]. Nonetheless, in recent years these algorithms have been used in hundreds or thousands of studies and have proven to be robust and reliable. The performance of these learned dictionaries for different image processing tasks may depend on factors such as the amount of training data and the scale and structure of features in the image. These factors may also influence the performance of the algorithm proposed in this study, but a detailed investigation of these factors is beyond

the scope of this dissertation. Nonetheless, such studies can be very helpful for the proper application of learned dictionaries in CT. We are aware of only one such study [300], in which the effect of the scale and orientation of features in the training data on the performance of dictionary-based iterative CT reconstruction were analyzed. Similar studies can be very helpful for better implementation of artifact-removal algorithms such as the algorithm proposed in this chapter.

Our experience shows that the performance of the proposed algorithm deteriorates when the number of projections is much less than 100. For example, we have tried applying our algorithm with 70 projections with very little success. Even though the proposed algorithm still suppresses some of the artifacts when the number of projections is around 70, the genuine image detail are distorted. This is simply because the artifacts are very strong and overshadow the image features. Overall, our experience shows that the gain in image quality obtained by our proposed algorithm is most significant when the number of projections used is approximately between 100 and 200.

There is also a slight blurring of a few of the image features in some of the images reconstructed by the proposed algorithm. An example can be seen in the bone structure shown in the top-right ROI in Figure 6.2. This blurring can be reduced by increasing the number of atoms of the artifact-free dictionary that are used in building each of the blocks of the reconstructed image. In the proposed algorithm, this is controlled by parameter T^c. As we mentioned earlier in this chapter, we used $T^c = 16$. We chose this value because it was found to be a very good choice in the denoising of 3D CT images in [274]. One can choose a larger T^c to reduce the blurring. Increasing the number of dictionary atoms that participate in the representation of the image blocks will improve the reconstruction of image features, thereby reducing the blurring. However, increasing the number of atoms may also reduce the denoising effect of the proposed algorithm because some of the added atoms will model the noise. The right value of T^c will ultimately depend on the desired trade-off between noise and spatial resolution. Another approach for reducing the blurring is to increase the block overlap. In almost all applications of dictionary-based image processing, better results are obtained by increasing the overlap between adjacent extracted blocks. The downside of this approach is increased computation.

As we mentioned in the Introduction section, an important consideration in learning and usage of overcomplete dictionaries is the computational time. In most applications, the learning of the dictionary is more computationally intensive than its later usage. This is also the case in the proposed algorithm.

In our proposed learning algorithm, the most computationally expensive part is the sparse coding steps required for updating Γ^a and Γ^c (Equations (6.4) and (6.5)). As we mentioned, we used the OMP algorithm for solving these equations. Given a dictionary D and a signal z, each iteration of OMP selects a new atom from D and projects z onto the space of columns selected so far. If we denote the sub-dictionary that contains the set of columns that have been selected up to the current iteration by D_S, the bottleneck of the OMP algorithm is the computation of the pseudo-inverse of D_S. In a straight-forward implementation, this pseudo-inverse must be computed at every step of OMP, i.e., every time a new dictionary atom is selected. This computation can be avoided by using progressive Cholesky or QR update strategy instead of explicit matrix inversion [30, 68]. Even faster implementation is possible by avoiding explicit computation of the signal residual at each iteration. This approach, which has been named Batch-OMP in [275], can result in very large speed-ups when the number of training signals is large. Of course, OMP is not the only tool for solving Equations (6.4) and (6.5) and there are other alternative algorithms that have been suggested for fast sparse coding of large numbers of signals [122]. The second most computationally expensive step is the dictionary update in Equation (6.3). In a straight-forward implementation of the K-SVD algorithm, the main computational burden is associated with the SVD decomposition of the error matrix. The implementation suggested in [275] avoids an explicit SVD computation, making the update substantially faster. As suggested in [274] even the computation of the error matrix is not necessary, making the algorithm faster. The speedup strategies mentioned above have been implemented in publicly-available software and they have been applied for CT denoising in [15]. The last step in the proposed training algorithm (update of P through Equation (6.7)) is computationally negligible. The required number of iterations of the dictionary learning algorithm strongly depends on the initialization. If some parameters (D^a, D^c, and P) have already been learned on a different dataset, they can be used to substantially reduce the required number of iterations of the learning algorithm. The time required for the reconstruction of the final image from the artifact-full images (once the algorithm parameters have been learned) is proportional to the image size. For example, with our Matlab implementation, an image of size $200 \times 200 \times 100$ voxels can be reconstructed in less than 15 minutes.

There are ways in which the performance of the proposed algorithm may be improved. One of these ways, as we have mentioned previously in this chapter, is to learn the algorithm parameters (i.e., the dictionary and the linear map) from a training dataset that is similar to the image that we

want to reconstruct. We have also observed that increasing the dictionary size slightly improves the quality of the reconstructed images. Increasing the number of atoms to 2048 (from 1024 used in the experiments reported here) slightly improved the results. However, increasing the dictionary size also increases the computational load. Another simple approach to improving the algorithm performance is to increase the amount of overlap between the neighboring blocks. As we mentioned above, in all of the experiments reported in this chapter we used an overlap of 5 voxels in each direction. Increasing the block overlap always improves the quality of the reconstructed image at the cost of increased computational load. In patch-based processing of 2D images it is common to use the maximum overlap (so that the neighboring patches are shifted by only one pixel). However, for large 3D images this may result in excessive computational load. For blocks of size 8^3 voxels used in this study, increasing the voxel overlap from 5 to the maximum possible overlap of 7 will increase the number of blocks by a factor of 27. However, in our experience this also improves the performance of the proposed algorithm in terms of both artifact suppression and, especially, denoising. Finally, it may also be possible to achieve improved image quality by using a more flexible model and more training data. For instance, a more general model was proposed for image super-resolution, cross-view action recognition, and sketch-to-photo face recognition in [132]. This model includes two linear maps and has been shown to be successful in various tasks. Training of more general models requires larger training data and longer training times. More general models may also include additional parameters that need tuning. Nevertheless, when properly trained, they may lead to improved results.

In recent years, several "iterative" dictionary-based CT reconstruction algorithms have been proposed [11, 197, 301, 305, 339]. In general, these algorithms have shown promising results. Therefore, an interesting question is whether the algorithm proposed in this chapter can be turned into an iterative reconstruction algorithm. A first step towards this transformation is to introduce a measurement misfit term into our objective function. Assuming that the algorithm parameters have been learned, a potential approach to recovering a high-quality image is by solving the following optimization

problem:

$$\underset{\{x,\Gamma^a,\Gamma^c\}}{\textbf{minimize}} \quad \Bigg(\lambda_{\text{data}}\|Ax-y\|_W^2 + \sum_i \Big\{\|R_i^c x - D^c\Gamma_i^c\|_2^2 + \lambda^c\|\Gamma_i^c\|_1 + \|R_i^a x^a - D^a\Gamma_i^a\|_F^2 + \lambda^a\|\Gamma_i^a\|_1\Big\} + \alpha\|\Gamma^c - P\Gamma^a\|_F^2\Bigg) \tag{6.8}$$

In this equation, A is the projection matrix, y represents the sinogram measurements, λ_{data} is the regularization parameter for the measurement misfit term, and $||.||_W^2$ is the weighted ℓ_2-norm that is preferred in CT reconstruction because the noise in y is signal-dependent. R_i^c is a binary matrix that extracts the ith block from the image x, so that $R_i^c x$ is the vectorized block. The same comment applies to $R_i^a x^a$, except that, as we explained earlier, two artifact-full images are reconstructed from odd and even projections. However, here we have used the simple notation of $R_i^a x^a$ to avid a cluttered equation. The artifact-full image x^a is reconstructed using the FDK algorithm and the high-quality image is obtained as the solution of the above optimization problem.

The optimization problem in (6.8) can be solved using alternating minimization. Minimization with respect to Γ^a and Γ^c will be similar to those shown in Equations (6.4) and (6.5). Minimization with respect to x can be performed, for example, using the separable paraboloid surrogate method [101]. This is the method used in the iterative dictionary-based reconstruction method proposed in [339]. One problem with this approach is that it requires access to individual elements of the projection matrix A. For large 3D images this is impractical because the matrix A is too large to be saved in the computer memory and with standard implementations of this matrix it is not easy to access individual elements [152]. Another approach to solving the minimization with respect to x is to use the conjugate gradient method, as suggested in [305]. Another potential challenge in using the iterative reconstruction method proposed in Equation (6.8) is the choice of the regularization parameter. In general, the choice of the regularization parameter is one of the critical aspects of regularized inverse problems [8]. In most dictionary-based iterative reconstruction algorithms that have been proposed for CT, this issue has not been properly addressed and the regularization parameter has been selected empirically [305, 308, 339].

Chapter 7

Two-Level Dictionary for Fast CT Image Denoising and Restoration

7.1 Introduction

As explained in Section 2.1.3, one of the major disadvantages of learned overcomplete dictionaries is that they are much more computationally costly than analytical dictionaries. Specifically, obtaining the sparse representation of a signal in these overcomplete and unstructured dictionaries requires solving an optimization problem. Denoting the dictionary with D, the sparse representation γ of a signal x in D will require solving:

$$\text{minimize} \, \|\gamma\|_0 \quad \text{subject to: } \|x - D\gamma\|_2^2 \leq \epsilon \tag{7.1}$$

where ϵ depends on the noise variance. This problem is typically solved by either using a greedy method such as the orthogonal matching pursuit (OMP) or by a convex relaxation of the ℓ_0 norm to ℓ_1 norm and using methods such as the basis pursuit [317–319].

The processing of large 3D images, in particular, is computationally highly intensive because the number and dimensionality of blocks are very high. Greedy methods, which are the focus of this chapter, may yield very sub-optimal results because they choose the dictionary atoms one at a time.

In this chapter, we propose a structured dictionary for sparse representation of large signals. The proposed dictionary structure will speed up the sparse coding by allowing multiple atoms to be selected in each iteration. Moreover, by structuring the dictionary atoms into clusters, the proposed dictionary structure will enable us to learn and effectively use a larger number of atoms, increasing the expressive power of the dictionary. In summary, the proposed dictionary has two levels. The first level consists of an off-the-shelf orthonormal basis while the second level consists of learned atoms that are adapted to the signal class of interest. The signal is first decomposed in

the first-level dictionary. Since the first-level dictionary is orthonormal, this decomposition can be computed very efficiently. The decomposition in the first-level dictionary is used to find the sparse representation of the signal in the second-level dictionary, which consists of learned atoms. Therefore, the proposed dictionary structure aims at combining the speed of analytical dictionaries with the flexibility and representational power of overcomplete dictionaries.

We will apply the proposed dictionary structure for removing noise and ring artifacts from CT images. Unlike the streak artifact that we considered in Chapter 6, the noise and ring artifacts considered in this chapter have a very different shape than the genuine image features. Therefore, our approach to removing noise and ring artifacts in this chapter follows the basic dictionary-based denoising method that we described in Section 2.1.3. In other words, we assume that the true image features will have a sparse representation in dictionaries trained on clean artifact-free images, whereas noise and ring artifacts will not have a sparse representation in such dictionaries. Therefore, sparse estimation of image blocks in a dictionary learned from clean images should lead to the suppression of noise and ring artifacts.

7.2 The proposed algorithm

We denote the image with x. We extract blocks of size 8^3 from this image for processing. As is commonly done in dictionary-based image processing, we vectorize each extracted block and refer to each vectorized block as a "signal". We also denote by $X \in \mathbb{R}^{512\times N}$ the matrix that contains the vectorized blocks as its columns, with N being the total number of blocks. Our choice of block size of 8^3 is to a large degree arbitrary and we made this choice only to simplify the presentation of the proposed methods.

We propose a structured dictionary that consists of two levels. The top level consists of a fixed orthonormal basis. To build this basis, we begin with a 3D DCT basis of size $4 \times 4 \times 4$ and upsample it by a factor of 2 in each direction. The resulting dictionary will be still orthonormal, although it will not be a basis because there will be only 64 basis vectors that cannot span the space of $\mathbb{R}^{512}$. The second level consists of atoms that are learned from training data as will be explained below. Figure 7.1 shows a schematic of the proposed structured dictionary. We denote the top level dictionary by D_u and the second level dictionary by D_l. Each atom in D_l is grouped under one of the atoms in D_u that has the smallest Euclidean distance with it. We write $D_l(i) \in D_u(j)$ to indicate that the atom i in D_l is grouped

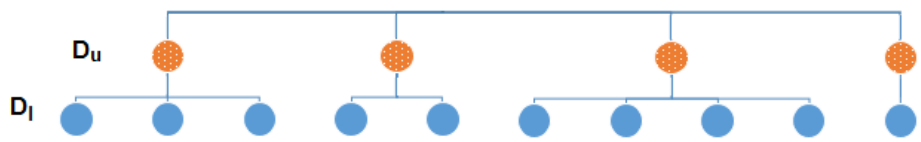

Figure 7.1: A schematic representation of the proposed dictionary structure. The top level dictionary, D_u, is an orthonormal basis such that the decomposition of a signal in D_u can be computed very efficiently. The second-level dictionary, D_l, contains atoms that are learned from training data. The goal is to find the sparse representation of a test signal in terms of atoms from D_l. The first-level dictionary D_u is used as a guide for selecting the most informative atoms from D_l.

under atom j in D_u.

To explain the rationale behind the proposed dictionary structure, we must note that the most computationally demanding step in greedy sparse coding algorithms is the identification of the most informative dictionary atom. In each iteration of the standard greedy algorithms, such as OMP presented in Algorithm 3, the inner product of the residual and every atom in the dictionary is computed and the atom that has the largest inner product is identified and added to the support (we assume that all dictionary atoms have equal norms). This has to be performed in a straight-forward fashion because the learned dictionary has no structure and there is no fast algorithm for computing the coefficients. Moreover, the number of atoms in learned overcomplete dictionaries is usually very large (usually at least twice the length of the signal). Therefore, this step can be very computationally demanding. This is especially the case when the length of the signal is large, such as in 3D image processing.

As mentioned above, in the proposed two-level dictionary the learned atoms are in D_l, which is the second level. Therefore, the goal is to find the sparse representation of the signal in D_l. The first-level dictionary, D_u, is used as an aid in identifying those atoms in D_l that can potentially be useful in sparse coding of the signal. Algorithm 4 shows how the sparse representation of a signal x in D_l is computed.

In each iteration of the algorithm, we first apply D_u on the residual and identify the s largest coefficients. These are the s atoms in the orthonormal dictionary D_u that are most correlated with the signal residual. For each

input : Dictionary D, the signal x, ϵ
output: c, sparse representation coefficient of x in D
$r = x$
$c = 0$
$I = \{\}$
while $\|r\|_F^2 > \epsilon$ **do**
  $\hat{k} = \text{argmax}_k |\langle r, D_k \rangle|$
  $I = I \cup \hat{k}$
  $c = (D_I^T D_I)^{-1} D_I^T x$
  $r = x - D_I c$
end

Algorithm 3: Greedy sparse coding of signal x in dictionary D using the orthogonal matching pursuit (OMP) algorithm.

of these s atoms in D_u, we separately search the atoms in D_l that are grouped under them and find the atom most correlated with the residual. These atoms are then added to the set of atoms that have been found in the previous iterations to update the support of the signal. The signal is then projected onto the space spanned by these atoms. Finally, the representation is pruned to the s largest coefficients (in magnitude). This process can be repeated for a predefined number of times or until the norm of the residual falls below a threshold. If the sparsity level s is known, the loop may even be performed only once because s atoms would have been already selected in the first iteration.

So far we have assumed that we know the dictionary D_l. In practice, this dictionary must be learned from the training data. To learn D_l, we use the K-SVD algorithm [2] that we described in Section 2.1.2, with slight modifications. The difference between our approach and K-SVD is that for the sparse coding step we use Algorithm 4. In addition, after updating the dictionary atoms in each iteration, we cluster them under D_u by assigning each learned atom to the closest atom in D_u (in terms of the Euclidean distance). Note that atoms with the smallest Euclidean distance are also atoms with the largest inner product because atoms in D_u and D_l have unit norms. We then perform a pruning step on the dictionary atoms. We prune each group of atoms (i.e., all atoms grouped under one of the atoms in D_u) by computing the inner product between all pairs of atoms in that group and eliminating one of the atoms in each pair that have an inner product more than 0.95. We also remove atoms that are used by less than a small fraction (p) of the training signals, where we usually choose p to be between

input : Dictionaries D_u and D_l and the signal x, sparsity level s
output: γ, vector of sparse representation coefficients of x in D_l
$r = x$
$I = \{\}$
$\gamma = 0$
while $||r||_2^2 > \epsilon$ **do**
  $w = D_u^T r$
  $J = \text{supp}(w_s)$
  $K = \{\}$
  for $j \in J$ **do**
    $\hat{k} = \text{argmax}_k \, |\langle D_l(k), r\rangle| \quad D_l(k) \in D_u(j)$
    $K = K \cup \hat{k}$
  end
  $I = K \cup \text{supp}(\gamma)$
  $\beta|_I = ([D_l]_I^T [D_l]_I)^{-1} [D_l]_I^T x$
  $\beta|_{I^c} = 0$
  $\gamma = \beta_s$
  $r = x - D_l \gamma$
end

Algorithm 4: Algorithm for greedy sparse coding in the two-level dictionary. We use u_s to denote a vector u restricted to its s largest (in magnitude) elements. In other words, u_s is equal to u at the location of the s largest components of u and zero elsewhere. We write $u|_I$ to denote vector u restricted to indices in the set I, i.e., $u|_I(i) = u(i)$ for all $i \in I$ and $u|_I(i) = 0$ for all $i \in I^c$. For a matrix D we write $[D]_K$ to denote this matrix restricted to its columns indexed by K. Also, $\text{supp}(u)$ denotes the support of u, which is the set of indices of its non-zero elements. We borrow these notations from [319].

10^{-3} and 10^{-2}.

7.3 Results and discussion

We apply the proposed algorithm for denoising and restoration of FDK-reconstructed images. We compare the proposed algorithm with the K-SVD denoising algorithm [100] which has also been applied for denoising/restoration of CT images [60]. For both the proposed algorithm and K-SVD denoising, we use the simple one-step denoising method that we described in Section 2.1.3.

7.3.1 Denoising

Figure 7.2 shows the performance of our algorithm and K-SVD denoising on noisy images of a brain phantom. This phantom was obtained from the BrainWeb database [63]. The noisy image was reconstructed from projections simulated with incident photon number of 10^5 and assuming additive Gaussian noise with a standard deviation of 100. From Figure 7.2, the proposed algorithm seems to have resulted in better denoising than K-SVD. For an objective comparison, in Table 7.1 we have shown the RMSE, SSIM, CNR, and computational time for two different dictionary sizes.

	Dictionary size= 1024		Dictionary size= 4096	
	Proposed algorithm	K-SVD denoising	Proposed algorithm	K-SVD denoising
RMSE	0.067	0.067	0.061	0.066
SSIM	0.763	0.762	0.781	0.760
CNR	16.1	16.0	16.5	15.9
time (h)	0.11	0.20	0.19	0.66

Table 7.1: Denoising of the FDK-reconstructed image of a brain phantom with the proposed two-level dictionary and K-SVD denoising.

The proposed algorithm has achieved comparable or better results than K-SVD, while having a shorter computational time as well. Increasing the dictionary size from 1024 to 4096 (that is, increasing the degree of overcompleteness from 2 to 8) has improved the performance of the proposed algorithm, but it has had little influence on the performance of the K-SVD denoising. We think that this is because the clustering of the atoms in the

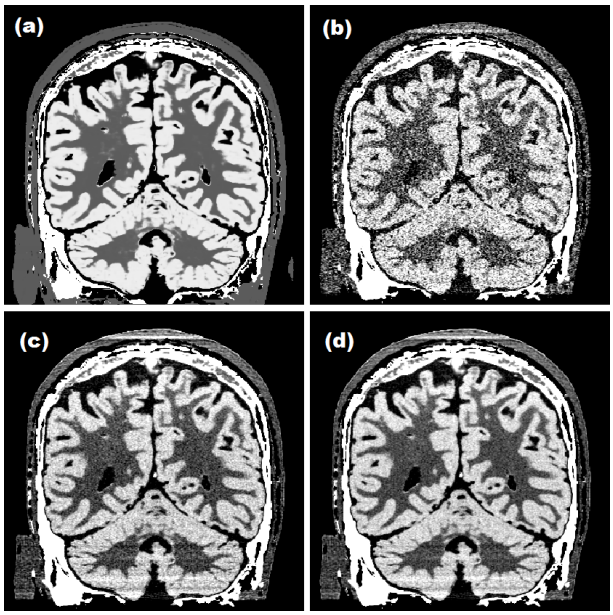

Figure 7.2: Visual comparison of the proposed algorithm and K-SVD denoising on denoising of brain phantom images. (a) the true phantom, (b) the noisy image, (c) denoised with K-SVD denoising, (d) denoised with the proposed algorithm.

proposed dictionary structure allows much larger number of atoms to be learned and effectively used during dictionary deployment.

We also applied the proposed algorithm for removing noise from a series of 3D micro-CT images. Figure 7.3 shows a slice from the noisy image of a rat and the images denoised with the proposed algorithm and K-SVD denoising. For a quantitative comparison of the proposed algorithm and K-SVD denoising, in Table 7.2 we have shown the RMSE, SSIM, CNR, and the computation time for denoising of a rat image. The values of RMSE and SSIM were computed by comparing the images with a reference image, also shown in Figure 7.3, that was reconstructed with 25 iterations of MFISTA. The proposed algorithm has achieved a slightly better image quality while reducing the computational time by approximately a factor of 3.

7.3.2 Restoration

We applied the proposed algorithm for removing ring artifacts from a series of 3D micro-CT images. The images used in this section contain substantial

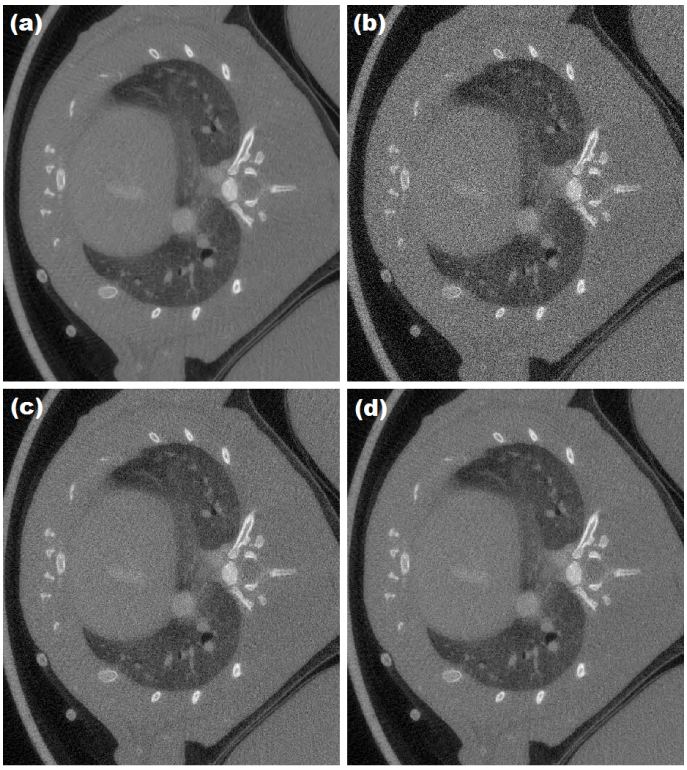

Figure 7.3: Visual comparison of the proposed algorithm and K-SVD denoising on noisy rat images. (a) the reference image, (b) the noisy image, (c) denoised with K-SVD denoising, (d) denoised with the proposed algorithm.

amounts of ring artifacts that are caused by detector saturation. We show some of our results as figures. Because we are unable to reconstruct high-quality reference images in this experiment, we cannot provide a quantitative evaluation.

Figure 7.4 shows some of our results from this experiment. The dictionary is learned from a set of artifact-free training images. This learned dictionary is then used for processing of images with ring artifacts. The rationale is simply that the learned dictionary will be adapted to representing the image features and not the ring artifacts. Therefore, sparse representation of image blocks in the dictionary will lead to the suppression of the artifacts. As can be seen in Figure 7.4, the proposed algorithm has resulted in a substantial reduction of artifacts. In general, we observed that the performance of the proposed algorithm in reducing these these artifacts is

	Proposed algorithm	K-SVD Denoising
RMSE	0.0140	0.0144
SSIM	0.770	0.770
CNR	22.6	22.4
time (h)	1.3	3.4

Table 7.2: Denoising of the FDK-reconstructed image of a rat from real micro-CT scan with the proposed two-level dictionary and K-SVD denoising.

slightly better than K-SVD, while being at least twice faster.

We should emphasize that the observed success of this simple strategy in suppressing the ring artifacts is because of the shape of these artifacts. Specifically, these artifacts have different shapes than the genuine image features. Because we learn the dictionaries from clean (artifact-free) training images, the dictionary atoms are adapted to representing the genuine image features but they cannot represent the ring artifacts as well. Therefore, sparse representation of image blocks in these dictionaries (which is equivalent to shrinkage/thresholding of the coefficients) leads to suppression of the artifacts. If the artifacts were similar in shape to the genuine image features, this simple strategy would probably not be as effective.

There are two parameters that could impact the performance of the proposed algorithm in suppression of ring artifacts. One of these parameters is the dictionary size. As we saw from Table 7.1, compared with an unstructured dictionary, the proposed structured dictionary has the potential to effectively learn and use a larger number of atoms for the denoising task. We would like to know how the dictionary size may influence the performance of the proposed algorithm for artifact suppression. Another very important setting is the sparsity level, i.e., the number of dictionary atoms used to represent each of the image blocks. The sparsity has been denoted with s in Algorithm 4.

