Alternating Minimization Algorithms for Dual-Energy X-Ray CT Imaging and Information Optimization

Yaqi Chen

Washington University in St. Louis

Follow this and additional works at: https://openscholarship.wustl.edu/etd

Recommended Citation
https://openscholarship.wustl.edu/etd/1291

This Dissertation is brought to you for free and open access by Washington University Open Scholarship. It has been accepted for inclusion in All Theses and Dissertations (ETDs) by an authorized administrator of Washington University Open Scholarship. For more information, please contact digital@wumail.wustl.edu.
Alternating Minimization Algorithms for
Dual-Energy X-Ray CT Imaging and Information Optimization

by

Yaqi Chen

A dissertation presented to the Graduate School of Arts and Sciences of Washington University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

August 2014
Saint Louis, Missouri
Contents

List of Figures ......................................................... v
List of Tables ......................................................... ix
Acknowledgments ...................................................... x
Abstract ................................................................. xii

1 Introduction ......................................................... 1
  1.1 Proton Therapy ................................................ 1
    1.1.1 Proton Therapy Overview ................................ 1
    1.1.2 Dose Prediction in Proton Therapy ....................... 3
  1.2 Brachytherapy Overview ...................................... 3
  1.3 Transmission Tomography ..................................... 4
    1.3.1 X-ray CT Overview ....................................... 4
    1.3.2 Statistical Image Reconstruction for Transmission Tomography ........................................... 6
  1.4 Dual-Energy X-ray CT ........................................ 8
    1.4.1 Dual-Energy X-ray CT Data Acquisition .................. 8
    1.4.2 Dual-Energy X-ray CT Algorithms ....................... 9
  1.5 Dissertation Organization .................................... 10

2 Dual-Energy AM Algorithm Reconstruction .................... 12
  2.1 Dual-Energy AM (DE-AM) Algorithm ........................ 12
  2.2 Regularized DE-AM Algorithm ................................ 14
  2.3 Experiments and Reconstructions ............................ 18
    2.3.1 Simulated Data Reconstruction in 2D ..................... 18
    2.3.2 Real Data Reconstruction in 2D ........................ 24
    2.3.3 Real Data Reconstruction in 3D ........................ 27
  2.4 Discussion .................................................... 31
  2.5 Conclusions and Conclusions ................................ 39

3 Line Integral Alternating Minimization (LIAM) Algorithm ... 40
  3.1 Motivation .................................................... 40
  3.2 Line Integral Alternating Minimization (LIAM) Algorithm ... 41
    3.2.1 Data Model .............................................. 41
5.5 Conclusions and Contributions ........................................ 128

6 Computation of Berger-Tung Bounds for Lossy Distributed Source Coding 130
   6.1 Abstract ................................................................. 130
   6.2 Introduction ............................................................ 131
     6