In Figure 7.5 we have shown the performance of the proposed algorithm for two different dictionary sizes (1024 and 4096) and three different sparsity levels (2, 4, and 8). The results show a very slight improvement in the quality of the resulting images with a dictionary of 4096 atoms, compared with a dictionary with 1024 atoms. The sparsity level has a more significant effect on the resulting image quality. As expected, smaller sparsity levels have led to a stronger artifact suppression because a smaller number of atoms are used in representing each image block and, hence, artifacts have a smaller chance of being represented. On the other hand, this also results in a blurring of

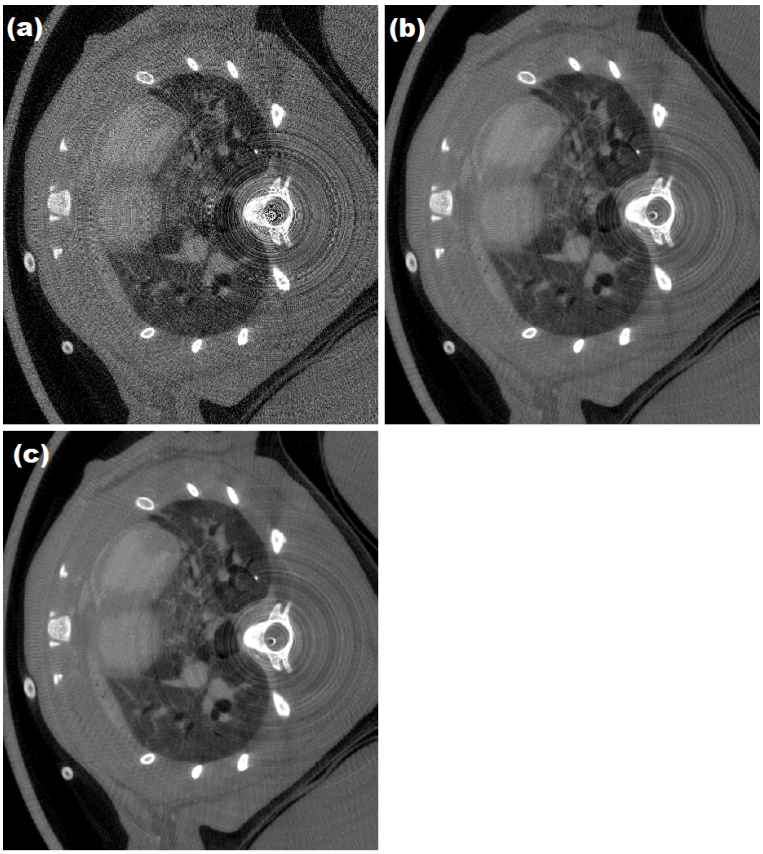

Figure 7.4: (a) A rat image with strong ring artifacts, (b) the same image after being processed with a standard dictionary, (c) the same image after being processed with the proposed algorithm.

true image features.

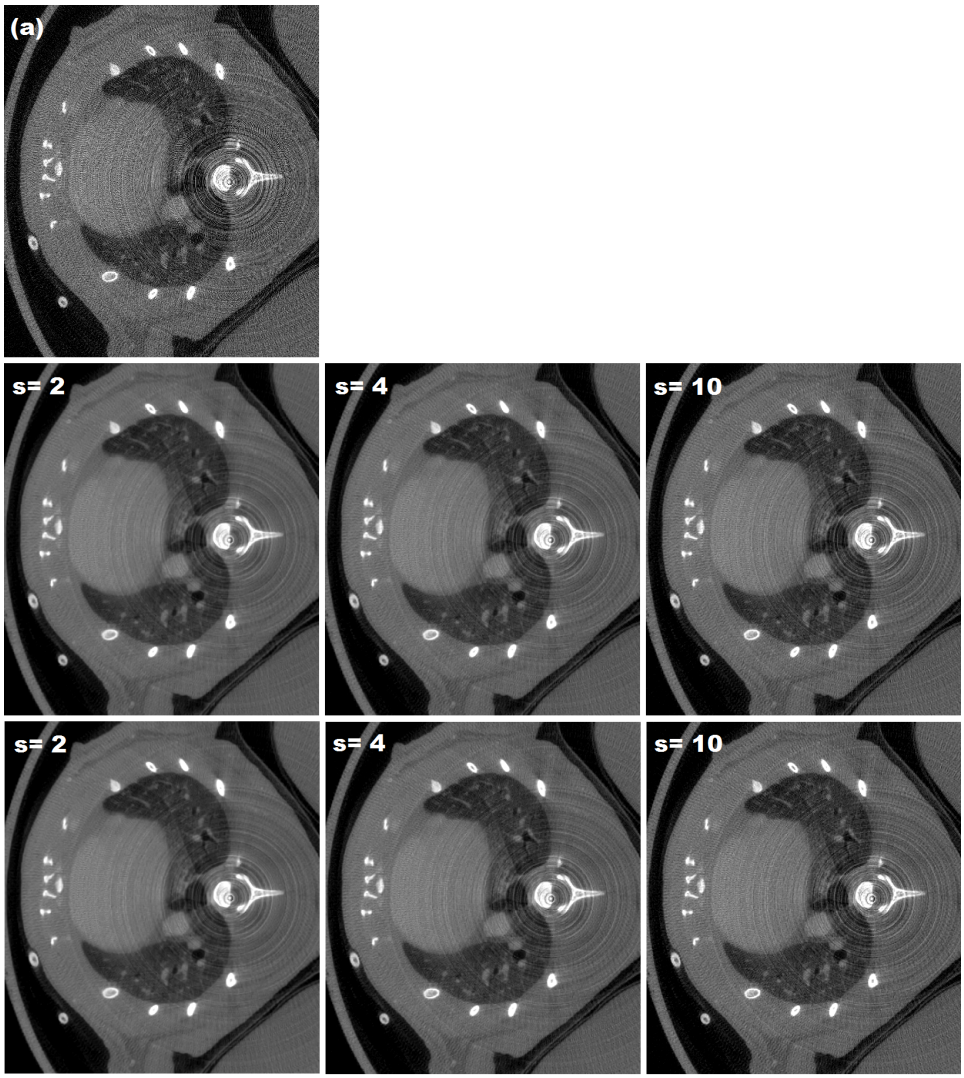

Figure 7.5: Effect of dictionary size and sparsity level on the performance of the two-level dictionary for suppressing ring artifacts. (a) the original artifact-full image. The second row shows the processed images with a dictionary of 1024 atoms. The third row shows the processed images with a dictionary of 4096 atoms. The sparsity levels are shown on each image.

Chapter 8

TV-Regularized Iterative Reconstruction

8.1 Introduction

8.1.1 Motivation and background

When the number of CT projection measurements is small and/or the measurements are very noisy, statistical and iterative reconstruction methods can lead to a much higher image quality compared to analytical reconstruction methods. Therefore, statistical and iterative methods can reduce the amount of radiation used for imaging. Despite their high computational requirements, iterative reconstruction methods have received increasing attention in recent years. This revival of interest is due to several factors including increased awareness of the health risks associated with exposure to radiation, availability of faster computers, and algorithmic advancements. In recent years, significant progress has been made in the development of CT reconstruction algorithms. State-of-the-art CT reconstruction methods rely on effective models of the image and use efficient optimization methods such as accelerated first-order methods and variable-splitting algorithms. These methods have led to very promising results in reconstructing high-quality images from undersampled and noisy measurements [61, 160, 240, 251, 263].

Even though there has been significant progress in reducing the number of measurements required for high-quality image reconstruction, the long computational times can still be a major limiting factor in the adoption of iterative algorithms in practice. Although new hardware options such as GPU offer significant speedups, the size and resolution of the reconstructed images also continue to grow and many clinical applications demand reconstruction of high-quality images in very short times. Therefore, there is a great need for algorithms that can converge to a high-quality image in a small number of iterations. In this chapter, we propose a reconstruction algorithm based on a new class of stochastic gradient descent methods. The basic stochastic gradient descent method has been widely used in various

applications in signal processing and machine learning for decades. However, it has a poor theoretical convergence rate and in practice it usually fails to converge to an accurate solution. In recent years, new stochastic gradient descent algorithms that overcome these shortcomings have been proposed. We will review some of these algorithmic developments and propose an algorithm for image reconstruction for cone-beam CT. We will apply our algorithm on simulated and real cone-beam projection data and compare it with some of the state-of-the-art reconstruction algorithms.

8.1.2 Formulation of the problem

We consider a linear model for CT projection measurements:

$$\hat{y} = \hat{A}x + v \tag{8.1}$$

where $\hat{A}$ represents the projection matrix, $\hat{y}$ denotes the projection measurements, x is the unknown image to be estimated, and v is the measurement noise.

As mentioned in Section 1.2, the noise, v, in the sinogram (i.e., after the log transformation) is very close to a Gaussian with zero mean and a signal-dependent variance, $\sigma_i^2 \propto \exp(\bar{y}_i)$, where $\bar{y}_i$ is the expected value of the sinogram at detector i. Therefore, following the maximum-likelihood principle, it is natural to use a weighted least-squares cost function of the following form:

$$F(x) = \frac{1}{2}\|\hat{A}x - \hat{y}\|_W^2 = \frac{1}{2}(\hat{A}x - \hat{y})^T W(\hat{A}x - \hat{y}) \tag{8.2}$$

where W is a diagonal matrix whose diagonal elements are proportional to the inverse of the measurement variances, σ_i^2. To simplify the notations, we define $A = W^{1/2}\hat{A}$ and $y = W^{1/2}\hat{y}$ so as to transform the cost function (8.2) into a standard least-squares cost function:

$$F(x) = \frac{1}{2}\|Ax - y\|_2^2 \tag{8.3}$$

It should be noted that the weight matrix W depends on the mean sinogram data, $\bar{y}$, as shown in Equation (1.4), but $\bar{y}$ is not available in practice because only one sinogram is measured. However, we can use the knowledge that the sinogram is always smooth and slowly-varying. Therefore, we smooth the measured sinogram and use it for computing the weights W.

Because the inverse problem of estimating x from the measurements y is ill-posed, it is common to add a regularization term to $F(x)$. Here we

use a total variation regularizer and also write $F(x)$ as an average over the projection views. The resulting composite cost function is as follows:

$$\Phi(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\|A_i x - y_i\|_2^2 + \lambda\,\mathrm{TV}(x) \tag{8.4}$$

where n represents the number of projection views and λ is the regularization parameter. Clearly, y_i denotes the vector of measurements of the ith projection view and A_i denotes the sub-matrix of A formed by keeping only those rows that correspond to the measurements in y_i. We will denote the measurement inconsistency term for the ith projection view by f_i, i.e., $f_i(x) = \frac{1}{2}\|A_i x - y_i\|_2^2$.

The algorithm proposed in this chapter makes use of the gradient of F and the gradient of the component functions, f_i. The gradient of the complete measurement inconsistency term in Equation (8.4) is:

$$\nabla F(x) = \frac{1}{n}A^T(Ax - y) \tag{8.5}$$

Similarly, the gradient of a component function, f_i, is:

$$\nabla f_i(x) = A_i^T(A_i x - y_i) \tag{8.6}$$

The gradient descent method for minimizing $F(x)$ iteratively performs the following update:

$$x^{k+1} = x^k - \alpha_k \nabla F(x^k) \tag{8.7}$$

where α_k is the step size.

To deal with the non-smooth regularization term, we use the proximal-gradient method [66, 67], which is an extension of gradient descent method in the following sense. It is easy to show that the update in (8.7) is equivalent to the following problem:

$$x^{k+1} = \arg\min_x \left\{ F(x^k) + \nabla F(x^k)^T(x - x^k) + \frac{1}{2\alpha_k}\|x - x^k\|_2^2 \right\} \tag{8.8}$$

where the expression being minimized is a quadratic approximation to $F(x)$ in the neighborhood of x^k [43]. The proximal gradient methods account for the non-smooth regularization term ($\mathrm{TV}(x)$ in our case) by simply adding it to this approximation:

$$x^{k+1} = \arg\min_x \left\{ F(x^k) + \nabla F(x^k)^T(x - x^k) + \frac{1}{2\alpha_k}\|x - x^k\|_2^2 + \mathrm{TV}(x) \right\}$$

The corresponding update rule for the above problem is:

$$x^{k+1} = \mathbf{prox}_{\alpha_k \mathrm{TV}} \left(x^k - \alpha_k \nabla F(x^k) \right) \tag{8.9}$$

where the proximal operator or proximal map is defined as:

$$\mathbf{prox}_{\mathrm{TV}}(x) = \arg\min_u \left\{ \frac{1}{2} \|u - x\|_2^2 + \mathrm{TV}(u) \right\} \tag{8.10}$$

In this chapter, we use the algorithm proposed in [46] for solving the proximal operation in (8.10). This step does not have to be solved to a great accuracy. In our experience, one to three iterations of the algorithm in [46] are sufficient for fast convergence of the algorithm proposed in this chapter. Theoretically, it has been proven that if the error in the computation of the proximal mapping decreases gradually, both the basic proximal gradient method and the accelerated proximal gradient method achieve the same convergence rate as in the error-free case [284].

Two properties of the objective function that are particularly important to the performance of first-order methods are Lipschitz continuity of the gradient and strong convexity. Function $F(x)$ is said to have a Lipschitz-continuous gradient if

$$\|\nabla F(x) - \nabla F(y)\|_2 \leq L \|x - y\|_2 \qquad \forall x, y \in \mathrm{dom}(F) \tag{8.11}$$

for some constant L. It is said to be strongly convex with parameter γ if

$$F(y) \geq F(x) + \nabla F(x)^T (y - x) + \frac{\gamma}{2} \|x - y\|_2^2 \qquad \forall x, y \in \mathrm{dom}(F) \tag{8.12}$$

For $F(x)$ defined as (8.3), L and γ are equal, respectively, to the largest and smallest eigenvalues of $A^T A$, which can be found easily using power methods [118]. Therefore, $\nabla F(x)$ and $\nabla f_i(x)$ are Lipschitz continuous and we will denote the Lipschitz constant of $\nabla f_i(x)$ with L_i. However, they are not strongly convex because A (and therefore all A_i) have more columns than rows; therefore $\gamma = 0$.

8.1.3 Stochastic gradient descent method

In many signal processing and machine learning problems, we are interested in finding a minimizer of the sum (or the average) of a large number of functions or the sum (or the average) of a function over a large number

of training examples. An example is empirical risk minimization, which is widely encountered in machine learning problems. In these applications, the objective function $F(x)$ can be written in terms of functions $f_i(x)$:

$$F(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x) \tag{8.13}$$

The standard (full) gradient descent (FGD) method for minimizing $F(x)$ suggests an iteration of the form:

$$x^{k+1} = x^k - \alpha_k \nabla F(x^k) = x^k - \alpha_k \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x^k), \tag{8.14}$$

where α_k is the step size. The problem is that computing the full gradient is usually very costly. On the other hand, there is usually a large amount of correlation between the measurements. This implies that computation of the full gradient is not necessary to make adequate progress. Instead, the descent direction suggested by a small subset of the component functions, $\{f_i,\, i = 1 : n\}$, can lead to good progress. Because computing ∇f_i is n times faster than computing $\nabla F(x)$, this can be of significant practical value when n is large, which is the case in many problems in machine learning and signal processing. Image reconstruction in CT also fits this model very well. Computing the gradients as shown in Equation (8.5) involves forward and back-projection operations, which are the most expensive operations in CT reconstruction. Also, the number of projections used for reconstruction is usually very large (several tens or hundreds) and there is a large amount of correlation between the measurements in different projection views. This has long been recognized by researchers working on CT reconstruction. The method of ordered subsets, which has been used to accelerate many of the standard CT reconstruction algorithms, is based on this idea [104, 338, 360].

Among the standard methods for minimizing a function like (8.13) is the stochastic gradient descent (SGD) method. SGD computes an update direction based on the gradient of one of the component functions, f_i. Therefore, each iteration of SGD has the following form:

$$x^{k+1} = x^k - \alpha_k \nabla f_{i_k}(x^k) \tag{8.15}$$

where the index i_k is chosen from among the set $\{1, ..., n\}$ based on some probability distribution. The computational cost of each iteration of SGD is $1/n$ that of FGD. However, although the expected value of the stochastic

gradient directions, $\nabla f_i(x)$, is equal to the full gradient, $\nabla F(x)$, their variance is very high. As a result, the convergence rate of SGD is much worse than that of FGD. Specifically, for smooth functions the convergence rate of FGD is $O(1/k)$ [236], which can be improved to $O(1/k^2)$ using acceleration methods [18, 235]. This $O(1/k^2)$ convergence rate is known to be optimal, meaning that no first-order method can achieve a faster convergence [236]. On the other hand, the optimal convergence rate of SGD is $O(1/\sqrt{k})$ [233]. Furthermore, unlike FGD, the convergence rate of SGD does not improve when the objective function has Lipschitz continuous gradient [173]. This means that SGD does not exploit this important property, which is satisfied by the measurement misfit term in CT (8.3). In practice, the basic SGD algorithm has a very fast initial convergence speed and is a good method for obtaining rough solutions to large-scale problems. However, obtaining an accurate solution with SGD will require a very large number of iterations and will demand that the step size be gradually reduced.

As mentioned above, the difference in the convergence rates of FGD and SGD is due to the fact that the variance of the SGD directions does not diminish as the signal estimate gets closer to a solution. There has been much research on improving the convergence behavior of the basic SGD algorithm and a complete review of this immense literature is beyond the limitations of this dissertation. Here, we mention some of the central ideas and main approaches. A well-known approach to reducing the effect of large variance of SGD directions is to use diminishing step sizes. There are various guidelines on how to gradually reduce the step size. Common choices include exponential decay ($\alpha_k = \alpha_0 a^k$) and inverse decay ($\alpha_k = \alpha_0/(1 + a\alpha_0 k)$) [10, 31]. The problem with this approach is that it requires careful tuning of the hyper-parameters (α_0 and a) and it can only achieve a sublinear convergence rate [26, 296]. Another natural approach to reducing the variance is averaging. Different studies have used averaging of the gradient directions ($\nabla f_{i_k}(x^k)$), averaging of the signal estimates (x^k), or a combination of the two [128, 238, 260, 288, 335]. In general, averaging will lead to faster convergence rates and increased robustness of the convergence rate to the selection of the step size. However, the improvement in the convergence speed obtained by averaging is usually small. Some algorithms start by performing simple SGD updates and gradually increase the number of functions involved in the computation of the gradient directions so that the update directions gradually approach the full gradient descent directions. Examples of these type of methods, which are called hybrid methods, can be found in [25, 109]. With careful selection of the step size and batch size, these algorithms can significantly improve the basic SGD method. However, these methods do

not achieve the convergence rate of variance-reduced SGD algorithms that we will describe below. Some studies have suggested SGD methods with momentum [320]. In these algorithms, the SGD update in (8.15) is augmented with a multiple of the previous update direction(s). However, these momentum methods still require diminishing step sizes and achieve a small improvement over the basic SGD algorithm.

8.1.4 Variance-reduced stochastic gradient descent

Although the modified SGD algorithms mentioned above can improve the basic SGD method, our focus here is on a new class of SGD algorithms that have been proposed very recently. The core idea in these methods is to keep a copy of the full gradient direction or copies of the stochastic gradient directions and use them in building the update directions. With this trick, these methods achieve a linear convergence rate on strongly convex functions. On smooth but not strongly convex functions, they achieve $O(1/k)$ convergence, which is a dramatic improvement over the $O(1/\sqrt{k})$ convergence rate of the basic SGD. To the best of our knowledge, the first such algorithm was the stochastic average gradient (SAG) algorithm proposed in [177]. In our notation, the update suggested by SAG is as follows:

$$x^{k+1} = x^k - \alpha_k \left(\nabla f_{i_k}(x^k) - \nabla f_{i_k}(\tilde{x}) + \nabla F(\tilde{x}) \right) \tag{8.16}$$

where $\tilde{x}$ is an old signal estimate for which we have stored the stochastic gradient directions $\{\nabla f_i(\tilde{x}),\ i = 1 : n\}$. Algorithms such as SAG have been shown to be very successful on a variety of machine learning problems in recent years [79, 165, 206, 241]. The reason for the success of these algorithms lies in the fact that as the signal estimate becomes closer to the solution, the variance of the update direction approaches zero. This is because as x^k and $\tilde{x}$ approach the optimal point x^*, $\nabla F(\tilde{x}) \to 0$, by the definition of optimality. Now, if $\nabla f_i(x^k) \to \nabla f_i(x^*)$ and $\nabla f_i(\tilde{x}) \to \nabla f_i(x^*)$ (which is the case if f_is have Lipschitz-continuous gradients), then:

$$\nabla f_i(x^k) - \nabla f_i(\tilde{x}) + \nabla F(\tilde{x}) \to 0 \tag{8.17}$$

Rigorous convergence proofs can be found in the original papers. For 3D CT, the problem with SAG and many similar algorithms that have been proposed recently (e.g., [79, 206]) is that they require that the most recent copies of all stochastic gradients, $\nabla f_i(x)$, to be saved. For reconstruction of a $500 \times 500 \times 500$ image from 100 projections, this will require at least $500^3 \times 100 \times 4$ bytes $= 50$ GB of memory. Therefore, we suggest an algorithm

that only requires the saving of the full gradient direction, $\nabla F(\tilde{x})$. The algorithm that we will propose is similar to the stochastic variance-reduced gradient (SVRG) algorithm [148] which we present below in Algorithm 5.

Data: x^0
Result: x^J
for $j \leftarrow 1$ **to** J **do**
  $\tilde{x} = x^{j-1}$;
  $\tilde{\mu} = \nabla F(\tilde{x})$;
  $x_j^0 = \tilde{x}$;
  for $k \leftarrow 1$ **to** K **do**
    select an index i_k from among the set $\{1, ..., n\}$;
    $x_j^{k+1} = x_j^k - \alpha(\nabla f_{i_k}(x_j^k) - \nabla f_{i_k}(\tilde{x}) + \tilde{\mu})$;
  end
  $x^j = x_j^K$;
end

Algorithm 5: SVRG algorithm [148].

The SVRG update is similar in form to the SAG update. The difference is that, unlike SAG, SVRG does not store copies of every $\nabla f_i(x)$. Instead, only the full gradient, $\nabla F(\tilde{x})$, is computed and stored. The price that one pays, on a general problem, is that every iteration will require evaluation of two SGD directions, $\nabla f_{i_k}(x_j^k)$ and $\nabla f_{i_k}(\tilde{x})$. However, as we will see below, this is not the case in our problem because the gradient is a linear function. Computation of $\nabla F(\tilde{x})$ is followed by a large number (K in Algorithm 5) of variance-reduced SGD updates, after which the algorithm recomputes $\nabla F(\tilde{x})$ and the process repeats. As for SAG, the reason for SVRG's effectiveness lies in the fact that the variance of the update direction, $\nabla f_{i_k}(x_j^k) - \nabla f_{i_k}(\tilde{x}) + \tilde{\mu}$, is very low compared to the update direction of the basic SGD method, especially as we get close to the solution. As a result, SVRG can work with a constant and large step size α that does not need tuning and it can achieve significantly faster convergence rates [148, 336].

SVRG and its proximal version [336] are among the state-of-the-art variance-reduced SGD algorithms. One of the aspects of SVRG that later algorithms have tried to improve upon is the frequency of computation of the full gradient, $\nabla F(\tilde{x})$. In the original SVRG algorithm presented in Algorithm 5, the full gradient is computed after a fixed number of SGD iterations. It is suggested that $\nabla F(\tilde{x})$ be recomputed after two SGD passes through the entire data, i.e., $K = 2n$ in Algorithm 5 [148]. Therefore, the computation

of the full gradient can account for up to 1/3 of the total computational cost of the SVRG algorithm. A few studies have tried to reduce the frequency of computation of the full gradient. The MixedGrad algorithm suggested in [206] reduces the number of computations of $\nabla F(\tilde{x})$ to $O(\log(K))$, where K is the number of SGD updates. MixedGrad is based on the assumption that as the signal estimate becomes closer to the solution, $\nabla F(\tilde{x})$ changes less significantly and it needs to be updated less frequently. Theoretical analysis shows that MixedGrad has a convergence rate similar to that of SVRG, while reducing the number of $\nabla F(\tilde{x})$ updates. Another strategy was suggested by S2GD algorithm [166]. In S2GD, the number of SGD updates after each computation of $\nabla F(\tilde{x})$ is a random variable following a specially-devised probability distribution function and it gradually increases as we approach a solution.

Below, we will propose an algorithm for image reconstruction in CBCT. The core iteration in our algorithm consists of a variance-reduced SGD update similar to SVRG. The main differences between our proposed algorithm and those described above include: (1) we suggest a heuristic strategy for deciding when to update the full gradient, $\nabla F(\tilde{x})$, and (2) our algorithm gradually increases the batch size and transforms into a limited-memory quasi-Newton method as the algorithm approaches a solution. In this sense, the proposed algorithm is similar to the hybrid methods that we briefly described above.

8.2 Methods

8.2.1 The proposed algorithm

Our proposed algorithm is presented below (Algorithm 6). In this algorithm, we use subscripts for the indices of the main loop (that updates $\tilde{\mu} = \nabla F(\tilde{x})$) and superscripts for the indices of the inner loop (involving SGD-type updates). To make the algorithm easy to follow, we have omitted some of the details that we will explain here and in the next subsection.

Our algorithm starts by running through all f_is and performing a simple proximal SGD update for each f_i. It has been shown that performing a basic SGD update during the first run through the data leads to much faster convergence than an SVRG-type update [285, 287, 336]. Also, at the end of each round of SGD-type updates, we return a weighted average of the $T+1$ latest signal estimates, rather than only the last one. The weighting scheme that we use here is similar to those used in [172] and in our experiments we set $T \approx n/10$ where n is the number of projection views. Another necessary

Data: x_0, $\{L_i, i = 1, ..., n\}$, $M = 1$
$x_1^0 = x_0$
for $k \leftarrow 1$ **to** n **do**
  select $i_k \in \{1, ..., n\}$
  $\delta = A_{i_k}^T (A_{i_k} x_1^{k-1} - y_{i_k})$
  $x_1^k = \mathbf{prox}_{\lambda \mathrm{TV}} \left(x_1^{k-1} - \frac{\alpha_k}{L_{i_k}} \delta \right)$
end
$\tilde{x} = \frac{2}{(T+1)(T+2)} \sum_{k=n-T}^{n} (k - n + T + 1) x_1^k$
$\tilde{\mu} = \frac{1}{n} A^T (A\tilde{x} - y))$
for $j \leftarrow 2$ **to** J **do**
  $x_j^0 = \tilde{x}$
  for $k \leftarrow 1$ **to** n_{max} **do**
    $\delta_{\text{old}} = \delta$
    select set S_k of size M from $\{1, ..., n\}$
    $\delta = \frac{1}{M} \sum_{i \in S_k} A_i^T A_i (x_j^{k-1} - \tilde{x}) + \tilde{\mu}$
    if $M \geq 5$ **then**
      $d = H_k \delta$
    else
      $d = \delta$
    end
    $x_j^k = \mathbf{prox}_{\lambda \mathrm{TV}} \left(x_j^{k-1} - \frac{\alpha_k}{L} d \right)$
    if $\langle \delta, \delta_{old} \rangle < 0$ & $(\Delta\Phi)_k < \epsilon_1$ **then**
      break
    end
  end
  $\tilde{x} = \frac{2}{(T+1)(T+2)} \sum_{k=n-T}^{n} (k - n + T + 1) x_j^k$
  $\tilde{\mu} = \frac{1}{n} A^T (A\tilde{x} - y))$
  if $\Delta\Phi < \epsilon_2$ **then**
    $M = 2 \times M$
  end
end

Algorithm 6: The proposed algorithm. (L_i: Lipschitz constants, n: the number of projection views, M; the batch size, λ: the regularization parameter, α: the step size, T: number of latest signal estimates that are averaged to find the new signal estimate at the end of each round of stochastic minimization loops, H: inverse Hessian estimate.

modification is that (because our cost function includes the non-smooth TV regularization term) we use proximal stochastic gradient steps instead of plain stochastic gradient steps.

Unlike SVRG (and more recent algorithms such as MixedGrad) that use a pre-set and fixed update frequency for $\nabla F(\tilde{x})$, our proposed algorithm determines when $\nabla F(\tilde{x})$ needs to be updated on the fly. Specifically, we recompute $\tilde{\mu} = \nabla F(\tilde{x})$ only if both of the following conditions are satisfied: (1) the inner product of two successive descent directions (δ) is negative (i.e., the two successive stochastic gradient directions make an angle larger than 90°), which is commonly taken as a sign that the quality of the update directions is poor, and (2) if the decrease in the objective function Φ is below a certain threshold, ϵ_1. Although Algorithm 6 shows that these conditions are checked after every iteration of the inner loop, this is not necessary. Instead, one can check these conditions after, e.g., every 10 iterations of the inner loop. Moreover, the change in the objective function $(\Delta\Phi)_k$ used in this step does not have to include the complete measurement misfit term, but only those f_is that are involved in the current iteration, i.e. $\{f_i : i \in S_k\}$, making the computation of $(\Delta\Phi)_k$ very cheap.

The other main feature of the proposed algorithm is that it gradually increases the batch size (the number of f_i used to calculate the update direction). This modification puts our algorithm in the class of hybrid methods that we mentioned above. As we explained, hybrid methods gradually increase the batch size in order to reduce the variance of the update directions as the algorithm approaches a solution. However, this is not our goal since our algorithm already uses variance-reduced update directions. Our goal is to try to exploit the curvature information to make faster progress towards a solution. When the batch size (M) is larger than 4, we use a quasi-Newton method to compute the update direction. This allows the algorithm to use the curvature information as the algorithms gets closer to a solution. We have shown this procedure with a simple notation, $d = H_k\delta$, in the algorithm, where H_k denotes the current inverse Hessian estimate. In fact, we use the limited-memory BFGS algorithm [243]. This algorithm uses an iterative procedure to compute d that only involves vector multiplications. When M is too small, the direction d generated by this procedure is poor.

The algorithm starts with a batch size of $M = 1$ and doubles the batch size every time the reduction in the objective function, $\Delta\Phi$, falls below a threshold, ϵ_2. Unlike the $(\Delta\Phi)_k$ used for deciding whether or not to update $\tilde{\mu}$ as explained above, the $\Delta\Phi$ used here will include the full measurement misfit term, but this will not require much extra computation because the projection $A\tilde{x}$ is already computed in the previous step. A

hybrid deterministic-stochastic algorithm is proposed in [109], in which the authors show that exponentially increasing the batch size after every pass through the data leads to good theoretical and practical convergence rates. However, in our experience with CBCT projection data, it is usually much better to increase the batch size very slowly. In fact, in all our experiments at least for three passes through the data the batch size remained $M = 1$. This is perhaps because in our application n is much smaller than in most machine learning applications.

8.2.2 Implementation details

Sampling.

Sampling refers to the strategy for selecting a function f_i at each step of the stochastic gradient descent algorithm. Cyclic and uniformly random sampling are the simplest and most widely used strategies. When the component functions f_i have Lipschitz gradients, choosing f_i with a probability proportional to its Lipschitz constant, L_i, leads to better theoretical and practical convergence [43, 336]. It has recently been shown that this strategy is a good approximation to the optimal sampling strategy for the basic SGD algorithm [232, 357]. In machine learning applications, an effective strategy is to divide the data into small clusters such that there is low within-cluster variance and use a stratified sampling strategy [356]. This last strategy is similar to a sampling technique that is commonly used in the implementation of the ordered subsets method for CT reconstruction: it is common to adopt subset orderings that lead to large angles between successive projection views used by the algorithm [130, 159]. The idea behind this method is that two projections that have a small angular spacing between them include much redundant information; therefore, convergence should be faster if successive projections are far apart. We use this ordering for the initial stage of the algorithm (which consists of ordinary SGD updates). However, for the rest of the algorithm we use a purely random sampling in which each projection is sampled with a probability proportional to its Lipschitz constant. In our experience, this leads to a slightly faster convergence in practice. We only make sure that a projection view used in the previous iteration is not sampled again in the current iteration.

Step size.

Standard step sizes for convex and strongly convex problems with Lipschitz gradient are $1/L$ and $2/(\gamma + L)$, respectively [43, 236]. For the initial

stage of the proposed algorithm that involves simple SGD updates, we found that a more aggressive step size of $2/L$ leads to faster convergence. For the next iterations we use a step size of $1/L$. However, as we mentioned above, when the batch size, M, is larger than 4, our algorithm transforms into a quasi-Newton method. When this happens, we need to perform a line search to find a suitable step-size. We used a backtracking line search with an initial step size of $2/L$ and Armijo rule [243]. Interestingly, our experience shows that we do not need to perform this line search at every iteration because the step size does not change much between iterations. In our implementation, we perform this line search after every 10 iterations of the inner loop. Moreover, the objective function used in the line search does not have to include the complete measurement misfit term, $F(x)$, but only those $f_i(x)$ that are in the current batch, making the line search much cheaper.

Parameters ϵ_1 and ϵ_2 determine, respectively, how often the full gradient ($\tilde{\mu} = \nabla F(\tilde{x})$) is recomputed and how fast the batch size (M) grows. These parameters are not fixed, but they are updated during the iteration of the algorithm. In fact, they should both gradually decrease because they are thresholds on the reduction in the objective function and this reduction is larger in the early iterations. We have developed heuristic methods for updating ϵ_1 and ϵ_2 that we explain here. Every time we start a round of updates in the inner loop, i.e., $k = 1$, we compute the reduction $(\Delta\Phi)_k$ and set $\epsilon_1 = (\Delta\Phi)_k/2$. The informal justification for this choice is that when we start the iteration of the inner loop we have a fresh $\tilde{\mu} = \nabla F(\tilde{x})$ and, therefore, the reduction in the objective function, $(\Delta\Phi)_k$, must be large. With more inner-loop iterations, the current estimate x_j^k departs further away from $\tilde{x}$ and, hence, $\tilde{\mu} = \nabla F(\tilde{x})$ will be less useful. When the reduction in the objective function is less than 50% of that in the beginning of the inner loop, we will decide that $\tilde{\mu} = \nabla F(\tilde{x})$ needs to be updated. For ϵ_2 we follow a similar heuristic. Specifically, at the end of each iteration of the outer loop, we expect that the reduction in the objective function, $\Delta\Phi$, be at least half of that of the previous iteration. If this is not the case, we double the batch size.

Study on the implementation of the system matrix

The implementation of the system matrix A is of critical importance because it can greatly influence the performance of any iterative reconstruction algorithm. As we briefly mentioned in Section 1.2, in theory the elements of A represent the intersection lengths of rays with voxels. However, this simplified view does not consider important factors such as the size of the

detector elements. Moreover, it is computationally very intensive. Therefore, several algorithms have been suggested for efficient implementation of the system matrix for cone-beam CT. Because A is too large to be saved in the computer memory, these algorithms implement *multiplication* with A and A^T, which in CT are referred to, respectively, as forward-projection and back-projection.

In brief, these efficient implementations of the system matrix represent the image using some form of voxel basis function and compute the system matrix by calculating the convolution of the footprints of these basis functions with the surfaces of the detectors. Let $\mu(\overrightarrow{x})$ denote the continuous 3D map of attenuation coefficients, which is a function of the spatial location $\overrightarrow{x}$. When discretized, $\mu(\overrightarrow{x})$ can be written in the general form shown below:

$$\mu(\overrightarrow{x}) = \sum_j \mu_j b(\overrightarrow{x} - \overrightarrow{x_j}) \tag{8.18}$$

where $b(\overrightarrow{x})$ is called the basis function and μ_j is the attenuation coefficient at location $\overrightarrow{x_j}$. Common types of basis functions include cubic and spherically-symmetric function [185, 193, 221]. The convolution of the footprints of these basis functions on the surfaces of the detectors is related to the values of the elements of A.

We performed a detailed study in which we considered three different implementations of the system matrix:

- The distance-driven algorithm proposed in [78].
- The separable-footprints algorithm [193].
- The algorithm proposed in [363] that uses Bessel functions of order 2 as the voxel basis function.
- In addition to the above state of the art algorithms, we implemented a simple forward and back-projection algorithm that projected only the center of each voxel on the detector plane.

The main results of our study are as follows [152].

- As expected, there is a trade-off between speed and accuracy. More accurate implementations are also more computationally demanding.
- Fast convergence of iterative image reconstruction methods requires accurate implementation of forward and back-projection operations, involving a direct estimation of the convolution of the footprint of the voxel basis function with the surfaces of the detectors.

- Reconstruction of the images of low-contrast objects needs more accurate implementation of the system matrix.
- In iterative image reconstruction, implementations of the system matrix that have a decent level of accuracy lead to faster convergence than implementations that are either inaccurate or too accurate. In our experiments, the implementation based on cubic voxels with separable footprints [193] resulted in the fastest convergence of iterative reconstruction algorithms. Therefore, we used this implementation in all experiments reported in this chapter and in the rest of this dissertation.

8.2.3 Evaluation

We applied the proposed algorithms on sets of simulated and real data described below and compared it with the Monotone Fast Iterative Shrinkage Thresholding Algorithm (MFISTA)[17], Nesterov's third method [234], a Gradient-Projection-Barzilai-Borwein (GP-BB) algorithm as suggested in [251], and also the proximal version of the original SVRG algorithm as described in [336]. All algorithms were implemented in Matlab version R2012b running on a Windows 7 PC with 32 GB of memory and 3.4 GHz Intel Core i7 CPU.

Simulated data Two sets of scans with average incident photon counts of $N_0 = 2 \times 10^3$ and $N_0 = 2 \times 10^4$ were simulated from a 3D Shepp-Logan phantom of size $256 \times 256 \times 256$ voxels with isotropic voxels of $0.1 \times 0.1 \times 0.1\ mm^3$. We will refer to these scans as high-noise and low-noise simulated scans, respectively. Each of these scans consisted of 180 projections between $0°$ and $360°$ with uniform angular spacing. A flat detector of 360×360 pixels was considered. The distances from the source to the detector panel and to the axis of rotation were assumed to be 450 mm and 400 mm, respectively, and the distance between the centers of adjacent pixels of the detector panel was assumed to be 0.113 mm.

We used the original image of the phantom for evaluating the quality of the reconstructed images. We used 30 uniformly-spaced projections for reconstruction with the proposed algorithm and other iterative algorithms. We also reconstructed the image of the phantom with the FDK algorithm both using 30 projections and using all 180 projections.

Real data We used micro-CT scans of the physical phantom and of a dead rat. Each scan consisted of 720 projections at 0.50° intervals. The tube voltage, tube current, and exposure time were 70 kV, 32 mA, and 16 ms, respectively, for both scans. We used the full set of 720 projections with the FDK algorithm to reconstruct high-quality images of the phantom and of the rat. We will call these high-quality images "the reference image" and regard them as the true images of the physical phantom and the rat. We used 120 equally-spaced projections from the phantom and 120 equally-spaced projections from the rat for reconstruction with the proposed algorithm and other iterative algorithms.

8.3 Results

8.3.1 Simulated data

Figure 8.1 shows the plots of RMSE and the objective function as a function of CPU time for different reconstruction algorithms for high-noise and low-noise simulated data. The proposed algorithm shows a much faster convergence rate than other algorithms on both low-noise and high-noise projection sets. In particular, the convergence rate of the proposed algorithm is very fast in the initial iterations. In Table 8.1, we have summarized some of the image quality criteria for images reconstructed using different algorithms. The values in this table correspond to images reconstructed after 1000 s. The images reconstructed using the proposed algorithm have a much higher quality criteria and are very close to or slightly better than the images reconstructed from 180 projections using the FDK algorithm.

For a visual comparison, in Figure 8.2 we have shown a central slice and a central profile through the images of the phantom reconstructed from the high-noise projections using the proposed algorithm and MFISTA (which performed best among the other algorithms on this data). The image reconstructed by the proposed algorithm includes all of the important features of the phantom with very little artifacts and is much superior to the MFISTA-reconstructed image.

8.3.2 Micro-CT scan of the physical phantom

Figure 8.3 shows the plots of RMSE, SNR, CNR, and SSIM for reconstruction of the image of the physical phantom using different algorithms. The horizontal axis is labeled as "iteration number". Here, one iteration means one forward-projection and one back-projection involving all n projection

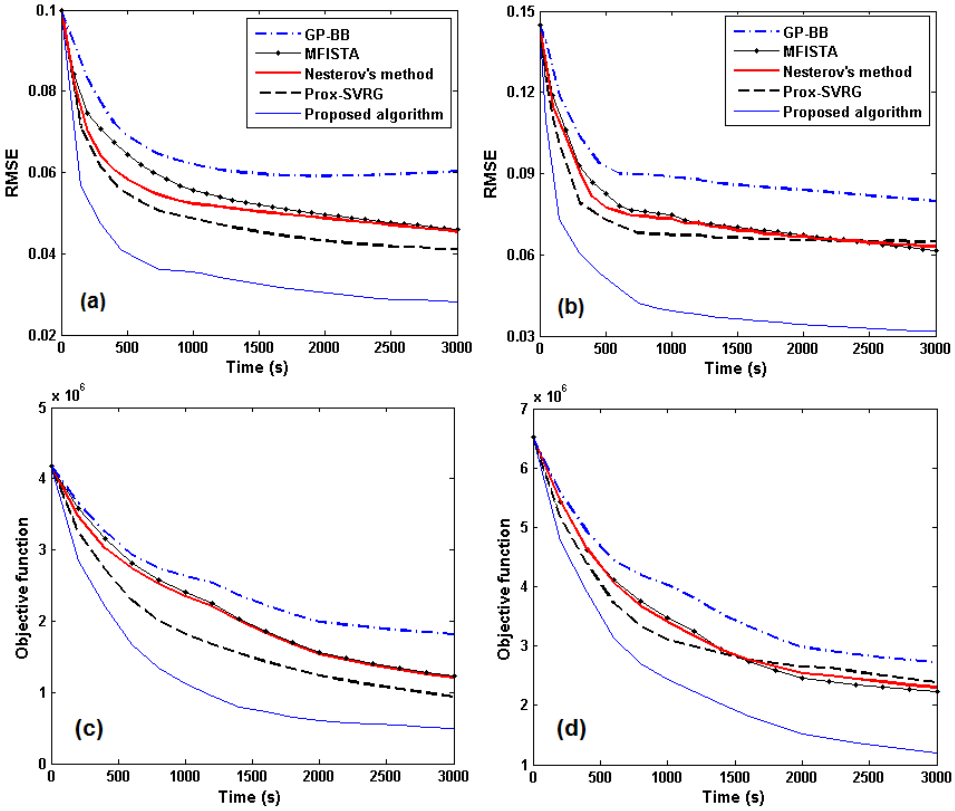

Figure 8.1: Evolution of the reconstruction error (RMSE) and the objective function for reconstruction of the Shepp-Logan phantom. (a) RMSE, low-noise projections; (b) RMSE, high-noise projections; (c) objective function, low-noise projections; (d) objective function, high-noise projections.

views. We use this more convenient measure of computation time because for large images the computational time is dominated by the time required for forward and back-projection. For all iterative algorithms in this study, forward and back-projection accounted for more than 90% of the computation time. The RMSE plots in Figure 8.3 indicate that the proposed algorithm converges to the reference image much faster than the other algorithms, particularly in the initial iterations. The plots of SNR, CNR, and SSIM in Figure 8.3 show that the objective image quality criteria improve much faster with the proposed algorithm than with the other algorithms. In Table 8.2 we have summarized the values of some of the image quality measures after 30 iterations of different algorithms. The image reconstructed

		Proposed	Prox-SVRG	Nesterov	MFI-STA	GP-BB	FDK-30	FDK-180
Low-noise	SSIM	0.741	0.696	0.661	0.643	0.619	0.546	0.730
	MI	0.642	0.555	0.524	0.506	0.488	0.426	0.683
	CNR	5.13	4.14	3.86	3.88	3.78	3.27	5.49
High-noise	SSIM	0.710	0.600	0.561	0.564	0.463	0.335	0.701
	MI	0.627	0.523	0.500	0.494	0.426	0.389	0.667
	CNR	4.81	4.17	3.74	3.74	3.45	2.960	4.99

Table 8.1: Image quality criteria for the images of the Shepp-Logan phantom reconstructed from 30 projections using different algorithms. The numbers next to FDK show the number of projections used with that algorithm.

using the proposed algorithm has higher quality criteria than the images reconstructed using all other algorithms and it is also superior to the image reconstructed using FDK with 240 projections.

	Proposed algorithm	Prox-SVRG	Nesterov	MFI-STA	GP-BB	FDK-120	FDK-240
SSIM	0.747	0.716	0.695	0.696	0.675	0.499	0.703
MI	0.593	0.564	0.540	0.532	0.493	0.370	0.528
CNR	24.0	23.6	23.6	23.5	23.0	17.0	23.3
SNR (dB)	20.9	20.4	20.2	20.2	20.3	18.0	20.1

Table 8.2: Performance comparison between different algorithms in reconstruction of the image of the physical phantom from real data after 30 iterations. The numbers next to FDK indicate the number of projections used with that algorithm.

The phantom includes a set of fine coils that are ideal for visual assessment of the spatial resolution in the reconstructed images. In Figure 8.4, we have shown slices through two of these coils in the images reconstructed using different algorithms. Compared with the images reconstructed by Nesterov's algorithm and Prox-SVRG, the image reconstructed by the proposed algorithm seems to be much closer to the reference image. For a closer comparison, in the same figure we have also plotted the difference between the reference image and the images reconstructed with the proposed algo-

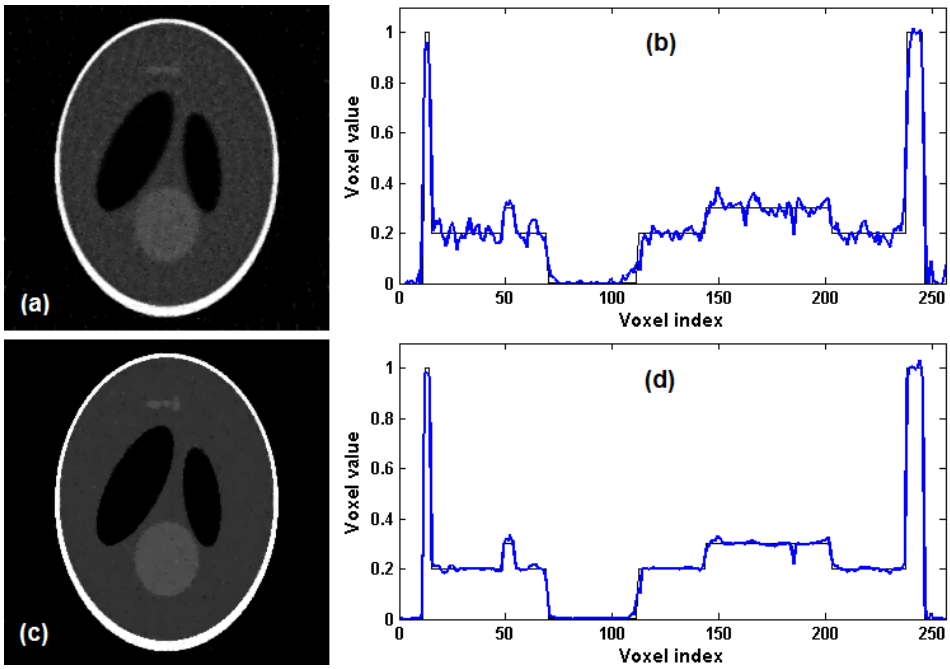

Figure 8.2: (a) The central slice and (b) the central profile of the Shepp-Logan phantom reconstructed using MFISTA; (c) the central slice and (d) the central profile of the Shepp-Logan phantom reconstructed using the proposed algorithm.

rithm and Nesterov's algorithm along a profile through the center of one of these coils. These plots clearly show that the image reconstructed using the proposed algorithm is closer to the reference image.

8.3.3 Micro-CT scan of a rat

Figure 8.5 shows the plots of RMSE, SSIM, MI, and CNR for different algorithms for reconstruction of the image of the rat. The general trends observable in this figure are very similar to those in Figure 8.3. What is most important to us is that the convergence of the image reconstructed with the proposed algorithm to the reference image is very fast. Also, all objective image quality measures improve much faster with the proposed algorithm. Prox-SVRG algorithm also has a fast initial convergence rate but falls behind the proposed algorithm with more iterations.

Table 8.3 summarizes some of the image quality criteria for the recon-

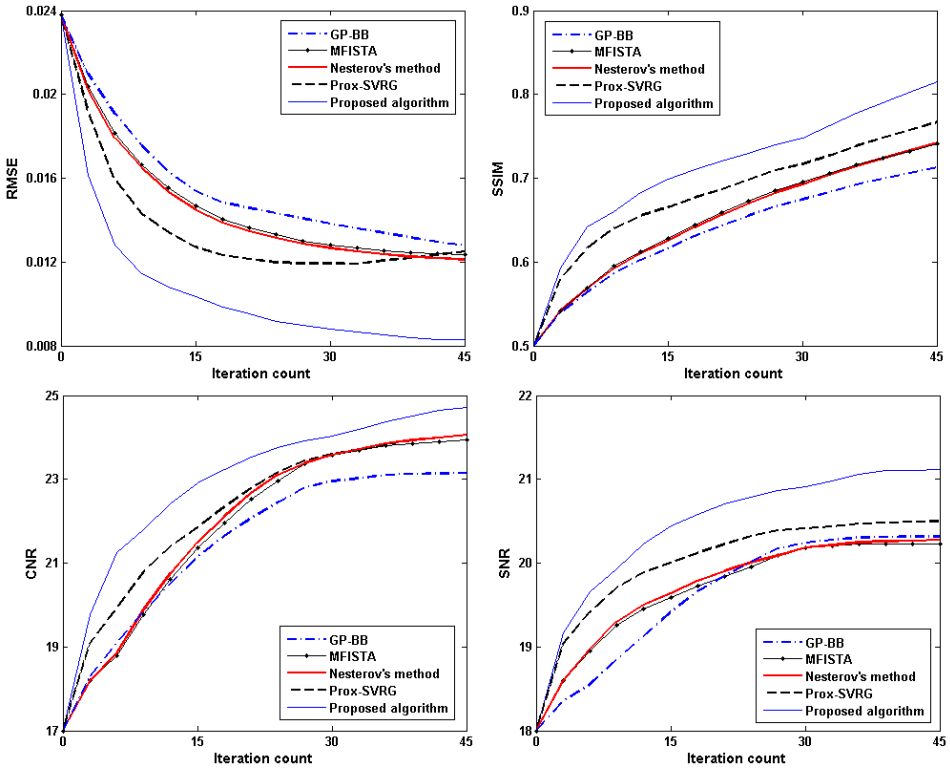

Figure 8.3: Plots of RMSE, SSIM, CNR, and SNR for reconstruction of the image of the physical phantom from real CBCT projections using different algorithms.

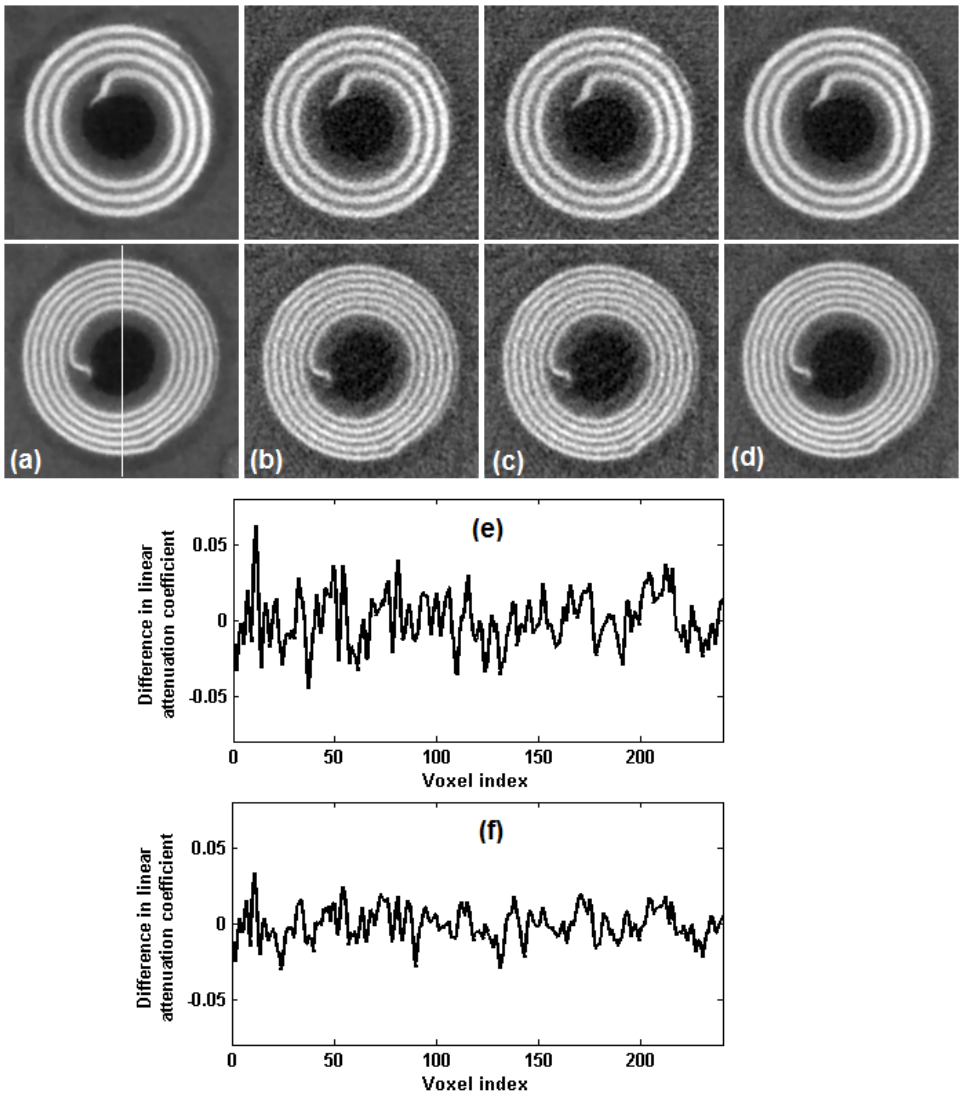

Figure 8.4: Slices through two of the coils in the images of the physical phantom: (a) the reference image, (b) reconstructed using Nesterov's method, (c) reconstructed using Prox-SVRG, (d) reconstructed using the proposed algorithm. The plots show the difference between the reference image and the images reconstructed using (e) Nesterov's method and (f) the proposed method along the vertical line shown in the image of the coil in the reference image.

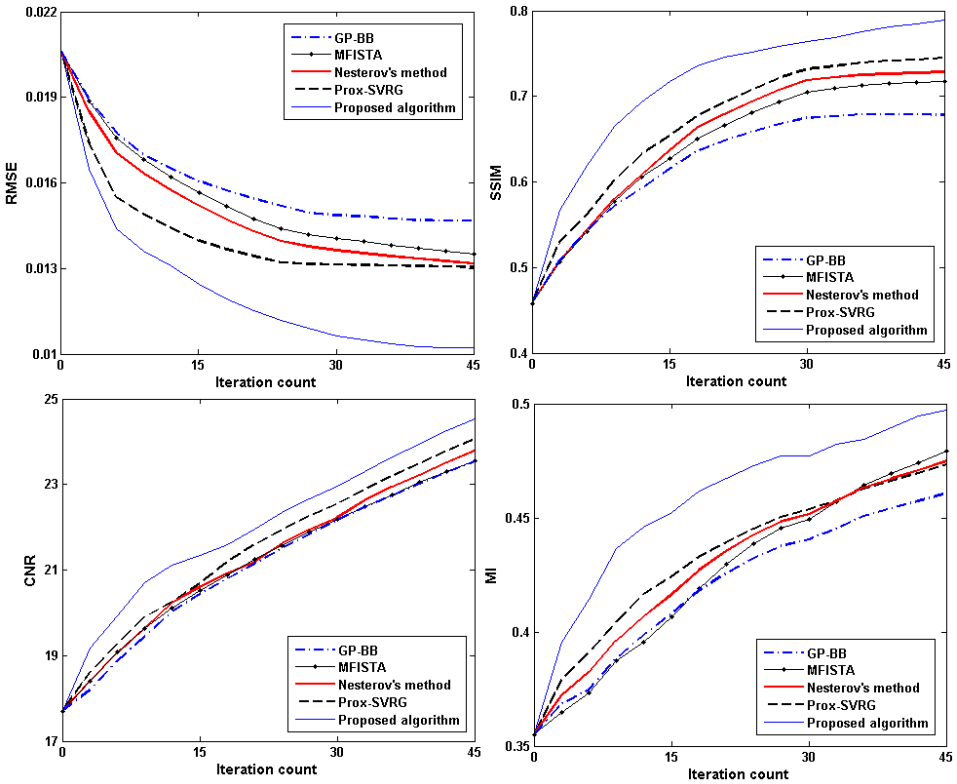

Figure 8.5: Plots of RMSE, SSIM, CNR, and MI for reconstruction of the image of the rat from real CBCT projections using different algorithms.

	Proposed algorithm	Prox-SVRG	Nes-terov	MFISTA	GP-BB	FDK-120	FDK-240
SSIM	0.761	0.732	0.719	0.706	0.675	0.459	0.708
MI	0.478	0.454	0.452	0.450	0.444	0.355	0.459
CNR	22.9	22.5	22.2	22.2	22.1	17.7	22.3

Table 8.3: Performance comparison between different algorithms in reconstruction of the image of the rat after 30 iterations. The numbers next to FDK indicate the number of projections used with that algorithm.

struction with different algorithms after 30 iterations. A comparison of the numbers in this table shows that the image reconstructed by the proposed algorithm has a higher quality than those reconstructed by the other algorithms. The image reconstructed by the proposed algorithm also has higher quality criteria than the image reconstructed with the FDK algorithm using 240 projections.

For a visual comparison, in Figure 8.6 we have shown a representative slice from the images of the rat reconstructed by different algorithms. For a better visual comparison between the images reconstructed with different algorithms, we have selected two regions of interest (ROI) and shown them in zoomed-in views with a narrower window of linear attenuation coefficient, μ. The μ-window for the entire slice is $[0, 0.55]$. The ROI shown on the lower left includes fat surrounded by soft tissue; the μ-window used for displaying this ROI is $[0.14, 0.22]$. The ROI shown on the lower right includes bone surrounded by soft tissue; the μ-window used for displaying this ROI is $[0.15, 0.50]$. It is quite clear, especially from the ROI displayed on the lower left, that the image reconstructed by the proposed algorithm has a higher quality that those reconstructed by the other algorithms.

8.4 Discussion

A comparison of the plots of RMSE and the objective function for our experiments with simulated and real data shows that the proposed algorithm has a much faster convergence to the true or reference image than the other algorithms considered in this study. Plots of the objective image quality criteria show that the proposed algorithm recovers a high-quality image much faster than the other algorithms. In the experiments with the simulated data, the proposed algorithm was able to reconstruct the Shepp-Logan phantom to

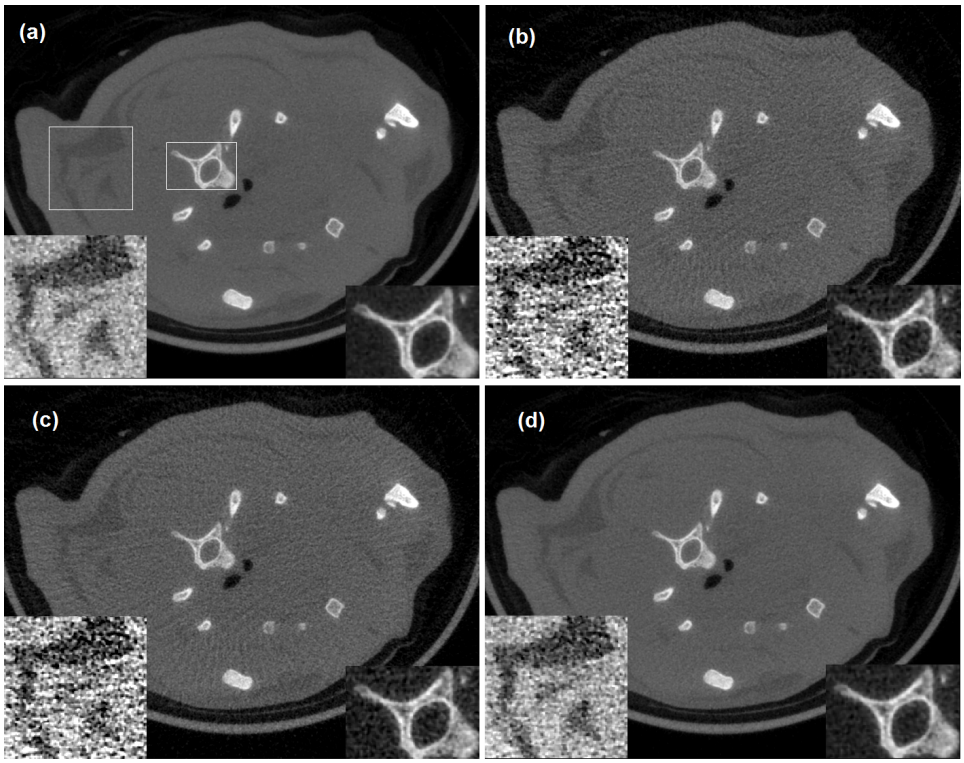

Figure 8.6: A representative slice of the image of the rat reconstructed using different algorithms: (a) The reference image, (b) reconstructed by Prox-SVRG, (c) reconstructed by Nesterov's method, and (d) reconstructed by the proposed algorithm. The zoomed-in views re-displayed with a narrow window of linear attenuation coefficient correspond to the rectangular ROIs shown on the reference image.

a high accuracy from undersampled and noisy measurements. The images reconstructed by the proposed algorithm were superior to the images reconstructed by other algorithms in terms of visual quality and all quantitative criteria used in this study. The same was true for our experiments with the real CBCT projections.

An important observation was the very fast convergence rate of the proposed algorithm in the early iterations. Moreover, the algorithm maintained a good convergence rate with more iterations. In reconstruction from real CBCT projections, the number of iterations of the proposed algorithm to achieve a certain RMSE was approximately 1/3 the number of iterations re-

quired by the other algorithms, as can be seen in Figures 8.3 and 8.5. This can be of high practical value in clinical applications where a fast image recovery is highly desirable.

The Prox-SVRG algorithm was better than MFISTA, the Nesterov's method, and GP-BB, which are all among the best methods for image reconstruction in 3D CT. As shown in Figures 8.3 and 8.5, Prox-SVRG had a good start but its convergence quickly slowed down. Our implementation of Prox-SVRG in this study was identical to that suggested in [336]. We have found that this simple algorithm can be significantly improved if we slightly reduced the step size or the regularization parameter with iteration number. Overall, the performance of our proposed algorithm and Prox-SVRG in this study suggests that variance-reduced SGD methods can form the basis of successful algorithms for image reconstruction in CT.

The value of the regularization parameter λ has a significant influence on the performance of the proposed algorithm as well as the performance of other algorithms considered in this chapter. In general inverse problems, and in CT reconstruction in particular, sometimes a trial-and-error method is used to find a proper value for λ [8, 251]. However, this can be a drawback in practice. There are also systematic methods for determining λ, but most of them are computationally very expensive or apply to a very limited class of problems [102, 323]. A heuristic approach that we followed in this study was to choose a trial value for λ, apply a small number (e.g., 3 to 5) of proximal SGD updates and monitor the change in the values of the two terms of the objective function, i.e., the measurement misfit term and the total variation. To avoid excessive computational costs, we only look at the change in one of the n components of the measurement misfit term. For values of λ that are far from the proper range of values, one or both of the two terms decrease very little. Only for a relatively short range of λ do both terms decrease consistently and we use a value towards the lower end of this range. Even though this approach requires trying several different value of λ, for each value only a small number of proximal SGD updates are applied. Therefore, a range of possible values for λ can be found with relatively little effort. For the low-noise data simulated from the Shepp-Logan phantom, for example, the identified range was $[50, 700]$ and we chose a value of 100. For Prox-SVRG we used the same λ that we used for the proposed algorithm. For MFISTA, Nesterov's method, and GP-BB we started with the value of λ that we used for the proposed algorithm as explained above, but then tried several larger and smaller values and chose the value that gave the best reconstruction results.

In addition to the regularization parameter λ, the proposed algorithm

includes other parameters that can affect its performance and the quality of the reconstructed image. The two most important of these are the step size (α) and the batch size (M). For the step size, we provided a description of the available guidelines in Section 8.2.2. As we mentioned there, for the first phase of the algorithm we use step sizes that are inversely proportional to the Lipschitz constants, L, which is known to be the optimal step size [236], and for the second phase of the algorithm we use a backtracking line search for step size selection. Since these guidelines are based on sound theory, we do not discuss tuning of the step size. Therefore, we focus on λ and M.

In order to study the effect of the choice of λ and M on the algorithm performance, we conducted an experiment on the scan of a rat obtained with the micro-CT scanner. For this scan, a lower tube voltage of 50 kV was used and a 2-mm copper filter was also used to increase the noise level. We used 240 equally-spaced projections from this scan. We applied the proposed algorithm with three different values of $\lambda \in \{75, 150, 350\}$ and three different values of $M \in \{1, 4, 10\}$. Note that the proposed algorithm gradually increases the batch size. Therefore, by M here we mean the batch size at the start of the algorithm. The results of this experiment are presented in Figures 8.7 and 8.8.

In Figure 8.7 we have shown plots of RMSE, CNR, and SSIM for different values of λ and M. In Figure 8.8 we have shown a slice of the image reconstructed with different parameter values. In both of these figures, we have shown the result obtained with the Nesterove's algorithm for comparison. On this dataset, Nesterov'e method performed better than MFISTA and GP-BB and was very close to Prox-SVRG. There are important conclusions that can be drawn from these figures. As expected, the visual and objective quality of the image reconstructed by the proposed algorithm is influenced by the choice of the parameters. The choice of the regularization parameter (λ) affects the convergence behavior of the proposed algorithm and the visual quality of the reconstructed image. A larger λ leads to a smoother image with stronger denoising but a simultaneous blurring and reduction in the sharpness of the edges. On the other hand, a smaller λ leads to a slower convergence in terms of RMSE and reconstruction of an image that is in general rougher. In terms of the batch size (M) the conclusion is more straightforward: using $M = 1$ always lead to the best result. As we argued earlier in this chapter, this is because the projection measurements for different view angles contain much shared information and, hence, the gradient computed based on the projection measurements from different view angles are also highly correlated. Therefore, in the early iterations of the algorithm, it is much more efficient to compute the update directions based

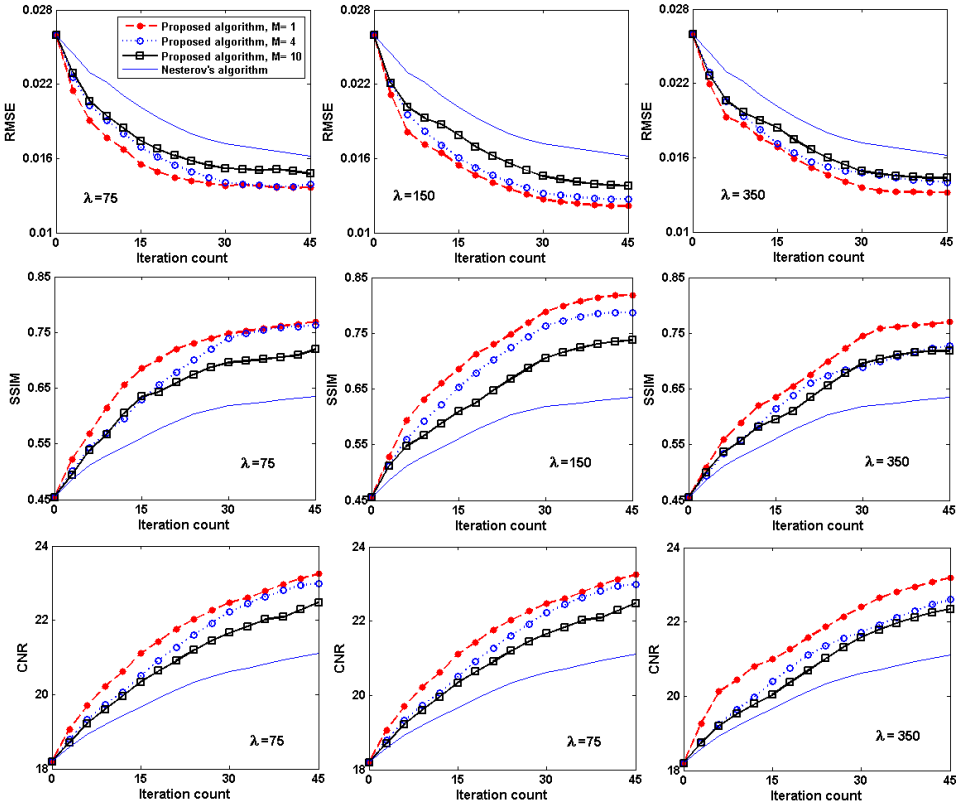

Figure 8.7: Plots of RMSE, SSIM, and CNR for different settings of the regularization parameter, λ, and the batch size at the start of the algorithm, M. The legend for all plots is similar to the one shown on the top-left plot.

on the gradient computed from one projection. Notwithstanding these influences, it is interesting to note that the proposed algorithm still recovers a high-quality image that is visually and quantitatively better than the image reconstructed with the Nesterov's algorithm for a relatively wide range of parameter values.

As we mentioned above, the most computationally expensive part of the proposed algorithm is the forward and back-projection operations, which we have denoted with A_i and A_i^T. These operations accounted for approximately 91% of the computational time. The second most computationally expensive operation was the computation of the proximal operators, denoted with $\mathbf{prox}_{\lambda\mathrm{TV}}(.)$ in Algorithm 6, which accounted for approximately 5% of the computational time. As we mentioned in Section 8.1.2, we used the

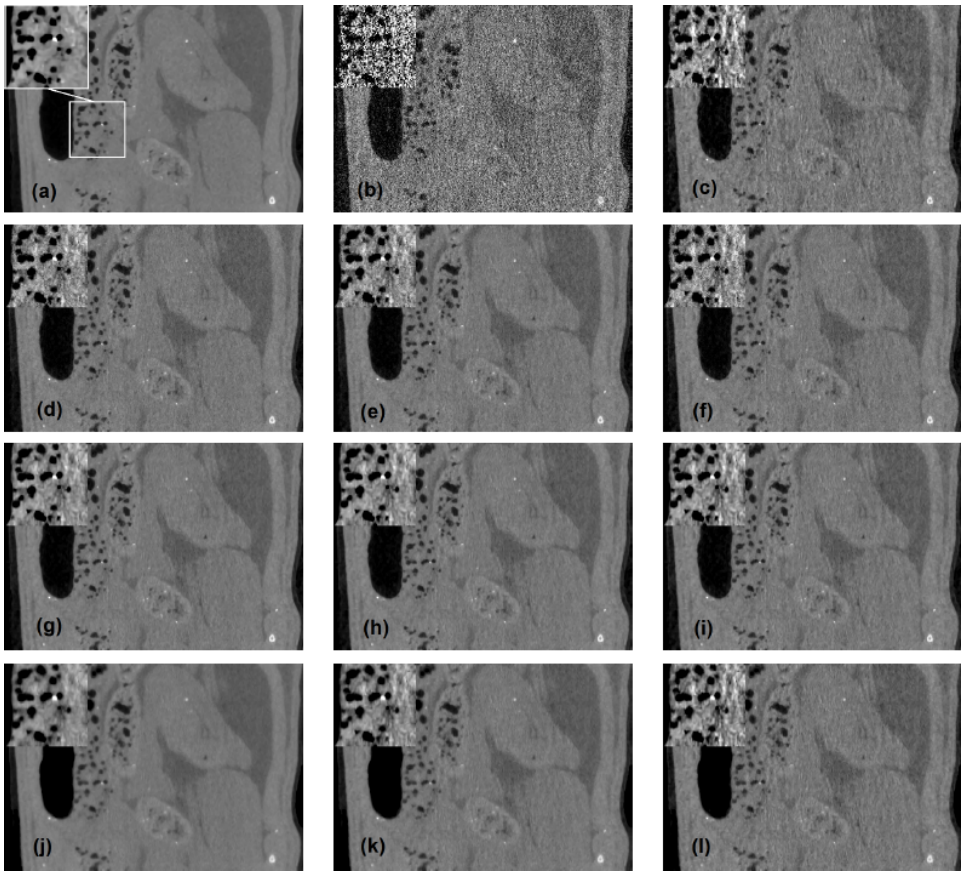

Figure 8.8: Effect of the choice of the regularization parameter, λ, and the batch size at the start of the algorithm, M, on the visual quality of the reconstructed image of a rat. (a) The reference image, (b) FDK-reconstructed, (c) reconstructed using Nesterov's algorithm, and the images reconstructed using the proposed algorithm with different parameter values: (d) $\lambda = 75$, $M = 1$, (e) $\lambda = 75$, $M = 4$, (f) $\lambda = 75$, $M = 10$, (g) $\lambda = 150$, $M = 1$, (h) $\lambda = 150$, $M = 4$, (i) $\lambda = 150$, $M = 10$, (j) $\lambda = 350$, $M = 1$, (k) $\lambda = 350$, $M = 4$, (l) $\lambda = 350$, $M = 10$.

Chambolle's well-known algorithm [46] to compute the proximal operations. Although a large number of iterations of this algorithm can be applied to compute the proximal operation to a high accuracy, we have found that one to three iterations are enough to give the proposed algorithm a good behavior. This experience of ours agrees with the theoretical results developed in [284]. The third most computationally expensive part of the proposed algorithm was the application of the limited-memory BFGS algorithm to find the update direction, the step denoted with $d = H_k\delta$ in Algorithm 6. This computation accounted for approximately 2% of the algorithm time. The rest of operations accounted for approximately 2% of the computational time.

Most of the variance-reduced SGD algorithms proposed in recent years focus on strongly convex functions or have much higher theoretical convergence rates for strongly convex functions [79, 148, 165, 177]. As we mentioned above, when the number of projection measurements is less than the number of unknown voxels (which is almost always the case in sparse-view reconstruction), the CT reconstruction problem is not strongly convex. Nonetheless, our experimental results show that variance-reduced SGD algorithms can lead to efficient CT reconstruction algorithms. It is an open question whether the convergence rate can be improved if the CT reconstruction problem is formulated as a strongly convex optimization problem. A simple approach, also suggested in [206], is to add an ℓ_2 regularization term to the objective function and gradually reduce the strength of this regularization with iterations. We applied this idea to CT reconstruction but did not obtain good results. An entirely different possible approach to improving the algorithm proposed in this chapter is to use acceleration techniques, which have been shown to work well with ordered-subsets method for CT reconstruction in recent years [159, 161].

Chapter 9

Iterative Reconstruction with Nonlocal Regularization

9.1 Introduction

The results of Chapter 8 showed that variance-reduced stochastic gradient descent (VR-SGD) algorithms are a very suitable optimization approach for CT reconstruction. Not only they show very good convergence behavior, they do not need manual tuning of the step size, unlike the basic SGD methods. The algorithm proposed in Chapter 8 improved the basic VR-SGD algorithm by gradually increasing the batch size and exploiting the curvature of the cost function as the image estimate approached a solution. The focus of this Chapter is on the problem regularization. In chapter 8 we relied on TV regularization, which has been used by many studies on CT reconstruction in the past decade. Recent studies, however, have shown that patch-based regularization methods can outperform TV-based regularization methods in image reconstruction. In this chapter, we consider a nonlocal patch-based regularization.

Because the problem of estimating the CT image from few-view and noisy projections is ill-posed, it is critical to properly regularize the reconstruction problem. Most of the published reconstruction algorithms in the past decade have used smoothness-promoting or edge-preserving regularization functions. These regularizers span a wide range of complexity, from simple roughness penalties to non-convex regularizations terms. In general, these regularizers encourage smooth or piecewise-constant solutions by penalizing jumps between neighboring pixels. Algorithms that are based on such regularizers have been successful, and their success has contributed significantly to the growing interest in iterative CT reconstruction. However, CT images usually include fine and texture-like features that are not suitable for reconstruction with these regularizers. Such features usually get blurred or are poorly reconstructed with these algorithms.

As we saw in the review of the literature in Chapter 2, research in the

past ten years has shown that nonlocal patch similarities can be used to devise very powerful models for natural images. Nonlocal patch similarities have been successfully exploited in various image processing tasks. These models are well known for preserving fine image features such as textures and low-contrast edges even in the presence of strong noise. This is because the redundant information in similar image patches will help preserve genuine image features even when these features are not very strong or when the amount of noise is substantial. As we described in Sections 2.2 and 2.6.2, nonlocal patch-based similarities have also been successfully used to regularize various inverse problems including CT reconstruction. Let us consider the linear model with Gaussian noise that we described in Section 8.1.2, i.e., $y = Ax + w$. In this model, y is the vector of sinogram data, A is the projection matrix, and w is the additive noise. To estimate x, it has been suggested to minimize a cost function of the form:

$$F(x) = \frac{1}{2}\|Ax - y\|_2^2 + R_{\mathrm{NL}}(x) \tag{9.1}$$

where the regularization term, $R_{\mathrm{NL}}(x)$, penalizes the difference between the pixel values based on their patch similarity. Various formulations have been proposed for $R_{\mathrm{NL}}(x)$. For example, two common formulations are the following [114, 194]:

$$\begin{aligned} R_{\text{NL-TV}}(x) &= \sum_i \frac{1}{C(i)} \sum_{j \in \mathcal{S}_i} G_a(\|x[j] - x[i]\|) \cdot |x(j) - x(i)| \\ R_{\text{NL-H1}}(x) &= \sum_i \frac{1}{C(i)} \sum_{j \in \mathcal{S}_i} G_a(\|x[j] - x[i]\|) \cdot |x(j) - x(i)|^2 \end{aligned} \tag{9.2}$$

In the above equations, $C(i)$ is a normalization constant, G_a is usually a Gaussian function with bandwidth a, and $\mathcal{S}_i$ is usually a window around the current pixel, $x(i)$, and $x[i]$ is a patch centered on $x(i)$.

Minimization of the cost function in Equation (9.1) is challenging, mainly because of the dependence of the patch similarity weights on the image x. Most of the proposed algorithms find an approximate solution by computing the weights from an initial image estimate [194] or by iteratively updating the weights from the latest image estimate [229, 354]. Moreover, many different approaches have been proposed for minimizing (9.1), including gradient descent [194], proximal gradient methods [255], majorization-minimization [348], graph-cuts methods [114], and Bregman methods [354].

In general, algorithms that include regularization functions that are based on nonlocal patch similarities have been reported to outperform algorithms based on smoothness-promoting regularizations. However, many of the proposed algorithms that use nonlocal patch-based regularization have the following issues:

1. They are only suitable for small-scale problems. In general, patch-based image processing algorithms are known to be very computationally intensive. This is especially the case for processing of large 3D images. Existing approaches for iteratively updating the patch similarity-based weights will be very costly when applied on large 3D images.

2. For the minimization of the measurement misfit term, most of the proposed algorithms use slow methods such as gradient descent or conjugate gradient descent [146, 154, 155, 202, 353]. Hence, most of the proposed algorithms have been evaluated on small 2D images [133, 194, 353].

3. Another limitation of almost all proposed algorithms is that they compute the patch similarity weights from a small window around each pixel. For large 3D images, this window should be very small to keep the computations manageable. However, this is not a good practice because there may be no similar patches in this window, while many similar patches may exist in other parts of the image.

In this chapter, we suggest an iterative CT reconstruction algorithm with nonlocal patch-based regularization that tries to address some of the shortcomings mentioned above. Unlike previous algorithms that use patches from a small window in the same image, we use patches from a high-quality reference image. In this respect, our approach is similar to that in Chapter 5, where we used a low-noise sinogram for interpolating noisy undersampled projections. Moreover, we use a stochastic algorithm to find a small number of similar patches that can come from any location in this reference image. For minimization of the cost function we use a VR-SGD method as we did in Chapter 8.

9.2 Methods

9.2.1 Problem formulation

We propose to estimate the unknown image x as a minimizer of the following cost function:

$$F(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\|A_i x - y_i\|_2^2 + \frac{\lambda_{\mathrm{NL}}}{2}\|x - x_{\mathrm{NL}}\|_2^2 + \lambda_{\mathrm{TV}}\mathrm{TV}(x) \tag{9.3}$$

We have written the measurement misfit term (the first of the three terms in the above objective function) as an average over the projection views, similar to our approach in Chapter 8. In this term, y_i denotes the vector of sinogram measurements in the ith projection view and A_i denotes the projection matrix for that projection view angle. In the first regularization term, x_{NL} is an estimate of x that is computed from a high-quality reference image using a patch-based approach similar to NLM denoising. The second regularization term is the total-variation of x. By using both types of regularizations, we will be able to study the effects of these two types of regularizations separately and jointly.

We compute x_{NL} using the following equation, which is the standard NLM-type formulation.

$$x_{\mathrm{NL}}(i) = \frac{1}{C(i)}\sum_{j\in\mathcal{S}_i} G_a(\|x[i] - x_{\mathrm{ref}}[j]\|_F^2)\cdot x_{\mathrm{ref}}(j) \tag{9.4}$$

Therefore, the regularization function that we suggest is slightly different from the more commonly used forms shown in Equation (9.2). Nonetheless, the justification behind this regularization is the same. Indeed, at least one study has used a similar regularization for CT reconstruction [202].

As mentioned above, in most previous studies the patch similarity weights are computed either from an initial image estimate or from the latest image estimate on the fly. Moreover, the set of patches used to regularize the value of the ith pixel, denoted with $\mathcal{S}_i$ in Equations (9.2) and (9.4), is usually all patches in a small window around that pixel. The proposed algorithm is different in both aspects as we explain below.

Firstly, we compute the patch similarity weights from a high-quality reference image, denoted with x_{ref} in Equation (9.4). Images reconstructed from low-dose scans usually contain much noise and streaking artifacts. Therefore, an initial image estimate, which is usually reconstructed using

a filtered backprojection algorithm, will be a poor choice for computing the patch similarities. One may expect that iteratively updating the patch similarities from the latest image estimate will gradually improve the patch similarity estimates. However, the opposite may happen because strong artifacts in the early image estimates can result in poor patch similarity estimates, further amplifying the artifacts with more iterations of the algorithm [229]. This can be avoided by estimating the patch similarity weights from a high-quality reference image.

Secondly, it is very likely that no patches similar to patch $x[i]$ exist in a small window around the pixel $x_{\text{ref}}(i)$ (or pixel $x(i)$, for that matter). Therefore, instead of defining $\mathcal{S}_i$ to be a window around pixel $x_{\text{ref}}(i)$, we define it to be the set of indices of k patches from the image x_{ref} that are similar to the patch $x[i]$. These patches can be located anywhere in x_{ref}. Therefore, for each pixel $x(i)$, we need to find a set of k patches similar to $x[i]$ in x_{ref}. Because of the very large size of the image, we cannot hope to find the k most similar patches. Therefore, we use a stochastic approach based on the Generalized PatchMatch algorithm [13]. We described the main steps of this algorithm in Section 5.2.1.

We follow the Generalized PatchMatch algorithm, except that we do not use random initializations. Instead, we use a more informed initialization that can lead to much better matches in the early iterations of the algorithm. As mentioned above, for block matching we use a high-quality prior image as the reference image, denoted with x_{ref} in Equation (9.4). This can be the image of the same patient (in situations where a patient is scanned several times) or of a different patient from a database. Let us denote with $\mathcal{S}_i$ the indices of the k patches in x_{ref} that are similar to $x[i]$. If the locations of the organs in the reference image and the image being reconstructed have not shifted much, one can initialize $\mathcal{S}_i$ to a set of random patches in a small window around $x_{\text{ref}}(i)$. However, this will not be a good strategy if the organs have shifted significantly between x and x_{ref}, or if x and x_{ref} are images of two very different patients. In that case, we suggest finding initial values for $\mathcal{S}_i$s using the following two steps. Figure 9.1 shows these steps.

1. Partition x and x_{ref} into *non-overlapping* blocks and run several iterations of the Generalized PatchMatch to find a set of at least k matches in x_{ref} for each (non-overlapping) block in x. Let us denote each of these non-overlapping blocks of x with $x_{\text{NO}}[j]$ and the indices of the set of matching blocks for $x_{\text{NO}}[j]$ with $\mathcal{S}_{\text{NO}}(j)$.

   This step will determine the overall mapping of the location of matching features between x and x_{ref}.

2. For each *overlapping* block in x, initialize $\mathcal{S}_i$ with the help of the findings of the above step. Suppose that the pixel $x(i)$ falls in $x_{\mathrm{NO}}[i_k]$. Then, we can initialize $\mathcal{S}_i$ to be a set of random pixels in $\mathcal{S}_{\mathrm{NO}}(i_k)$. Alternatively, we can do a search in blocks pointed to by $\mathcal{S}_{\mathrm{NO}}(i_k)$ to find a set of matching blocks. Our experience shows that the two approaches lead to comparable results. We used the latter approach in the experiments reported in this chapter.

Note that because in this initialization approach (specifically, in step 1 above) we run the Generalized PatchMatch for non-overlapping patches, its cost will be very low.

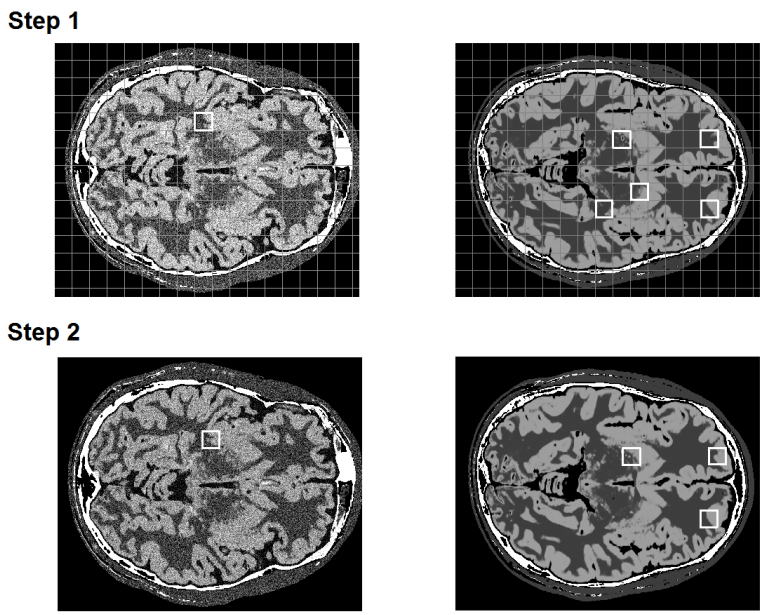

Figure 9.1: The proposed initialization for the Generalized PatchMatch algorithm. Step 1: The source image (left) and the reference image (right) are partitioned into non-overlapping blocks. For each block in the source image, at least k matching blocks are found in the reference image. Step 2: for each block in the source image, k matching blocks are found in the reference image with the help of the mapping discovered in Step 1. In this simple illustration, 5 matching blocks are found in Step 1 and then $k = 3$ blocks are identified in Step 2.

9.2.2 Optimization algorithm

We start by rewriting the proposed cost function in Equation (9.3) as:

$$F(x) = G(x) + R(x) \tag{9.5}$$

Assuming that x_{NL} is constant, the first term, $G(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\|A_i x - y_i\|_2^2 + \frac{\lambda_{\mathrm{NL}}}{2}\|x - x_{\mathrm{NL}}\|_2^2$, is convex and differentiable. The second term, $R(x) = \lambda_{\mathrm{TV}}\mathrm{TV}(x)$, is convex but non-differentiable. We would like to minimize the proposed cost function using a proximal VR-SGD approach. To this end, we rewrite the cost function as:

$$\begin{aligned} F(x) &= \frac{1}{n}\sum_{i=1}^{n} g_i(x) + R(x) \\ &= \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{2}\|A_i x - y_i\|_2^2 + \frac{\lambda_{\mathrm{NL}}}{2}\|x - x_{\mathrm{NL}}\|_2^2\right) + \lambda_{\mathrm{TV}}\mathrm{TV}(x) \end{aligned} \tag{9.6}$$

We suggest Algorithm 7 for minimizing this cost function. The update direction in this algorithm is very similar to the basic VR-SGD update that we described in Chapter 8. The only difference is that we weight the full gradient direction ($\tilde{\mu}$) using the average of the Lipschitz constants, as suggested in [336]. We use the algorithm proposed in [46] for computing the proximal operation for the TV regularization term.

In Algorithm 7, L_i denotes the Lipschitz constant of g_i, which is equal to $\lambda_i + \lambda_{\mathrm{NL}}$, where λ_i is the largest eigenvalue of $A_i^T A_i$. $L_{\mathrm{mean}} = \lambda_{\mathrm{mean}} + \lambda_{\mathrm{NL}}$ denotes the average of L_is, where λ_{mean} is the average of the λ_is, and α is the step size. Let us also denote the strong convexity parameter of the whole cost function, $F(x)$, with μ. This means that for all x and y in the domain of F we have $F(x) \geq F(y) + z^T(x-y) + \frac{\mu}{2}\|x-y\|_2^2$, where z is any subgradient of F at y. For the proposed cost function in Equation (9.6) we have $\mu \geq \lambda_{\mathrm{NL}}$, and in most cases of interest where A is a wide matrix (i.e., more unknown image voxels than sinogram measurements) we have $\mu = \lambda_{\mathrm{NL}}$. From Theorem 1 in [336], if we choose α and λ_{NL} such that the value of r given below is less than 1, then the proposed algorithm will have a geometric convergence with rate r. This means that $\mathbb{E}(F(x_n)) - F(x^*) \leq r^n\big(F(x_0) - F(x^*)\big)$, where x^* is the global minimizer of F.

$$r \simeq \frac{\frac{\lambda_{\mathrm{mean}}}{\lambda_{\mathrm{NL}}} + 1}{2\alpha(1-4\alpha)n} + \frac{4\alpha}{1-4\alpha} \tag{9.7}$$

input : initial image estimate x_0 (obtained using the FDK algorithm).
output: final image estimate after N iterations, x_N.
option 1: compute x_{NL} based on x_0 by performing several iterations of the Generalized PatchMatch
for $j \leftarrow 1$ **to** N **do**
  $\tilde{x} = x_{j-1}$
  option 2: update x_{NL} based on $\tilde{x}$ by performing one iteration of the Generalized PatchMatch
  $\tilde{\mu} = \frac{1}{n} A^T(A\tilde{x} - y)) + \lambda_{\mathrm{NL}}(\tilde{x} - x_{\mathrm{NL}})$
  $x_j^0 = \tilde{x}$
  for $k \leftarrow 1$ **to** $2n$ **do**
    select $i_k \in \{1, ..., n\}$ with probability $\frac{L_i}{\sum_i L_i}$
    $x_j^k = \mathrm{prox}_{\alpha\lambda_{\mathrm{TV}}\mathrm{TV}}\Big(x_j^{k-1} - \alpha\big(\frac{1}{L_{i_k}} A_{i_k}^T A_{i_k}(x_j^{k-1} - \tilde{x}) + \frac{1}{L_{\mathrm{mean}}}\tilde{\mu}\big)\Big)$
  end
  $x_j = x_j^{2n}$
end

Algorithm 7: The proposed proximal VR-SGD algorithm for CT reconstruction by minimizing Equation (9.6).

It is always possible to choose α and λ_{NL} such that $r < 1$. However, the step size suggested by this analysis is always less than 1/4, usually around 0.1 or smaller. As we will see later, our experiments show that larger step sizes result in faster convergence in our application.

In the above analysis, we assumed that x_{NL} was constant. In order to obey this assumption, we can compute x_{NL} from x_0 using the Generalized PatchMatch before the start of the algorithm. Alternatively, we can continually update x_{NL} by performing one iteration of the Generalized PatchMatch in each iteration of the proposed algorithm. Both of these options have been shown in Algorithm 7, labeled Option 1 and Option 2, respectively. The theoretical convergence rate mentioned above does not apply if we choose Option 2, i.e., if x_{NL} is updated based on the current estimate of x. Nonetheless, Option 2 is intuitively better than option 1. This is because with more iterations of the main algorithm, noise and artifacts in the image estimate are reduced and, therefore, x_{NL} will be closer to the true image. Therefore, it makes more sense to gradually improve x_{NL} than to spend much effort to estimate it from x_0 at the algorithm start. Our results, which we will present in the next section, support this intuition.

9.3 Results and Discussion

9.3.1 Simulated data

We simulated 720 noisy projections from a digital brain phantom, which we obtained from the BrainWeb database [63]. Figure 9.2(a) shows a slice of this phantom. Figure 9.2(b) shows a slice from a different brain phantom, obtained from the same database, which we used as x_{ref} for block matching to compute x_{NL} (see Equation (9.4) and Figure 9.1). The rest of the images in Figure 9.2 show the images reconstructed with different algorithms. For comparison, we have used FDK and MFISTA algorithms. For the proposed algorithm, we have shown the reconstruction results with two different values of $\lambda_{\text{TV}} = \{0, 200\}$ and two different values of $\lambda_{\text{NL}} = \{0, 0.01\}$. The case with $\lambda_{\text{TV}} = 0$ corresponds to using patch-based regularization only; similarly, the case with $\lambda_{\text{NL}} = 0$ corresponds to using TV regularization only. We used $\alpha = 1$ in all experiments in this chapter.

A visual comparison of the reconstruction results in Figure 9.2 suggests that both regularization terms contribute positively to the quality of the reconstructed images. However, the patch-based regularization alone has resulted in a better image than the TV regularization alone. In particular, reconstruction with $(\lambda_{\text{TV}} = 0, \lambda_{\text{NL}} = 0.01)$ seems to have better preserved the image sharpness than reconstruction with $(\lambda_{\text{TV}} = 200, \lambda_{\text{NL}} = 0)$. The objective image quality criteria summarized in Table 9.1 supports this statement. Figure 9.2 and Table 9.1 are for reconstruction after 15 iterations.

	FDK	MFISTA	Proposed algorithm $\lambda_{\text{TV}} = 200$ $\lambda_{\text{NL}} = 0$	Proposed algorithm $\lambda_{\text{TV}} = 0$ $\lambda_{\text{NL}} = 0.01$	Proposed algorithm $\lambda_{\text{TV}} = 200$ $\lambda_{\text{NL}} = 0.01$
RMSE	0.068	0.039	0.032	0.027	0.026
SSIM	0.710	0.745	0.750	0.785	0.785
CNR	14.6	19.3	21.0	21.2	21.3

Table 9.1: Image quality metrics for the experiment with simulated data.

Figure 9.3(a) shows the RMSE plots for different values of the regularization parameters for up to 30 iterations. These plots show that patch-based regularization results in much better convergence, especially as the number of iterations increases. When $\lambda_{\text{NL}} = 0.01$, the presence of the TV regularization leads to faster initial convergence, but has no significant added positive effect after the first few iterations.

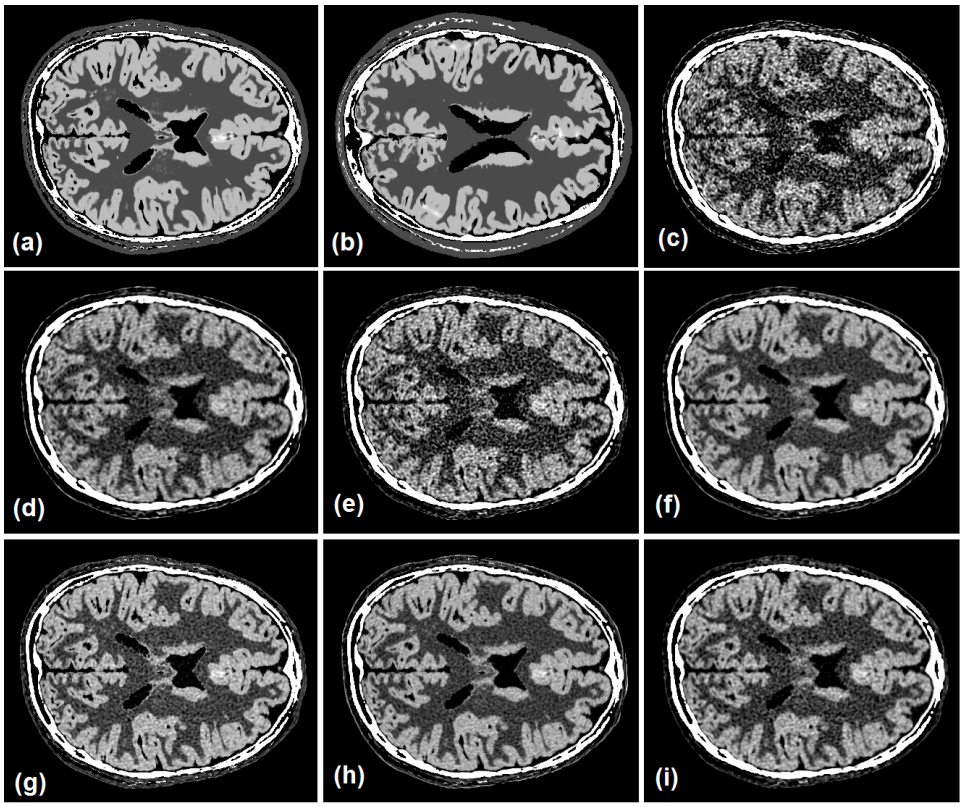

Figure 9.2: Reconstruction results of the experiment with the brain phantom. (a) the reference image, (b) a slice of the brain phantom used for block matching, x_{ref}, (c) FDK, (d) MFISTA, (e) proposed algorithm with $(\lambda_{\text{TV}} = 0, \lambda_{\text{NL}} = 0)$, (f) proposed algorithm with $(\lambda_{\text{TV}} = 200, \lambda_{\text{NL}} = 0)$, (g) proposed algorithm with $(\lambda_{\text{TV}} = 0, \lambda_{\text{NL}} = 0.01)$, (h) proposed algorithm with $(\lambda_{\text{TV}} = 200, \lambda_{\text{NL}} = 0.01)$, (i) proposed algorithm with $(\lambda_{\text{TV}} = 200, \lambda_{\text{NL}} = 0.01)$ with reconstruction of the patch-based image estimate, x_{NL}, at the start of the algorithm (Option 1 in Algorithm 7).

The RMSE plots shown in Figure 9.3(a) and the images in Figure 9.2(e)-(h) were obtained by using Option 2 in Algorithm 7. In Figure 9.2(i) we have shown the image reconstructed using Option 1. This image clearly contains artifacts that do not exist in Figure 9.2(e)-(h). To better understand why Option 2 leads to better results, in Figure 9.3(b) we comparethe two options in terms of the RMSE of the reconstructed image and the RMSE of x_{NL}. For Option 1, we applied 30 iterations of the Generalized PatchMatch algorithm before the start of the algorithm, whereas for Option 2 we applied one iteration of the Generalized PatchMatch at the beginning of each iteration of the proposed algorithm. With Option 1, x_{NL} is created from the initial image estimate (x_0) and, hence, its error is fixed. The quality of x_{NL} generated in Option 1 is not very high, as indicated by its relatively high RMSE. Therefore, patch-based regularization with Option 1 does not lead to fast convergence. In fact, after the initial iterations, regularization in terms of proximity with x_{NL} hurts the algorithm convergence because it forces the image estimate to remain close to x_{NL}. On the other hand, the quality of x_{NL} generated with Option 2 continues to improve with more iterations of the algorithm because it is updated using the latest image estimate. Therefore, patch-based regularization with Option 2 constantly pushes the image estimate towards a lower RMSE. In the experiments with real data reported in the next section, we will only show the results obtained with Option 2.

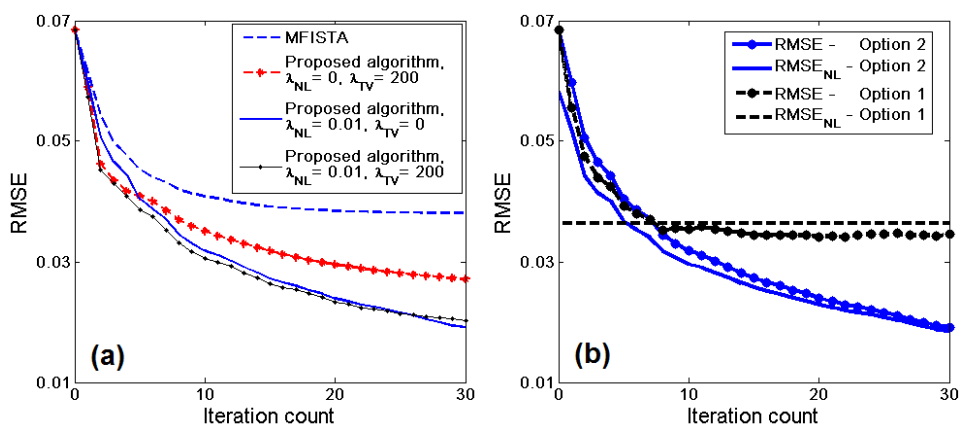

Figure 9.3: (a) RMSE plots for reconstruction of the brain phantom. (b) Comparison between the two approaches for estimating the nonlocal patch-based image estimate, x_{NL}; as shown in Algorithm 7, in Option 1 x_{NL} is estimated before the start of the image reconstruction algorithm, whereas in Option 2 x_{NL} is iteratively updated based on the latest image estimate.

9.3.2 Real data

We evaluated the proposed algorithm on the micro-CT scan of a rat. The scan consisted of 720 projections, all of which were used to reconstruct a reference image. The proposed algorithm was then applied to reconstruct an image from a subset of 180 projections from this scan.

Figure 9.4 shows slices of the reconstructed image of the rat after 15 iterations of the proposed algorithm. In the same figure, we have also shown two profiles, the locations of which have been marked with the line segments $L1$ and $L2$ in Figure 9.4(a). This figure shows that the nonlocal patch-based regularization has resulted in a marked improvement in the quality of the reconstructed image, especially of the fine details. The plots of RMSE for this experiment were very similar to that for the simulation experiment shown in Figure 9.3, and hence they are omitted. Table 9.2 shows a summary of the objective image quality for this experiment.

	FDK	MFISTA	Proposed algorithm $\lambda_{\text{TV}} = 300$ $\lambda_{\text{NL}} = 0$	Proposed algorithm $\lambda_{\text{TV}} = 0$ $\lambda_{\text{NL}} = 0.04$	Proposed algorithm $\lambda_{\text{TV}} = 300$ $\lambda_{\text{NL}} = 0.04$
RMSE	0.0170	0.0136	0.0122	0.0109	0.0105
SSIM	0.604	0.731	0.755	0.781	0.770

Table 9.2: Image quality metrics for the experiment with real data.

An important setting with the patch-based regularization is the value of the regularization parameter, λ_{NL}. Our extensive experiments show that this value should be approximately equal to or smaller than the smallest λ_i. Recall that λ_i is the largest eigenvalue of A_i, the projection matrix for the ith projection view. For our simulation experiment with the brain phantom, for example, λ_i ranged from 0.32 (for projection view angles that were integer multiples of $\pi/2$) to 0.43 (for projection view angles that were odd multiples of $\pi/4$) and values of the regularization parameter in the range $\lambda_{\text{NL}} \in [0.05, 0.25]$ gave us good results. In the experiment with the rat scan λ_i ranged between 0.087 and 0.124, and values of λ_{NL} in the range $[0.02, 0.06]$ gave us good results.

Overall, our experience with the algorithms proposed in Chapter 8 and this chapter shows that when approximately 100 to 200 noisy projections are used for reconstruction, TV regularization alone is enough to faithfully reconstruct large image features, provided that a good value is selected for the regularization parameter. On the other hand, using only TV regulariza-

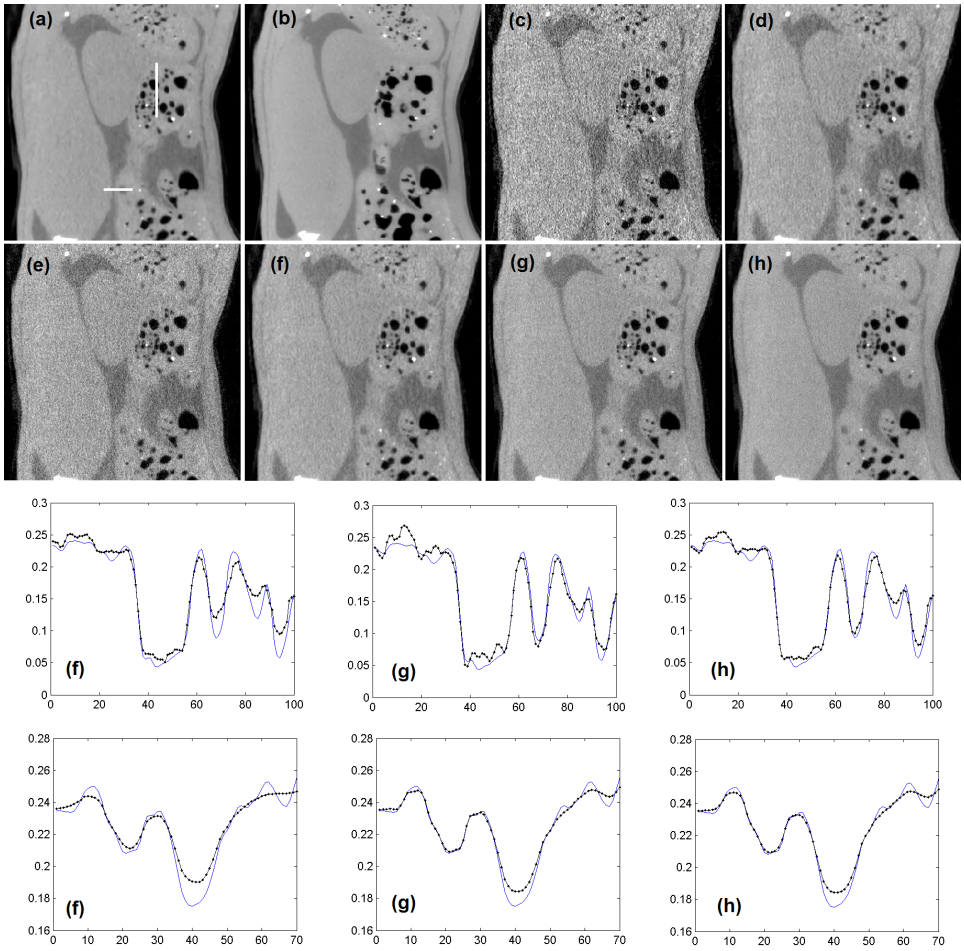

Figure 9.4: Reconstruction results of the experiment with the rat scan. (a) the reference image, (b) the corresponding slice of the image used for block matching, x_{ref}, (c) FDK- reconstructed, (d) MFISTA, (e) proposed algorithm with $(\lambda_{\text{TV}} = 0, \lambda_{\text{NL}} = 0)$, (f) proposed algorithm with $(\lambda_{\text{TV}} = 300, \lambda_{\text{NL}} = 0)$, (g) proposed algorithm with $(\lambda_{\text{TV}} = 0, \lambda_{\text{NL}} = 0.04)$, (h) proposed algorithm with $(\lambda_{\text{TV}} = 300, \lambda_{\text{NL}} = 0.04)$. The lower panel shows two small profile segments in the images reconstructed by the proposed algorithm with three different regularization parameter settings. The locations of these profiles have been marked in the reference image with white line segments.

tion, faithful reconstruction of fine image features is much more difficult and sometimes impossible within a small number of iterations, even after careful tuning of the regularization parameter. Such features can be reconstructed by increasing the number of iterations. On the other hand, fine image features are easier to reconstruct by using nonlocal patch-based regularization. We showed an example of the effectiveness of patch-based regularization for recovery of fine features in Figure 9.4. The success of patch-based regularization in reconstructing fine features is because they exploit the redundant information in similar patches, thereby preserving fine edges and textures that are normally blurred by TV-based regularization. We can say that TV is a global regularization function, treating all parts of the image with equal strength, determined by the value of the regularization parameter. Therefore, if we choose a sufficiently large regularization parameter to ensure strong noise suppression, some fine image features could be lost. Nonlocal patch-based regularization, on the other hand, treats each image location differently by finding similar patches that can help preserve and amplify the local features. Therefore, nonlocal patch-based regularization can recover fine image features much easier than TV regularization. As expected, this advantage of patch-based regularization comes with certain costs. In particular, the additional computational and memory requirements can be significant, especially for large-scale images. The algorithm proposed in this chapter also requires a high-quality reference image, which is not always available.

Chapter 10

Conclusions

X-ray computed tomography (CT) has become one of the most essential and widely-used tools in medicine. As its usage continues to increase, the need to maintain the radiation dose at a reasonably safe level becomes even more important. Therefore, in order for CT to fulfill the growing demands, the image reconstruction and processing algorithms should be greatly improved.

This dissertation investigated the potential of some of the powerful concepts and tools in image processing and optimization for image reconstruction and processing in CT. These included patch-based image models, total variation, and variance-reduced stochastic optimization algorithms. We proposed new algorithms for denoising and interpolation of CT measurements, denoising and restoration of CT images, and for iterative CT reconstruction. Experiments with simulated and real CT data showed that the proposed algorithms can compete with, and often outperform, some of the state of the art algorithms.

10.1 Contributions of this dissertation

This section summarizes the main findings and contributions of this dissertation under three categories of pre-processing, post-processing, and iterative reconstruction algorithms.

10.1.1 Pre-processing algorithms

Only a very small fraction of the published algorithms for CT have focused on processing the measured CT projections (i.e., the sinogram). The results of this dissertation show that denoising and interpolation of the CT projection measurements can result in substantial improvements in the quality of the reconstructed CT images.

- The results of Chapter 3 show that patch-based methods can be used to devise very effective sinogram denoising methods. To the best of our knowledge, this is the first study that exploits both the nonlocal

patch similarities and sparse representation in learned dictionaries for sinogram denoising.

- The results of Chapter 4 show that effective sinogram denoising algorithms can be designed based on total variation minimization. We suggested two approaches to account for the signal-dependent nature of the noise and the smooth nature of the projection measurements. We are unaware of any previously-published studies to suggest any of these approaches. Both of these approaches proved to be effective in experiments with low-dose CT projections.

- In general, the patch-based method proposed in Chapter 3 leads to better results than the TV-based algorithms proposed in Chapter 4. The advantage of TV-based denoising methods is that they are in general much faster than patch-based denoising methods. The patch-based method proposed in Chapter 3 is fast but it can also include a dictionary learning step that can add substantially to the computational time. The dictionary has to be re-trained every time scan geometry or the angular spacing between successive projections changes. On the other hand, a shortcoming of the TV-based denoising algorithms is that they involve regularization parameters that need to be tuned carefully in order to obtain good results.

- Two very important properties of CT projection measurements are smoothness and self-similarity. The results of Chapter 5 show that these properties can be exploited to effectively interpolate and denoise noisy undersampled projections, leading to a large improvement in the quality of low-dose CT images. To the best of our knowledge, no previous study has used nonlocal patch similarities for interpolation of CT projections.

10.1.2 Post-processing algorithms

Post-processing of low-dose CT images is very challenging because of the presence of artifacts and strong noise with unknown and spatially-varying distribution. This dissertation focused on using sparse representation in learned dictionaries for low-dose CT image denoising and restoration. Our results show that learned overcomplete dictionaries are effective in denoising and restoration of low-dose CT images.

- In Chapter 6, we proposed a method for removing streak artifacts that arise in images reconstructed from a small number of projections. To

the best of our knowledge, this algorithm is the first algorithm to use coupled dictionaries for artifact suppression in CT images. The results of that chapter show that the proposed algorithm substantially reduces the artifacts without degrading the true image features.

- The two-level dictionary structure proposed in Chapter 7 aimed at removing the noise or artifacts that had very different shapes than the genuine image features, unlike the streak artifacts considered in Chapter 6. This dictionary structure combined the advantages of analytical and learned dictionaries. In our experiments, the proposed dictionary structure effectively suppressed the noise and ring artifacts in low-dose CT images, achieving results that were comparable with or better than standard dictionary-based processing.

10.1.3 Iterative reconstruction algorithms

Two iterative reconstruction algorithms were proposed in this dissertation. An important feature of both algorithms was the use of variance-reduced stochastic gradient descent (VR-SGD) methods. To the best of our knowledge, VR-SGD algorithms have never been used for CT reconstruction before. Our results show that VR-SGD algorithms can be used to build very efficient CT reconstruction algorithms. Although the method of ordered subsets has long been used for CT reconstruction, our results indicate that VR-SGD algorithms show a very good convergence, especially as the image estimate becomes close to a solution. Moreover, step size selection for VR-SGD algorithms is much easier than for the ordered subsets method and there is no need to reduce the step size or increase the batch size.

- The hybrid stochastic-deterministic algorithm that we proposed in Chapter 8 further improved the VR-SGD method by exploiting the curvature of the cost function as the image estimate became close to a solution. Our results showed that this algorithm performed better than the basic proximal VR-SGD.

- The results of Chapter 9 show that regularization in terms of nonlocal patch similarities can be used to develop very effective CT reconstruction algorithms. Algorithms that use patch-based regularization are more computationally intensive than algorithms that use edge-preserving regularizations, such as the algorithm proposed in Chapter 8. However, fine image features such as texture, small features, and

low-contrast edges are much better preserved by exploiting nonlocal patch similarities.

Implementation of forward and back-projection operations on GPU can reduce the per-iteration cost of iterative reconstruction algorithms by large factors. Nonetheless, in order to make the reconstruction time of large 3D images clinically acceptable, the number of iterations needs to be reduced too. This dissertation showed that VR-SGD methods offer an efficient approach towards achieving this goal.

10.2 Future work

The methods proposed in this dissertation can be improved in various ways. There are also many related research directions that have not been explored in this dissertation. In this section, some of these potential research topics are pointed out.

10.2.1 Pre-processing algorithms

- The sinogram denoising and interpolation algorithms proposed in this dissertation were based on simplified noise models. It is well known that in low-dose CT, these models are less accurate. Therefore, it will be very important to study how the performance of the proposed algorithms is affected by the accuracy of the noise model.
- Some of the studies on patch-based Poisson denoising that we reviewed in Section 2.4 have focused on extremely low Poisson counts. The results of some of these studies have been very impressive. It would be very interesting to investigate the significance of these results for very low-dose CT.
- An important limitation of the image processing methods that are based on nonlocal patch similarities is the computational cost of finding similar patches. Therefore, most algorithms employ small patch sizes to reduce the computational cost. As we showed in Chapter 3, for CT projections it is possible to project large patches/blocks into much smaller spaces. This possibility will pose several important research questions. For example, it will be interesting to know how the performance of patch-based denoising and interpolation algorithms such as those proposed in this dissertation will be affected when much larger patch sizes are used.

10.2.2 Post-processing algorithms

- This dissertation uses learned dictionaries for suppressing two types of artifacts, i.e., streaking artifacts that arise when the number of projections is small (Chapter 6) and ring artifacts (Chapter 7). Artifacts in CT images can originate from various sources and have different shapes [14]. These artifacts can be very strong and can thus significantly reduce the image quality. In many cases, artifacts have very different geometrical and statistical properties than the true image features. In that case, a simple algorithm such as the dictionary-based processing method proposed in 7 may be able to significantly reduce the artifacts. More often, however, artifacts have much similarities with the true image features. For example, this is true for the streak artifacts that we studied in Chapter 6. In such cases, more sophisticated algorithms will be needed to suppress the artifacts without damaging the true image features. The results of this dissertation suggest that learned dictionaries may be able to reduce other types of artifacts also.

10.2.3 Iterative reconstruction algorithms

- All dictionary-based iterative CT reconstruction algorithms that we are aware of, including all those reviewed in Chapter 2, have been proposed for 2D CT. For reconstruction of large 3D images these methods are not efficient because they require access to individual elements of the system matrix and this is not possible to do efficiently for large 3D images because the system matrix will be too large to save in the computer memory. Therefore, there is a need for efficient optimization methods that can solve iterative dictionary-based reconstruction problems such as Equation (2.36) for large-scale 3D images. This topic was not addressed in this dissertation. However, there are existing methods that could be employed for solving this problem, for example the recently-proposed plug-and-play approach [271, 322].

- The method of ordered subsets, which is equivalent to incremental/stochastic gradient descent, has long been used to accelerate various iterative reconstruction algorithms in CT [150, 220]. In recent years, this method has been employed in designing some of the state of the art algorithms by combining it with other optimization techniques such as momentum [160, 161, 240]. Our results show that VR-SGD has certain important advantages over the method of ordered subsets. Therefore, it would be very interesting to investigate whether VR-SGD methods

could also be combined with other optimization techniques to achieve faster reconstruction.

Bibliography

[1] Michal Aharon and Michael Elad. Sparse and redundant modeling of image content using an image-signature-dictionary. *SIAM Journal on Imaging Sciences*, 1(3):228–247, 2008.

[2] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54(11):4311–4322, 2006.

[3] Michal Aharon, Michael Elad, and Alfred M. Bruckstein. On the uniqueness of overcomplete dictionaries, and a practical way to retrieve them. *Linear Algebra and its Applications*, 416(1):48 – 67, 2006.

[4] A AlAfeef, P Cockshott, I MacLaren, and S McVitie. Compressed sensing electron tomography using adaptive dictionaries: a simulation study. *Journal of Physics: Conference Series*, 522(1):012021, 2014.

[5] J.B. Allen and L. Rabiner. A unified approach to short-time fourier analysis and synthesis. *Proceedings of the IEEE*, 65(11):1558–1564, Nov 1977.

[6] Francois Alter, Yasuyuki Matsushita, and Xiaoou Tang. An intensity similarity measure in low-light conditions. Association for Computing Machinery, Inc., March 2006.

[7] F. J. Anscombe. The transformation of poisson, binomial and negative-binomial data. *Biometrika*, 35(3/4):246–254, 1948.

[8] Richard C Aster, Clifford H Thurber, and Brian Borchers. *Parameter estimation and inverse problems*, volume 90. Academic Press, 2005.

[9] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Optimization with sparsity-inducing penalties. *Foundation and Trends in Machine Learning*, 4(1):1–106, 2012.

[10] Francis Bach and Eric Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate O(1/n). In C.j.c. Burges,

L. Bottou, M. Welling, Z. Ghahramani, and K.q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 773–781. 2013.

[11] Ti Bai, Xuanqin Mou, Qiong Xu, and Yanbo Zhang. Low-dose ct reconstruction based on multiscale dictionary. In *SPIE Medical Imaging*, pages 86683L–86683L. International Society for Optics and Photonics, 2013.

[12] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), August 2009.

[13] Connelly Barnes, Eli Shechtman, DanB. Goldman, and Adam Finkelstein. The Generalized PatchMatch correspondence algorithm. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision ECCV 2010*, volume 6313 of *Lecture Notes in Computer Science*, pages 29–43. Springer Berlin Heidelberg, 2010.

[14] Julia F. Barrett and Nicholas Keat. Artifacts in CT: Recognition and avoidance. *RadioGraphics*, 24(6):1679–1691, 2004.

[15] Borsdorf A. Kstler H. Rubinstein R. Bartuschat, D. and M. Strmer. A parallel K-SVD implementation for CT image denoising. Technical Report CS 10, University of Erlangen-Nrnberg, 2009.

[16] M.J. Bastiaans. Gabor's expansion of a signal into gaussian elementary signals. *Proceedings of the IEEE*, 68(4):538–539, April 1980.

[17] Amir Beck and Marc Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *Image Processing, IEEE Transactions on*, 18(11):2419–2434, 2009.

[18] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Img. Sci.*, 2(1):183–202, March 2009.

[19] Marcel Beister, Daniel Kolditz, and Willi A. Kalender. Iterative reconstruction methods in x-ray CT. *Physica Medica*, 28(2):94 – 108, 2012.

[20] Anthony J. Bell and Terrence J. Sejnowski. The independent components of natural scenes are edge filters. *Vision Research*, 37(23):3327 – 3338, 1997.

[21] Martin Benning, Christoph Brune, Martin Burger, and Jahn Müller. Higher-order tv methods—enhancement via bregman iteration. *Journal of Scientific Computing*, 54(2):269–310, 2012.

[22] Frank Bergner, Timo Berkus, Markus Oelhafen, Patrik Kunz, Tinsu Pan, Rainer Grimmer, Ludwig Ritschl, and Marc Kachelrie. An investigation of 4d cone-beam ct algorithms for slowly rotating scanners. *Medical Physics*, 37(9), 2010.

[23] Maïtine Bergounioux and Loic Piffet. A second-order model for image denoising. *Set-Valued and Variational Analysis*, 18(3):277–306, 2010.

[24] A. Berrington de Gonzlez, M. Mahesh, K. Kim, M. Bhargavan, R. Lewis, F. Mettler, and C. Land. Projected cancer risks from computed tomographic scans performed in the United States in 2007. *Archives of Internal Medicine*, 169(22):2071–2077, 2009.

[25] Dimitri P. Bertsekas. A new class of incremental gradient methods for least squares problems. *SIAM J. on Optimization*, 7(4):913–926, April 1997.

[26] Dimitri P Bertsekas. *Nonlinear programming.* Athena Scientific, 1999.

[27] H. Bhujle and S. Chaudhuri. Novel speed-up strategies for non-local means denoising with patch and edge patch based dictionaries. *Image Processing, IEEE Transactions on*, 23(1):356–365, Jan 2014.

[28] Junguo Bian, Jeffrey H Siewerdsen, Xiao Han, Emil Y Sidky, Jerry L Prince, Charles A Pelizzari, and Xiaochuan Pan. Evaluation of sparse-view reconstruction from flat-panel-detector cone-beam CT. *Physics in Medicine and Biology*, 55(22):6575, 2010.

[29] J.M. Bioucas-Dias and M. A T Figueiredo. A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration. *Image Processing, IEEE Transactions on*, 16(12):2992–3004, 2007.

[30] Thomas Blumensath and Mike E Davies. Gradient pursuits. *Signal Processing, IEEE Transactions on*, 56(6):2370–2382, 2008.

[31] Lon Bottou. Stochastic gradient descent tricks. In Grgoire Montavon, GeneviveB. Orr, and Klaus-Robert Mller, editors, *Neural Networks: Tricks of the Trade*, volume 7700 of *Lecture Notes in Computer Science*, pages 421–436. Springer Berlin Heidelberg, 2012.

[32] C. Bouman and K. Sauer. A generalized gaussian image model for edge-preserving map estimation. *Image Processing, IEEE Transactions on*, 2(3):296–310, Jul 1993.

[33] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[34] Kristian Bredies, Karl Kunisch, and Thomas Pock. Total generalized variation. *SIAM Journal on Imaging Sciences*, 3(3):492–526, 2010.

[35] Alex Bronstein, Pablo Sprechmann, and Guillermo Sapiro. Learning efficient structured sparse models. *arXiv preprint arXiv:1206.4649*, 2012.

[36] Ori Bryt and Michael Elad. Compression of facial images using the k-svd algorithm. *Journal of Visual Communication and Image Representation*, 19(4):270 – 282, 2008.

[37] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490–530, 2005.

[38] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. Image enhancement by non-local reverse heat equation. *Preprint CMLA*, 22:2006, 2006.

[39] Egbert Buhr, Susanne Günther-Kohfahl, and Ulrich Neitzel. Simple method for modulation transfer function determination of digital imaging detectors from edge images. In *Medical Imaging 2003*, pages 877–884. International Society for Optics and Photonics, 2003.

[40] Cesar F Caiafa and Andrzej Cichocki. Multidimensional compressed sensing and their applications. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 3(6):355–380, 2013.

[41] EJ Candes and DL Donoho. Curvelets and reconstruction of images from noisy radon data (invited paper)[4119-10]. *Proc. SPIE*, (1):108–117, 2000.

[42] Emmanuel Candes, Laurent Demanet, David Donoho, and Lexing Ying. Fast discrete curvelet transforms. *Multiscale Modeling & Simulation*, 5(3):861–899, 2006.

[43] V. Cevher, S. Becker, and M. Schmidt. Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics. *Signal Processing Magazine, IEEE*, 31(5):32–43, Sept 2014.

[44] P. Chainais. Towards dictionary learning from images with non gaussian noise. In *Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on*, pages 1–6, Sept 2012.

[45] A. Chambolle, M. Novaga, D. Cremers, and T. Pock. An introduction to total variation for image analysis. In *in Theoretical Foundations and Numerical Methods for Sparse Recovery, De Gruyter*, 2010.

[46] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1-2):89–97, 2004.

[47] Antonin Chambolle and Pierre-Louis Lions. Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76(2):167–188, 1997.

[48] T. F. Chan, S. Esedoglu, F. E. Park, and A. M. Yip. Recent developments in total variation image restoration. In *Handbook of Mathematical Models in Computer Vision*. Springer, Berlin, 2005.

[49] Rick Chartrand, Emil Y Sidky, and Xiaochuan Pan. Nonconvex compressive sensing for x-ray ct: an algorithm comparison. In *Signals, Systems and Computers, 2013 Asilomar Conference on*, pages 665–669. IEEE, 2013.

[50] P. Chatterjee and P. Milanfar. Clustering-based denoising with locally learned dictionaries. *Image Processing, IEEE Transactions on*, 18(7):1438–1451, July 2009.

[51] P. Chatterjee and P. Milanfar. Patch-based near-optimal image denoising. *Image Processing, IEEE Transactions on*, 21(4):1635–1649, April 2012.

[52] Biao Chen and Ruola Ning. Cone-beam volume CT breast imaging: Feasibility study. *Medical Physics*, 29(5):755–770, 2002.

[53] Qiang Chen, Philippe Montesinos, Quan Sen Sun, Peng Ann Heng, and De Shen Xia. Adaptive total variation denoising based on difference curvature. *Image and Vision Computing*, 28(3):298 – 306, 2010.

[54] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

[55] Wei Chen and M.R.D. Rodrigues. Dictionary learning with optimized projection design for compressive sensing applications. *Signal Processing Letters, IEEE*, 20(10):992–995, Oct 2013.

[56] Yang Chen, Wufan Chen, Xindao Yin, Xianghua Ye, Xudong Bao, Limin Luo, Qianjing Feng, Yinsheng li, and Xiaoe Yu. Improving low-dose abdominal {CT} images by weighted intensity averaging over large-scale neighborhoods. *European Journal of Radiology*, 80(2):e42 – e49, 2011.

[57] Yang Chen, Luyao Shi, Qianjing Feng, Jian Yang, Huazhong Shu, Limin Luo, J.-L. Coatrieux, and Wufan Chen. Artifact suppressed dictionary learning for low-dose CT image processing. *Medical Imaging, IEEE Transactions on*, 33(12):2271–2292, Dec 2014.

[58] Yang Chen, Luyao Shi, Yining Hu, Qing Cao, Fei Yu, Limin Luo, and C. Toumoulin. Confidence weighted dictionary learning algorithm for low-dose ct image processing. In *Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 2013 IEEE*, pages 1–4, Oct 2013.

[59] Yang Chen, Luyao Shi, Jiang Yang, Yining Hu, Limin Luo, Xindao Yin, and Jean-Louis Coatrieux. Radiation dose reduction with dictionary learning based processing for head ct. *Australasian Physical & Engineering Sciences in Medicine*, 37(3):483–493, 2014.

[60] Yang Chen, Xindao Yin, Luyao Shi, Huazhong Shu, Limin Luo, Jean-Louis Coatrieux, and Christine Toumoulin. Improving abdomen tumor low-dose CT images using a fast dictionary learning based processing. *Physics in Medicine and Biology*, 58(16):5803, 2013.

[61] Kihwan Choi, Jing Wang, Lei Zhu, Tae-Suk Suh, Stephen Boyd, and Lei Xing. Compressed sensing based cone-beam computed tomography reconstruction with a first-order method. *Medical Physics*, 37(9):5113–5125, 2010.

[62] A. Cichocki, D. Mandic, L. De Lathauwer, Guoxu Zhou, Qibin Zhao, C. Caiafa, and H.A. Phan. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *Signal Processing Magazine, IEEE*, 32(2):145–163, March 2015.

[63] Chris A Cocosco, Vasken Kollokian, Remi K-S Kwan, G Bruce Pike, and Alan C Evans. Brainweb: Online interface to a 3d mri simulated brain database. In *NeuroImage*, 1997.

[64] Ronald R Coifman, Yves Meyer, and Victor Wickerhauser. Wavelet analysis and signal processing. In *In Wavelets and their Applications*. Citeseer, 1992.

[65] Michael Collins, S. Dasgupta, and Robert E Schapire. A generalization of principal components analysis to the exponential family. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 617–624. MIT Press, 2002.

[66] Patrick L Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer New York, 2011.

[67] Patrick L Combettes and Valérie R Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.

[68] S.F. Cotter, R. Adler, R.D. Rao, and K. Kreutz-Delgado. Forward sequential algorithms for best basis selection. *Vision, Image and Signal Processing, IEE Proceedings* -, 146(5):235–244, Oct 1999.

[69] Florent Couzinie-Devy, Julien Mairal, Francis Bach, and Jean Ponce. Dictionary learning for deblurring and digital zoom. *Arxiv preprint arXiv:1110.0957*, 2011.

[70] Antonio Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *Image Processing, IEEE Transactions on*, 13(9):1200–1212, Sept 2004.

[71] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *Image Processing, IEEE Transactions on*, 16(8):2080–2095, 2007.

[72] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Bm3d image denoising with shape-adaptive principal component analysis. In *SPARS '09 - Signal Processing with Adaptive Sparse Structured Representations*, pages 6–pp, 2009.

[73] J. Darbon, A. Cunha, T.F. Chan, S. Osher, and G.J. Jensen. Fast nonlocal filtering applied to electron cryomicroscopy. In *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, pages 1331–1334, May 2008.

[74] Ingrid Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41(7):909–996, 1988.

[75] John G. Daugman. Two-dimensional spectral analysis of cortical receptive field profiles. *Vision Research*, 20(10):847 – 856, 1980.

[76] John G Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *JOSA A*, 2(7):1160–1169, 1985.

[77] A. Dauwe, B. Goossens, H. Q. Luong, and W. Philips. A fast non-local image denoising algorithm, 2008.

[78] Bruno De Man and Samit Basu. Distance-driven projection and backprojection in three dimensions. *Physics in Medicine and Biology*, 49(11):2463–75, 2004.

[79] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2014.

[80] A.H. Delaney and Y. Bresler. Globally convergent edge-preserving regularized reconstruction: an application to limited-angle tomography. *Image Processing, IEEE Transactions on*, 7(2):204–221, Feb 1998.

[81] C.-A. Deledalle, F. Tupin, and L. Denis. Poisson nl means: Unsupervised non local means for poisson noise. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 801–804, Sept 2010.

[82] Charles-Alban Deledalle, Loïc Denis, and Florence Tupin. How to compare noisy patches? patch similarity beyond gaussian noise. *International journal of computer vision*, 99(1):86–102, 2012.

[83] Charles-Alban Deledalle, Vincent Duval, and Joseph Salmon. Non-local methods with shape-adaptive patches (nlm-sap). *Journal of Mathematical Imaging and Vision*, 43(2):103–120, 2012.

[84] Charles-Alban Deledalle, Joseph Salmon, and Arnak Dalalyan. Image denoising with patch based pca: local versus global. In *BMVC*, volume 81, pages 425–455, 2011.

[85] Charles-Alban Deledalle, Florence Tupin, and Loïc Denis. Patch similarity under non Gaussian noise. In *International Conference on Image Processing*, pages 1845 – 1848, Brussels, Belgium, September 2011.

[86] M.N. Do and M. Vetterli. The contourlet transform: an efficient directional multiresolution image representation. *Image Processing, IEEE Transactions on*, 14(12):2091–2106, Dec 2005.

[87] Weisheng Dong, Xin Li, D. Zhang, and Guangming Shi. Sparsity-based image denoising via dictionary learning and structural clustering. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 457–464, June 2011.

[88] Weisheng Dong, D. Zhang, Guangming Shi, and Xiaolin Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *Image Processing, IEEE Transactions on*, 20(7):1838–1857, July 2011.

[89] Yiqiu Dong, Michael Hintermüller, and M Monserrat Rincon-Camacho. Automated regularization parameter selection in multi-scale total variation models for image restoration. *Journal of Mathematical Imaging and Vision*, 40(1):82–104, 2011.

[90] DAVID L. DONOHO and JAIN M. JOHNSTONE. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

[91] V. Dore and M. Cheriet. Robust nl-means filter with optimal pixel-wise smoothing parameter for statistical image denoising. *Signal Processing, IEEE Transactions on*, 57(5):1703–1716, May 2009.

[92] L. Y. Du, J. Umoh, H. N. Nikolov, S. I. Pollmann, T. Y. Lee, and D. W. Holdsworth. A quality assurance phantom for the performance evaluation of volumetric micro-CT systems. *Physics in Medicine and Biology*, 52(23):7087–7108, 2007.

[93] J.M. Duarte-Carvajalino and G. Sapiro. Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization. *Image Processing, IEEE Transactions on*, 18(7):1395–1408, July 2009.

[94] F.-X. Dupe and S. Anthoine. A greedy approach to sparse poisson denoising. In *Machine Learning for Signal Processing (MLSP), 2013 IEEE International Workshop on*, pages 1–6, Sept 2013.

[95] Vincent Duval, Jean-Franois Aujol, and Yann Gousseau. A bias-variance approach for the nonlocal means. *SIAM Journal on Imaging Sciences*, 4(2):760–788, 2011.

[96] Glenn Easley, Demetrio Labate, and Wang-Q Lim. Sparse directional image representations using the discrete shearlet transform. *Applied and Computational Harmonic Analysis*, 25(1):25–46, 2008.

[97] A.A. Efros and T.K. Leung. Texture synthesis by non-parametric sampling. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1033–1038 vol.2, 1999.

[98] M. Elad. Sparse and redundant representation modeling; what next? *Signal Processing Letters, IEEE*, 19(12):922–928, Dec 2012.

[99] Michael Elad. *Sparse and redundant representations: from theory to applications in signal and image processing.* Springer, 2010.

[100] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *Image Processing, IEEE Transactions on*, 15(12):3736–3745, 2006.

[101] IA Elbakri and J.A Fessler. Statistical image reconstruction for polyenergetic x-ray computed tomography. *Medical Imaging, IEEE Transactions on*, 21(2):89–99, Feb 2002.

[102] Yonina C Eldar. Generalized sure for exponential families: Applications to regularization. *Signal Processing, IEEE Transactions on*, 57(2):471–481, 2009.

[103] K. Engan, S.O. Aase, and J. Hakon Husoy. Method of optimal directions for frame design. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, volume 5, pages 2443–2446 vol.5, 1999.

[104] Hakan Erdogan and Jeffrey A Fessler. Ordered subsets algorithms for transmission tomography. *Physics in medicine and biology*, 44(11):2835–2851, 1999.

[105] Vincent Etter, Ivana Jovanovic, and Martin Vetterli. Use of learned dictionaries in tomographic reconstruction, 2011.

[106] Benjamin P Fahimian, Yu Mao, Peter Cloetens, and Jianwei Miao. Low-dose x-ray phase-contrast and absorption ct using equally sloped tomography. *Physics in Medicine and Biology*, 55(18):5383, 2010.

[107] L. A. Feldkamp, L. C. Davis, and J. W. Kress. Practical cone-beam algorithm. *J. Opt. Soc. Am. A*, 1(6):612–619, Jun 1984.

[108] N. L. Ford, M. M. Thornton, and D. W. Holdsworth. Fundamental image quality limits for microcomputed tomography in small animals. *Medical Physics*, 30(11):2869–2877, 2003.

[109] Michael P. Friedlander and Mark W. Schmidt. Hybrid deterministic-stochastic methods for data fitting. *CoRR*, abs/1104.2373, 2011.

[110] Jrgen Frikel. Sparse regularization in limited angle tomography. *Applied and Computational Harmonic Analysis*, 34(1):117 – 141, 2013.

[111] Piotr Fryzlewicz and Guy P Nason. A haar-fisz algorithm for poisson intensity estimation. *Journal of Computational and Graphical Statistics*, 13(3):621–638, 2004.

[112] Pascal Getreuer. Rudin–osher–fatemi total variation denoising using split bregman. *Image Processing On Line*, 2012.

[113] G. Gilboa, N. Sochen, and Y.Y. Zeevi. Variational denoising of partly textured images by spatially varying constraints. *Image Processing, IEEE Transactions on*, 15(8):2281–2289, Aug 2006.

[114] Guy Gilboa, Jerome Darbon, Stanley Osher, and Tony Chan. Nonlocal convex functionals for image regularization. *UCLA CAM-report*, pages 06–57, 2006.

[115] Raja Giryes and Michael Elad. Sparsity based poisson denoising with dictionary learning. *arXiv preprint arXiv:1309.4306*, 2013.

[116] Donald Goldfarb and Wotao Yin. Second-order cone programming methods for total variation-based image restoration. *SIAM Journal on Scientific Computing*, 27(2):622–645, 2005.

[117] T. Goldstein and S. Osher. The split Bregman method for l1-regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.

[118] G.H. Golub and C. F. Van-Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, 2013.

[119] I.F. Gorodnitsky and B.D. Rao. Sparse signal reconstruction from limited data using focuss: a re-weighted minimum norm algorithm. *Signal Processing, IEEE Transactions on*, 45(3):600–616, Mar 1997.

[120] Markus Grasmair. Locally adaptive total variation regularization. In *Scale Space and Variational methods in computer Vision*, pages 331–342. Springer Berlin Heidelberg, 2009.

[121] Markus Grasmair and Frank Lenzen. Anisotropic total variation filtering. *Applied Mathematics & Optimization*, 62(3):323–339, 2010.

[122] Karol Gregor and Yann Lecun. Learning fast approximations of sparse coding. In *Machine Learning (ICML), 2010, International Conference on*, pages 1–8. Omnipress, 2010.

[123] Jie Tang Guang-Hong Chen and Shuai Leng. Prior image constrained compressed sensing (piccs): A method to accurately reconstruct dynamic ct images from highly undersampled projection data sets. *Medical Physics*, 35(2):660–663, 2008.

[124] C. Guillemot and O. Le Meur. Image inpainting : Overview and recent advances. *Signal Processing Magazine, IEEE*, 31(1):127–144, Jan 2014.

[125] Weihong Guo and Feng Huang. Adaptive total variation based filtering for mri images with spatially inhomogeneous noise and artifacts. In *Biomedical Imaging: From Nano to Macro, 2009. ISBI '09. IEEE International Symposium on*, pages 101–104, June 2009.

[126] Xiao Han, Junguo Bian, D.R. Eaker, T.L. Kline, E.Y. Sidky, E.L. Ritman, and Xiaochuan Pan. Algorithm-enabled low-dose micro-ct imaging. *Medical Imaging, IEEE Transactions on*, 30(3):606–620, March 2011.

[127] Simon Hawe, Matthias Seibert, and Martin Kleinsteuber. Separable dictionary learning. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 438–445. IEEE, 2013.

[128] Elad Hazan and Satyen Kale. Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization. *The Journal of Machine Learning Research*, 15(1):2489–2512, 2014.

[129] Gabor T Herman. *Fundamentals of computerized tomography: image reconstruction from projections*. Springer, 2009.

[130] G.T. Herman and L.B. Meyer. Algebraic reconstruction techniques can be made computationally efficient [positron emission tomography application]. *Medical Imaging, IEEE Transactions on*, 12(3):600–609, Sep 1993.

[131] H Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.

[132] De-An Huang and Yu-Chiang Frank Wang. Coupled dictionary and feature space learning with applications to cross-domain image synthesis and recognition. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2496–2503. IEEE, 2013.

[133] Jing Huang, Jianhua Ma, Nan Liu, Hua Zhang, Zhaoying Bian, Yanqiu Feng, Qianjin Feng, and Wufan Chen. Sparse angular {CT} reconstruction using non-local means based iterative-correction {POCS}. *Computers in Biology and Medicine*, 41(4):195 – 205, 2011.

[134] Junzhou Huang, Tong Zhang, and Dimitris Metaxas. Learning with structured sparsity. *The Journal of Machine Learning Research*, 12:3371–3412, 2011.

[135] J.M. Hughes, D.N. Rockmore, and Yang Wang. Bayesian learning of sparse multiscale image representations. *Image Processing, IEEE Transactions on*, 22(12):4972–4983, Dec 2013.

[136] Aapo Hyvärinen, Jarmo Hurri, and Patrick O Hoyer. *Natural Image Statistics: A probabilistic approach to early computational vision*, volume 39. Springer-Verlag New York Inc, 2009.

[137] David A Jaffray, Jeffrey H Siewerdsen, John W Wong, and Alvaro A Martinez. Flat-panel cone-beam computed tomography for image-guided radiation therapy. *International Journal of Radiation Oncology*Biology*Physics*, 53(5):1337 – 1349, 2002.

[138] John W. Tukey James W. Cooley. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965.

[139] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2146–2153, Sept 2009.

[140] Rodolphe Jenatton, Jean-Yves Audibert, and Francis Bach. Structured variable selection with sparsity-inducing norms. *J. Mach. Learn. Res.*, 12:2777–2824, November 2011.

[141] Rodolphe Jenatton, Julien Mairal, Francis R Bach, and Guillaume R Obozinski. Proximal methods for sparse hierarchical dictionary learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 487–494, 2010.

[142] Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, and Francis Bach. Proximal methods for hierarchical sparse coding. *Journal of Machine Learning Research*, (12):2297–2334, 2011.

[143] T. L. Jensen, J. H. Jørgensen, P. C. Hansen, and S. H. Jensen. Implementation of an optimal first-order method for strongly convex total variation regularization. *BIT Numerical Mathematics*, 52(2):329–356, 2011.

[144] Xun Jia, Bin Dong, Yifei Lou, and Steve B Jiang. GPU-based iterative cone-beam CT reconstruction using tight frame regularization. *Physics in Medicine and Biology*, 56(13):3787, 2011.

[145] Xun Jia, Yifei Lou, Ruijiang Li, William Y. Song, and Steve B. Jiang. GPU-based fast cone beam CT reconstruction from undersampled and noisy projection data via total variation. *Medical Physics*, 37(4):1757–1760, 2010.

[146] Xun Jia, Zhen Tian, Yifei Lou, Jan-Jakob Sonke, and Steve B. Jiang. Four-dimensional cone beam ct reconstruction and enhancement using a temporal nonlocal means method. *Medical Physics*, 39(9):5592–5602, 2012.

[147] Zhuolin Jiang, Zhe Lin, and L.S. Davis. Label consistent k-svd: Learning a discriminative dictionary for recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(11):2651–2664, Nov 2013.

[148] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 315–323. Curran Associates, Inc., 2013.

[149] Willi A. Kalender. *Computed Tomography: Fundamentals, System Technology, Image Quality, Applications*. Publicis, 2011.

[150] C. Kamphuis and F. J. Beekman. Accelerated iterative transmission ct reconstruction using an ordered subsets convex algorithm. *IEEE Transactions on Medical Imaging*, 17(6):1101–1105, Dec 1998.

[151] Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu. Automatic single-image-based rain streaks removal via image decomposition. *Image Processing, IEEE Transactions on*, 21(4):1742–1755, April 2012.

[152] Davood Karimi and Rabab Ward. On the computational implementation of forward and back-projection operations for cone-beam computed tomography. *Medical & Biological Engineering & Computing*, pages 1–12, 2015.

[153] K. Kavukcuoglu, M.A. Ranzato, R. Fergus, and Yann Le-Cun. Learning invariant features through topographic filter maps. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1605–1612, June 2009.

[154] D Kazantsev, G Van Eyndhoven, WRB Lionheart, PJ Withers, KJ Dobson, SA McDonald, R Atwood, and PD Lee. Employing temporal self-similarity across the entire time domain in computed tomography reconstruction. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 373(2043):20140389, 2015.

[155] Daniil Kazantsev, William M Thompson, G Van Eyndhoven, KJ Dobson, Anders P Kaestner, WRB Lionheart, Philip J Withers, and Peter D Lee. 4d-ct reconstruction with unified spatial-temporal patch-based regularization. *Inverse Probl. Imaging*, 9:447–467, 2015.

[156] Z.S. Kelm, D. Blezek, B. Bartholmai, and B.J. Erickson. Optimizing non-local means for denoising low dose ct. In *Biomedical Imaging: From Nano to Macro, 2009. ISBI '09. IEEE International Symposium on*, pages 662–665, June 2009.

[157] C. Kervrann and J. Boulanger. Optimal spatial adaptation for patch-based image denoising. *Image Processing, IEEE Transactions on*, 15(10):2866–2878, Oct 2006.

[158] Charles Kervrann and Jrme Boulanger. Local adaptivity to variable smoothness for exemplar-based image regularization and representation. *International Journal of Computer Vision*, 79(1):45–69, 2008.

[159] Donghwan Kim and J.A. Fessler. Ordered subsets acceleration using relaxed momentum for x-ray ct image reconstruction. In *Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 2013 IEEE*, pages 1–5, Oct 2013.

[160] Donghwan Kim, S. Ramani, and J.A. Fessler. Combining ordered subsets and momentum for accelerated x-ray CT image reconstruction. *Medical Imaging, IEEE Transactions on*, 34(1):167–178, Jan 2015.

[161] Donghwan Kim, Sathish Ramani, and Jeffrey A Fessler. Accelerating x-ray ct ordered subsets image reconstruction with nesterovs first-order methods. *Proc. Intl. Mtg. on Fully 3D Image Recon. in Rad. and Nuc. Med*, pages 22–5, 2013.

[162] Hojin Kim, Ruijiang Li, Rena Lee, Thomas Goldstein, Stephen Boyd, Emmanuel Candes, and Lei Xing. Dose optimization with first-order total-variation minimization for dense angularly sampled and sparse intensity modulated radiation therapy (dassim-rt). *Medical Physics*, 39(7), 2012.

[163] Stefan Kindermann, Stanley Osher, and Peter W. Jones. Deblurring and denoising of images by nonlocal functionals. *Multiscale Modeling & Simulation*, 4(4):1091–1115, 2005.

[164] Nick Kingsbury. Complex wavelets for shift invariant analysis and filtering of signals. *Applied and computational harmonic analysis*, 10(3):234–253, 2001.

[165] Jakub Konecný, Jie Liu, Peter Richtárik, and Martin Takác. mS2GD: Mini-batch semi-stochastic gradient descent in the proximal setting. *CoRR*, abs/1410.4744, 2014.

[166] Jakub Konecný, Zheng Qu, and Peter Richtárik. Semi-stochastic coordinate descent. *CoRR*, abs/1412.6293, 2014.

[167] Kenneth Kreutz-Delgado, Joseph F Murray, Bhaskar D Rao, Kjersti Engan, Te-Won Lee, and Terrence J Sejnowski. Dictionary learning algorithms for sparse representation. *Neural computation*, 15(2):349–396, 2003.

[168] D. Krishnan, Ping Lin, and A.M. Yip. A primal-dual active-set method for non-negativity constrained total variation deblurring problems. *Image Processing, IEEE Transactions on*, 16(11):2766–2777, Nov 2007.

[169] S. Krstulovic and R. Gribonval. Mptk: Matching pursuit made tractable. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 3, pages III–III, May 2006.

[170] N. Kumar, L. Zhang, and S. K. Nayar. What is a Good Nearest Neighbors Algorithm for Finding Similar Patches in Images? In *European Conference on Computer Vision (ECCV)*, pages 364–378, Oct 2008.

[171] Demetrio Labate, Wang-Q Lim, Gitta Kutyniok, and Guido Weiss. Sparse multidimensional representation using shearlets. In *Optics & Photonics 2005*, pages 59140U–59140U. International Society for Optics and Photonics, 2005.

[172] Simon Lacoste-Julien, Mark Schmidt, and Francis Bach. A simpler approach to obtaining an o(1/t) convergence rate for the projected stochastic subgradient method. *arXiv preprint arXiv:1212.2002*, 2012.

[173] Guanghui Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1-2):365–397, 2012.

[174] Patrick J LaRivire. Penalized-likelihood sinogram smoothing for low-dose CT. *Medical Physics*, 32(6):1676–1683, 2005.

[175] Triet Le, Rick Chartrand, and ThomasJ. Asaki. A variational approach to reconstructing images corrupted by Poisson noise. *Journal of Mathematical Imaging and Vision*, 27(3):257–263, 2007.

[176] E. Le Pennec and S. Mallat. Sparse geometric image representations with bandelets. *Image Processing, IEEE Transactions on*, 14(4):423–438, April 2005.

[177] Nicolas Le Roux, Mark Schmidt, and Francis Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. *arXiv preprint arXiv:1202.6258*, 2012.

[178] M. Lebrun, A. Buades, and J. M. Morel. A nonlocal bayesian image denoising algorithm. *SIAM Journal on Imaging Sciences*, 6(3):1665–1688, 2013.

[179] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *Advances in neural information processing systems*, pages 801–808, 2006.

[180] Shuai Leng, Jie Tang, Joseph Zambelli, Brian Nett, Ranjini Tolakanahalli, and Guang-Hong Chen. High temporal resolution and streak-free four-dimensional cone-beam computed tomography. *Physics in Medicine and Biology*, 53(20):5653, 2008.

[181] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2014.

[182] A. Levin and B. Nadler. Natural image denoising: Optimality and inherent bounds. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2833–2840, June 2011.

[183] Michael S Lewicki and Bruno A Olshausen. Probabilistic framework for the adaptation and comparison of image codes. *JOSA A*, 16(7):1587–1601, 1999.

[184] Michael S Lewicki and Terrence J Sejnowski. Learning overcomplete representations. *Neural computation*, 12(2):337–365, 2000.

[185] R M Lewitt. Alternatives to voxels for image representation in iterative reconstruction algorithms. *Physics in Medicine and Biology*, 37(3):705, 1992.

[186] Shutao Li, Leyuan Fang, and Haitao Yin. An efficient dictionary learning algorithm and its application to 3-d medical image denoising. *Biomedical Engineering, IEEE Transactions on*, 59(2):417–427, 2012.

[187] Shutao Li, Haitao Yin, and Leyuan Fang. Group-sparse representation with dictionary learning for medical image denoising and fusion. *Biomedical Engineering, IEEE Transactions on*, 59(12):3450–3459, Dec 2012.

[188] Si Li, Qing Cao, Yang Chen, Yining Hu, Limin Luo, and Christine Toumoulin. Dictionary learning based sinogram inpainting for CT sparse reconstruction. *Optik - International Journal for Light and Electron Optics*, 125(12):2862 – 2867, 2014.

[189] Tianfang Li, Xiang Li, Jing Wang, Junhai Wen, Hongbing Lu, Jiang Hsieh, and Zhengrong Liang. Nonlinear sinogram smoothing for low-dose x-ray CT. *Nuclear Science, IEEE Transactions on*, 51(5):2505–2513, Oct 2004.

[190] Yinsheng Li, Yang Chen, Yining Hu, Ahmed Oukili, Limin Luo, Wufan Chen, and Christine Toumoulin. Strategy of computed tomography sinogram inpainting based on sinusoid-like curve decomposition and eigenvector-guided interpolation. *J Opt Soc Am A Opt Image Sci Vis.*, 29(1):153–163, 2012.

[191] Zhoubo Li, Lifeng Yu, Joshua D. Trzasko, David S. Lake, Daniel J. Blezek, Joel G. Fletcher, Cynthia H. McCollough, and Armando Manduca. Adaptive nonlocal means filtering based on local noise level for ct denoising. *Medical Physics*, 41(1):–, 2014.

[192] Baodong Liu, Hengyong Yu, Scott S. Verbridge, Lizhi Sun, and Ge Wang. Dictionary-learning-based reconstruction method for electron tomography. *Scanning*, 36(4):377–383, 2014.

[193] Yong Long, J.A. Fessler, and J.M. Balter. 3D forward and back-projection for x-ray CT using separable footprints. *Medical Imaging, IEEE Transactions on*, 29(11):1839–1850, 2010.

[194] Yifei Lou, Xiaoqun Zhang, Stanley Osher, and Andrea Bertozzi. Image recovery via nonlocal operators. *Journal of Scientific Computing*, 42(2):185–197, 2010.

[195] C. Louchet and L. Moisan. Total variation as a local filter. *SIAM Journal on Imaging Sciences*, 4(2):651–694, 2011.

[196] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[197] Yang Lu, Jun Zhao, and Ge Wang. Few-view image reconstruction with dual dictionaries. *Physics in Medicine and Biology*, 57(1):173, 2012.

[198] Y.M. Lu and M.N. Do. Multidimensional directional filter banks and surfacelets. *Image Processing, IEEE Transactions on*, 16(4):918–931, April 2007.

[199] Marius Lysaker and XueCheng Tai. Iterative image restoration combining total variation minimization and a second-order functional. *International Journal of Computer Vision*, 66(1):5–18, 2006.

[200] Jianhua Ma, Jing Huang, Qianjin Feng, Hua Zhang, Hongbing Lu, Zhengrong Liang, and Wufan Chen. Low-dose computed tomography image restoration using previous normal-dose scan. *Medical Physics*, 38(10):5713–5731, 2011.

[201] Jianhua Ma, Zhengrong Liang, Yi Fan, Yan Liu, Jing Huang, Wufan Chen, and Hongbing Lu. Variance analysis of x-ray ct sinograms in the presence of electronic noise background. *Medical Physics*, 39(7), 2012.

[202] Jianhua Ma, Hua Zhang, Yang Gao, Jing Huang, Zhengrong Liang, Qianjing Feng, and Wufan Chen. Iterative image reconstruction for cerebral perfusion ct using a pre-contrast scan induced edge-preserving prior. *Physics in Medicine and Biology*, 57(22):7519, 2012.

[203] A. Plenevaux P. Choquet A. Constantinesco E. Salmon A. Luxen A. Seret M.A. Bahri, G. Warnock. Performance evaluation of the general electric explore ct 120 micro-ct using the vmct phantom. *Nuclear Instruments and Methods in Physics Research A*, 648:S181–S185, 2011.

[204] Albert Macovski. *Medical Imaging Systems*. Prentice Hall, Upper Saddle River, NJ, 1983.

[205] Matteo Maggioni, Vladimir Katkovnik, Karen Egiazarian, and Alessandro Foi. A nonlocal transform-domain filter for volumetric data denoising and reconstruction. *IEEE Transactions on Image Processing*, 22(1):1057–7149, 2013.

[206] Mehrdad Mahdavi and Rong Jin. MixedGrad: an O(1/T) convergence rate algorithm for stochastic smooth optimization. *arXiv preprint arXiv:1307.7192*, 2013.

[207] M. Mahmoudi and G. Sapiro. Fast image and video denoising via nonlocal means of similar neighborhoods. *Signal Processing Letters, IEEE*, 12(12):839–842, Dec 2005.

[208] Andreas Maier, Hannes G. Hofmann, Martin Berger, Peter Fischer, Chris Schwemmer, Haibo Wu, Kerstin Mller, Joachim Hornegger, Jang-Hwan Choi, Christian Riess, Andreas Keil, and Rebecca Fahrig. Conrada software framework for cone-beam imaging in radiology. *Medical Physics*, 40(11):–, 2013.

[209] Andreas Maier, Lars Wigstrm, Hannes G. Hofmann, Joachim Hornegger, Lei Zhu, Norbert Strobel, and Rebecca Fahrig. Three-dimensional anisotropic adaptive filtering of projection data for noise reduction in cone beam ct. *Medical Physics*, 38(11), 2011.

[210] Julien Mairal, Francis Bach, and Jean Ponce. Task-driven dictionary learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(4):791–804, 2012.

[211] Julien Mairal, Francis Bach, and Jean Ponce. Sparse modeling for image and vision processing. *Foundations and Trends® in Computer Graphics and Vision*, 8(2-3):85–283, 2014.

[212] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *International Conference on Machine Learning (ICML)*, 2009.

[213] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research (JMLR)*, 11:19–60, 2010.

[214] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Non-local sparse models for image restoration. In *International Conference on Computer Vision (ICCV)*. I, 2009.

[215] Julien Mairal, Michael Elad, and Guillermo Sapiro. Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 17(1):53–69, 2008.

[216] Julien Mairal, Guillermo Sapiro, and Michael Elad. Learning multiscale sparse representations for image and video restoration. *Multiscale Modeling & Simulation*, 7(1):214–241, 2008.

[217] S.G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(7):674–693, Jul 1989.

[218] Stéphane Mallat. *A wavelet tour of signal processing*. Academic press, 1999.

[219] Armando Manduca, Lifeng Yu, Joshua D. Trzasko, Natalia Khaylova, James M Kofler, Cynthia M McCollough, and Joel G. Fletcher. Projection space denoising with bilateral filtering and CT noise modeling for dose reduction in CT. *Medical Physics*, 36(11):4911–4919, 2009.

[220] S H Manglos, G M Gagne, A Krol, F D Thomas, and R Narayanaswamy. Transmission maximum-likelihood reconstruction with ordered subsets for cone beam CT. *Physics in Medicine and Biology*, 40(7):1225, 1995.

[221] S. Matej and R.M. Lewitt. Practical considerations for 3-D image reconstruction using spherically symmetric volume elements. *Medical Imaging, IEEE Transactions on*, 15(1):68–78, Feb 1996.

[222] Cynthia H. McCollough, Michael R. Bruesewitz, and James M. Kofler. CT dose reduction and dose management tools: Overview of available options. *RadioGraphics*, 26(2):503–512, 2006.

[223] Cynthia H. McCollough, Andrew N. Primak, Natalie Braun, James Kofler, Lifeng Yu, and Jodie Christner. Strategies for reducing radiation dose in CT. *Radiologic Clinics of North America*, 47(1):27 – 40, 2009.

[224] M.G. McGaffin and J.A. Fessler. Edge-preserving image denoising via group coordinate descent on the gpu. *Image Processing, IEEE Transactions on*, 24(4):1273–1281, April 2015.

[225] Daniele Menon and Giancarlo Calvagno. Color image demosaicking: An overview. *Signal Processing: Image Communication*, 26(89):518 – 533, 2011.

[226] Bhargavan M Gilley DB Gray JE Lipoti JA McCrohan J Yoshizumi TT Mahesh M. Mettler FA Jr, Thomadsen BR. Medical radiation exposure in the u.s. in 2006: preliminary results. *Health Physics*, 95(5):502–507, 2008.

[227] Jianwei Miao, Friedrich Förster, and Ofer Levi. Equally sloped tomography with oversampling reconstruction. *Phys. Rev. B*, 72:052103, Aug 2005.

[228] P. Milanfar. A tour of modern image filtering: New insights and methods, both practical and theoretical. *Signal Processing Magazine, IEEE*, 30(1):106–128, Jan 2013.

[229] Y.Q. Mohsin, G. Ongie, and M. Jacob. Iterative shrinkage algorithm for patch-smoothness regularized medical image recovery. *Medical Imaging, IEEE Transactions on*, 34(12):2417–2428, Dec 2015.

[230] G.P. Nason and B.W. Silverman. The stationary wavelet transform and some statistical applications. In Anestis Antoniadis and Georges Oppenheim, editors, *Wavelets and Statistics*, volume 103 of *Lecture Notes in Statistics*, pages 281–299. Springer New York, 1995.

[231] F. Natterer. *The Mathematics of Computerized Tomography*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2001.

[232] Deanna Needell, Rachel Ward, and Nati Srebro. Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. In *Advances in Neural Information Processing Systems*, pages 1017–1025, 2014.

[233] AS Nemirovskii and DB Yudin. *Problem complexity and method efficiency in optimization*. Wiley (Chichester and New York), 1983.

[234] Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

[235] Yurii Nesterov. A method of solving a convex programming problem with convergence rate O (1/k2). *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[236] Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer, Boston, MA, 2003.

[237] Yurii Nesterov. Gradient methods for minimizing composite objective function. Technical Report CCIT 559, Universite Catholique de Louvain, 2007.

[238] Yurii Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1):221–259, 2009.

[239] Brian E Nett, Robert Brauweiler, Willi Kalender, Howard Rowley, and Guang-Hong Chen. Perfusion measurements by micro-ct using prior image constrained compressed sensing (piccs): initial phantom results. *Physics in Medicine and Biology*, 55(8):2333, 2010.

[240] Hung Nien, Jeffrey Fessler, et al. Fast x-ray ct image reconstruction using a linearized augmented lagrangian method with ordered subsets. *Medical Imaging, IEEE Transactions on*, 34(2):388–399, 2015.

[241] Atsushi Nitanda. Stochastic proximal gradient descent with acceleration techniques. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1574–1582. Curran Associates, Inc., 2014.

[242] Tianye Niu and Lei Zhu. Accelerated barrier optimization compressed sensing (abocs) reconstruction for cone-beam ct: Phantom studies. *Medical Physics*, 39(7), 2012.

[243] Jorge Nocedal and Stephen J Wright. Numerical optimization, second edition. *Numerical optimization*, pages 497–528, 2006.

[244] Johan Nuyts, Bruno De Man, Jeffrey A Fessler, Wojciech Zbijewski, and Freek J Beekman. Modelling the physics in the iterative reconstruction for transmission computed tomography. *Physics in medicine and biology*, 58(12):R63, 2013.

[245] Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[246] B. Ophir, M. Lustig, and M. Elad. Multi-scale dictionary learning using wavelets. *Selected Topics in Signal Processing, IEEE Journal of*, 5(5):1014–1024, Sept 2011.

[247] X. Pan, E. Y. Sidky, and M. Vannier. Why do commercial CT scanners still employ traditional, filtered back-projection for image reconstruction? *Inverse Problems*, 25(12):123009, 2009.

[248] K. Papafitsoros and C.B. Schnlieb. A combined first and second order variational approach for image reconstruction. *Journal of Mathematical Imaging and Vision*, 48(2):308–338, 2014.

[249] Konstantinos Papafitsoros, Carola Bibiane Schoenlieb, and Bati Sengul. Combined first and second order total variation inpainting using split bregman. *Image Processing On Line*, 2013:112–136, 2013.

[250] V. Papyan and M. Elad. Multi-scale patch-based image restoration. *Image Processing, IEEE Transactions on*, 25(1):249–261, Jan 2016.

[251] Justin C. Park, Bongyong Song, Jin Sung Kim, Sung Ho Park, Ho Kyung Kim, Zhaowei Liu, Tae Suk Suh, and William Y. Song. Fast compressed sensing-based CBCT reconstruction using Barzilai-Borwein formulation for application to on-line IGRT. *Medical Physics*, 39(3):1207–1217, 2012.

[252] Yagyensh Chandra Pati, Ramin Rezaiifar, and PS Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, pages 40–44. IEEE, 1993.

[253] Tomer Peleg and Michael Elad. A statistical prediction model based on sparse representations for single image super-resolution. *Image Processing, IEEE Transactions on*, 23(6):2569–2582, 2014.

[254] Gabriel Peyr. Sparse modeling of textures. *Journal of Mathematical Imaging and Vision*, 34(1):17–31, 2009.

[255] Gabriel Peyr, Sbastien Bougleux, and Laurent Cohen. Non-local regularization of inverse problems. In David Forsyth, Philip Torr, and Andrew Zisserman, editors, *Computer Vision ECCV 2008*, volume 5304 of *Lecture Notes in Computer Science*, pages 57–68. Springer Berlin Heidelberg, 2008.

[256] Luke Pfister. Tomographic reconstruction with adaptive sparsifying transforms. Master's thesis, University of Illinois at Urbana-Champaign, 2013.

[257] Luke Pfister and Yoram Bresler. Model-based iterative tomographic reconstruction with adaptive sparsifying transforms, 2014.

[258] Nicolas Pinto, David D Cox, and James J DiCarlo. Why is real-world visual object recognition hard? *PLoS Computational Biology*, 4(1):e27, jan 2008. PMID: 18225950.

[259] J.P.W. Pluim, J.B.A. Maintz, and M.A. Viergever. Mutual-information-based registration of medical images: a survey. *Medical Imaging, IEEE Transactions on*, 22(8):986–1004, 2003.

[260] B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

[261] G. Pratx, G. Chinn, P.D. Olcott, and C.S. Levin. Fast, accurate and shift-varying line projections for iterative reconstruction using the gpu. *Medical Imaging, IEEE Transactions on*, 28(3):435–445, 2009.

[262] M. Protter, M. Elad, H. Takeda, and P. Milanfar. Generalizing the nonlocal-means to super-resolution reconstruction. *Image Processing, IEEE Transactions on*, 18(1):36–51, Jan 2009.

[263] S. Ramani and J.A. Fessler. A splitting-based iterative algorithm for accelerated statistical x-ray ct reconstruction. *Medical Imaging, IEEE Transactions on*, 31(3):677–688, March 2012.

[264] Sathish Ramani and Jeffrey A Fessler. Convergent iterative ct reconstruction with sparsity-based regularization. *Proc. Intl. Mtg. on Fully 3D Image Recon. in Rad. and Nuc. Med*, pages 302–5, 2011.

[265] I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3501–3508, June 2010.

[266] M. Rantala, S. Vanska, S. Jarvenpaa, Martti Kalke, M. Lassas, J. Moberg, and S. Siltanen. Wavelet-based reconstruction for limited-angle x-ray tomography. *Medical Imaging, IEEE Transactions on*, 25(2):210–217, Feb 2006.

[267] H. Rauhut, Karin Schnass, and P. Vandergheynst. Compressed sensing and redundant dictionaries. *Information Theory, IEEE Transactions on*, 54(5):2210–2219, May 2008.

[268] Saiprasad Ravishankar and Yoram Bresler. Learning doubly sparse transforms for images. *IEEE Trans. Image Process.*, 22(12):4598–4612, 2013.

[269] Harold L. Kundel Richard L. Van Metter, Jacob Beutel. *Handbook of Medical Imaging, Volume 1. Physics and Psychophysics.* SPIE Publications, Bellingham, WA, 2000.

[270] Ludwig Ritschl, Frank Bergner, Christof Fleischmann, and Marc Kachelrie. Improved total variation-based ct image reconstruction applied to clinical data. *Physics in Medicine and Biology*, 56(6):1545, 2011.

[271] Arie Rond, Raja Giryes, and Michael Elad. Poisson inverse problems by the plug-and-play scheme. *arXiv preprint arXiv:1511.02500*, 2015.

[272] Jeffrey M Rosen, Junjie Wu, TF Wenisch, and JA Fessler. Iterative helical ct reconstruction in the cloud for ten dollars in five minutes. *Proc. Intl. Mtg. on Fully 3D Image Recon. in Rad. and Nuc. Med*, pages 241–4, 2013.

[273] R. Rubinstein, AM. Bruckstein, and M. Elad. Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6):1045–1057, 2010.

[274] R. Rubinstein, M. Zibulevsky, and M. Elad. Double sparsity: Learning sparse dictionaries for sparse signal approximation. *Signal Processing, IEEE Transactions on*, 58(3):1553–1564, March 2010.

[275] Ron Rubinstein, Michael Zibulevsky, and Michael Elad. Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit. *CS Technion*, 40(8):1–15, 2008.

[276] Ron Rubinstein, Michael Zibulevsky, and Michael Elad. Learning sparse dictionaries for sparse signal approximation. *Computer Science Department, Techical Report CS*, 13, 2009.

[277] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(14):259 – 268, 1992.

[278] Curtis R.Vogel. *Computational Methods for Inverse Problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2002.

[279] Elham Sakhaee and Alireza Entezari. Learning splines for sparse tomographic reconstruction. In George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Ryan McMahan, Jason Jerald, Hui Zhang, StevenM. Drucker, Chandra Kambhamettu, Maha El Choubassi, Zhigang Deng, and Mark Carlson, editors, *Advances in Visual Computing*, volume 8887 of *Lecture Notes in Computer Science*, pages 1–10. Springer International Publishing, 2014.

[280] J. Salmon, C.-A. Deledalle, R. Willett, and Z. Harmany. Poisson noise reduction with non-local pca. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 1109–1112, March 2012.

[281] Joseph Salmon, Zachary Harmany, Charles-Alban Deledalle, and Rebecca Willett. Poisson noise reduction with non-local pca. *Journal of Mathematical Imaging and Vision*, 48(2):279–294, 2014.

[282] Alex Sawatzky, Christoph Brune, Jahn Mller, and Martin Burger. Total variation processing of images with Poisson statistics. In Xiaoyi Jiang and Nicolai Petkov, editors, *Computer Analysis of Images and Patterns*, volume 5702 of *Lecture Notes in Computer Science*, pages 533–540. Springer Berlin Heidelberg, 2009.

[283] William C. Scarfe and Allan G. Farman. What is cone-beam ct and how does it work? *The Dental Clinics of North America*, 52:707–730, 2008.

[284] Mark Schmidt, Nicolas L Roux, and Francis R Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. In *Advances in neural information processing systems*, pages 1458–1466, 2011.

[285] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *arXiv preprint arXiv:1309.2388*, 2013.

[286] SIMON Setzer and GABRIELE Steidl. Variational methods with higher order derivatives in image processing. *Approximation XII, Nashboro Press, Brentwood*, pages 360–386, 2008.

[287] Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss. *The Journal of Machine Learning Research*, 14(1):567–599, 2013.

[288] Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. *arXiv preprint arXiv:1212.1824*, 2012.

[289] Ling Shao, Ruomei Yan, Xuelong Li, and Yan Liu. From heuristic optimization to dictionary learning: A review and comprehensive comparison of image denoising algorithms. *Cybernetics, IEEE Transactions on*, 44(7):1001–1013, July 2014.

[290] JohnA. Shepherd. Body composition. In Colin G. Miller, Joel Krasnow, and Lawrence H. Schwartz, editors, *Medical Imaging in Clinical Trials*, pages 253–275. Springer London, 2014.

[291] J. Shtok, P. Elad, and M. Zibulevsky. Sparsity-based sinogram denoising for low-dose computed tomography. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 569–572, May 2011.

[292] Joseph Shtok, Michael Elad, and Michael Zibulevsky. Learned shrinkage approach for low-dose reconstruction in computed tomography. *Journal of Biomedical Imaging*, 2013:7, 2013.

[293] E. Y. Sidky and X. Pan. Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization. *Physics in Medicine and Biology*, 53(17):4777–4807, 2008.

[294] E.Y. Sidky, R. Chartrand, and Xiaochuan Pan. Image reconstruction from few views by non-convex optimization. In *Nuclear Science Symposium Conference Record, 2007. NSS '07. IEEE*, volume 5, pages 3526–3530, Oct 2007.

[295] E.P. Simoncelli, W.T. Freeman, E.H. Adelson, and D.J. Heeger. Shiftable multiscale transforms. *Information Theory, IEEE Transactions on*, 38(2):587–607, March 1992.

[296] Yoram Singer and John C Duchi. Efficient learning using forward-backward splitting. In Y. Bengio, D. Schuurmans, J.D. Lafferty, C.K.I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 495–503. Curran Associates, Inc., 2009.

[297] AjitP. Singh and GeoffreyJ. Gordon. A unified view of matrix factorization models. In Walter Daelemans, Bart Goethals, and Katharina Morik, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 5212 of *Lecture Notes in Computer Science*, pages 358–373. Springer Berlin Heidelberg, 2008.

[298] K. Skretting and K. Engan. Image compression using learned dictionaries by rls-dla and compared with k-svd. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 1517–1520, May 2011.

[299] R. Smith-Bindman, J. Lipson, R. Marcus, K. Kim, M. Mahesh, R. Gould, A. Gonzalez, and D.l Miglioretti. Radiation dose associated with common computed tomography examinations and the associated lifetime attributable risk of cancer. *Archives of Internal Medicine*, 169(22):2078–2086, 2009.

[300] Sara Soltani. *Studies of Sensitivity in the Dictionary Learning Approach to Computed Tomography: Simplifying the Reconstruction Problem, Rotation, and Scale*. DTU Compute-Technical Report-2015. Technical University of Denmark, 2015.

[301] Sara Soltani, Martin S. Andersen, and Per Christian Hansen. Tomographic image reconstruction using dictionary priors. *CoRR*, abs/1503.01993, 2015.

[302] Jiayu Song, Qing H. Liu, G. Allan Johnson, and Cristian T. Badea. Sparseness prior based iterative image reconstruction for retrospectively gated cardiac micro-CT. *Medical Physics*, 34(11):4476–4483, 2007.

[303] Emmanuel Candes Stephen Becker, Jerome Bobin. Nesta: A fast and accurate first-order method for sparse recovery. Technical report, California Institute of Technology, 2009.

[304] Andrew Stevens, Hao Yang, Lawrence Carin, Ilke Arslan, and Nigel D. Browning. The potential for bayesian compressive sensing to significantly reduce electron dose in high-resolution stem images. *Microscopy*, 63(1):41–51, 2014.

[305] I Stojanovic, H. Pien, Synho Do, and W.C. Karl. Low-dose X-ray CT reconstruction based on joint sinogram smoothing and learned dictionary-based representation. In *Biomedical Imaging (ISBI), 2012 9th IEEE International Symposium on*, pages 1012–1015, May 2012.

[306] C. Sutour, C.-A. Deledalle, and J.-F. Aujol. Adaptive regularization of the nl-means: Application to image and video denoising. *Image Processing, IEEE Transactions on*, 23(8):3506–3521, Aug 2014.

[307] Katsuyuki Taguchi, Be-Shan S Chiang, and Michael D Silver. A new weighting scheme for cone-beam helical CT to reduce the image noise. *Physics in Medicine and Biology*, 49(11):2351, 2004.

[308] Shengqi Tan, Yanbo Zhang, Ge Wang, Xuanqin Mou, Guohua Cao, Zhifang Wu, and Hengyong Yu. Tensor-based dictionary learning for dynamic tomographic reconstruction. *Physics in Medicine and Biology*, 60(7):2803, 2015.

[309] Jie Tang, Brian E Nett, and Guang-Hong Chen. Performance comparison between total variation (TV)-based compressed sensing and statistical iterative reconstruction algorithms. *Physics in Medicine and Biology*, 54(19):5781, 2009.

[310] Xiangyang Tang, Jiang Hsieh, Roy A Nilsen, Sandeep Dutta, Dmitry Samsonov, and Akira Hagiwara. A three-dimensional-weighted cone beam filtered backprojection (CB-FBP) algorithm for image reconstruction in volumetric CT - helical scanning. *Physics in Medicine and Biology*, 51(4):855, 2006.

[311] T. Tasdizen. Principal neighborhood dictionaries for nonlocal means image denoising. *Image Processing, IEEE Transactions on*, 18(12):2649–2660, Dec 2009.

[312] Jean-Baptiste Thibault, Charles A Bouman, Ken D Sauer, and Jiang Hsieh. A recursive filter for noise reduction in statistical iterative tomographic imaging. In *Electronic Imaging 2006*, pages 60650X–60650X. International Society for Optics and Photonics, 2006.

[313] Markus Thom, Matthias Rapp, and Gnther Palm. Efficient dictionary learning with sparseness-enforcing projections. *International Journal of Computer Vision*, 114(2-3):168–194, 2015.

[314] Zhen Tian, Xun Jia, Bin Dong, Yifei Lou, and Steve B. Jiang. Low-dose 4dct reconstruction via temporal nonlocal means). *Medical Physics*, 38(3), 2011.

[315] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[316] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846, Jan 1998.

[317] J.A Tropp and S.J. Wright. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 98(6):948–958, June 2010.

[318] Joel A Tropp. Algorithms for simultaneous sparse approximation. part ii: Convex relaxation. *Signal Processing*, 86(3):589–602, 2006.

[319] Joel A Tropp, Anna C Gilbert, and Martin J Strauss. Algorithms for simultaneous sparse approximation. part i: Greedy pursuit. *Signal Processing*, 86(3):572–588, 2006.

[320] Paul Tseng. An incremental gradient(-projection) method with momentum term and adaptive stepsize rule. *SIAM Journal on Optimization*, 8(2):506–531, 1998.

[321] Bert Vandeghinste, Bart Goossens, Roel Van Holen, Christian Vanhove, Aleksandra Pizurica, Stefaan Vandenberghe, and Steven Staelens. Iterative ct reconstruction using shearlet-based regularization. *IEEE Transactions on Nuclear Science*, 60(5):3305–3317, 2013.

[322] S. 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*, pages 945–948, Dec 2013.

[323] Jing Wang, Huaiqun Guan, and Timothy Solberg. Inverse determination of the penalty parameter in penalized weighted least-squares algorithm for noise reduction of low-dose cbct. *Medical Physics*, 38(7):4066–4072, 2011.

[324] Jing Wang, Tianfang Li, Hongbing Lu, and Zhengrong Liang. Penalized weighted least-squares approach to sinogram noise reduction and image reconstruction for low-dose X-ray computed tomography. *Medical Imaging, IEEE Transactions on*, 25(10):1272–1283, Oct 2006.

[325] Jing Wang, Tianfang Li, and Lei Xing. Iterative image reconstruction for cbct using edge-preserving prior. *Medical Physics*, 36(1), 2009.

[326] Jing Wang, Hongbing Lu, Zhengrong Liang, Daria Eremina, Guangxiang Zhang, Su Wang, John Chen, and James Manzione. An experimental study on the noise properties of x-ray CT sinogram data in Radon space. *Physics in Medicine and Biology*, 53(12):3327, 2008.

[327] Shenlong Wang, Lei Zhang, Yan Liang, and Quan Pan. Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2216–2223. IEEE, 2012.

[328] Yilun Wang, Wotao Yin, and Yin Zhang. A fast algorithm for image deblurring with total variation regularization, CAAM Technical Report TR07-10. Technical report, Rice University, Houston, 2007.

[329] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, 2004.

[330] Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 479–488. ACM Press/Addison-Wesley Publishing Co., 2000.

[331] Bihan Wen, Saiprasad Ravishankar, and Yoram Bresler. Structured overcomplete sparsifying transform learning with convergence guarantees and applications. *International Journal of Computer Vision*, 114(2-3):137–167, 2015.

[332] C. Wu and X. Tai. Augmented Lagrangian method, dual methods, and Split Bregman iteration for ROF, vectorial TV, and high order models. *SIAM Journal on Imaging Sciences*, 3(3):300–339, 2010.

[333] Yue Wu, B. Tracey, P. Natarajan, and J.P. Noonan. Probabilistic non-local means. *Signal Processing Letters, IEEE*, 20(8):763–766, Aug 2013.

[334] Zhen J Xiang, Hao Xu, and Peter J Ramadge. Learning sparse representations of high dimensional data on large scale dictionaries. In *Advances in Neural Information Processing Systems*, pages 900–908, 2011.

[335] Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11:2543–2596, October 2010.

[336] Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

[337] Zhengming Xing, Mingyuan Zhou, Alexey Castrodad, Guillermo Sapiro, and Lawrence Carin. Dictionary learning for noisy and incomplete hyperspectral images. *SIAM Journal on Imaging Sciences*, 5(1):33–56, 2012.

[338] Fang Xu, Wei Xu, Mel Jones, Bettina Keszthelyi, John Sedat, David Agard, and Klaus Mueller. On the efficiency of iterative ordered subset reconstruction algorithms for acceleration on GPUs. *Computer Methods and Programs in Biomedicine*, 98(3):261 – 270, 2010. HP-MICCAI 2008.

[339] Qiong Xu, Hengyong Yu, Xuanqin Mou, Lei Zhang, Jiang Hsieh, and Ge Wang. Low-dose x-ray CT reconstruction via dictionary learning. *Medical Imaging, IEEE Transactions on*, 31(9):1682–1697, 2012.

[340] Wei Xu, Sungsoo Ha, and Klaus Mueller. Database-assisted low-dose ct image restoration. *Medical Physics*, 40(3):–, 2013.

[341] Wei Xu and Klaus Mueller. Efficient low-dose ct artifact mitigation using an artifact-matched prior scan. *Medical Physics*, 39(8), 2012.

[342] John Lewis Ruijiang Li Xuejun Gu Chunhua Men William Y. Song Xun Jia, Yifei Lou and Steve B. Jiang. GPU-based fast low-dose cone beam ct reconstruction via total variation. *Journal of XRay Science and Technology*, 19:139–154, 2011.

[343] Hao Yan, Xin Zhen, Laura Cervio, Steve B. Jiang, and Xun Jia. Progressive cone beam ct dose control in image-guided radiation therapy. *Medical Physics*, 40(6):–, 2013.

[344] Jianchao Yang, Zhaowen Wang, Zhe Lin, Scott Cohen, and Tingwen Huang. Coupled dictionary training for image super-resolution. *Image Processing, IEEE Transactions on*, 21(8):3467–3478, 2012.

[345] Jianchao Yang, J. Wright, T.S. Huang, and Yi Ma. Image super-resolution via sparse representation. *Image Processing, IEEE Transactions on*, 19(11):2861–2873, Nov 2010.

[346] Jiansheng Yang, Hengyong Yu, Ming Jiang, and Ge Wang. High-order total variation minimization for interior tomography. *Inverse Problems*, 26(3):035013, 2010.

[347] Shuyuan Yang, Min Wang, Y. Chen, and Y. Sun. Single-image super-resolution reconstruction via learned geometric dictionaries and clustered sparse coding. *Image Processing, IEEE Transactions on*, 21(9):4016–4028, Sept 2012.

[348] Zhili Yang and M. Jacob. Nonlocal regularization of inverse problems: A unified variational framework. *Image Processing, IEEE Transactions on*, 22(8):3192–3203, Aug 2013.

[349] Leonid Yaroslavsky. *Digital Picture Processing, An Introduction*. Springer-Verlag, Berling, Germany, 1985.

[350] Chia-Hung Yeh, Li-Wei Kang, Yi-Wen Chiou, Chia-Wen Lin, and Shu-Jhen Fan Jiang. Self-learning-based post-processing for image/video deblocking via sparse representation. *Journal of Visual Communication and Image Representation*, 25(5):891 – 903, 2014.

[351] Stanislav Zabic, Qiu Wang, Thomas Morton, and Kevin M. Brown. A low dose simulation tool for ct systems with energy integrating detectors. *Medical Physics*, 40(3):–, 2013.

[352] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Proceedings of the 7th International Conference on Curves and Surfaces*, pages 711–730, Berlin, Heidelberg, 2012. Springer-Verlag.

[353] Hao Zhang, Jianhua Ma, Jing Wang, Yan Liu, Hongbing Lu, and Zhengrong Liang. Statistical image reconstruction for low-dose {CT} using nonlocal means-based regularization. *Computerized Medical Imaging and Graphics*, 38(6):423 – 435, 2014.

[354] Xiaoqun Zhang, Martin Burger, Xavier Bresson, and Stanley Osher. Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. *SIAM Journal on Imaging Sciences*, 3(3):253–276, 2010.

[355] Bo Zhao, Huanjun Ding, Yang Lu, Ge Wang, Jun Zhao, and Sabee Molloi. Dual-dictionary learning-based iterative image reconstruction for spectral computed tomography application. *Physics in Medicine and Biology*, 57(24):8217, 2012.

[356] Peilin Zhao and Tong Zhang. Accelerating minibatch stochastic gradient descent using stratified sampling. *arXiv preprint arXiv:1405.3080*, 2014.

[357] Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling. *arXiv preprint arXiv:1401.2753*, 2014.

[358] Mingyuan Zhou, Haojun Chen, John Paisley, Lu Ren, Lingbo Li, Zhengming Xing, David Dunson, Guillermo Sapiro, and Lawrence Carin. Nonparametric bayesian dictionary learning for analysis of noisy and incomplete images. *Image Processing, IEEE Transactions on*, 21(1):130–144, 2012.

[359] Mingyuan Zhou, Haojun Chen, John Paisley, Lu Ren, Guillermo Sapiro, and Lawrence Carin. Non-parametric bayesian dictionary learning for sparse image representations 1. In *NIPS 2009*, 2009.

[360] Hongqing Zhu, Huazhong Shu, Jian Zhou, and Limin Luo. A weighted least squares pet image reconstruction method using iterative coordinate descent algorithms. In *Nuclear Science Symposium Conference Record, 2004 IEEE*, volume 6, pages 3380–3384 Vol. 6, Oct 2004.

[361] Mingqiang Zhu, Stephen J. Wright, and Tony F. Chan. Duality-based algorithms for total-variation-regularized image restoration. *Computational Optimization and Applications*, 47(3):377–400, 2008.

[362] Yue Ting Zhuang, Yan Fei Wang, Fei Wu, Yin Zhang, and Wei Ming Lu. Supervised coupled dictionary learning with group structures for multi-modal retrieval. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[363] Nielsen T Ziegler A, Khler T and Proksa R. Efficient projection and backprojection scheme for spherically symmetric basis functions in divergent beam geometry. *Med Phys*, 33(12):4653–63, 2010.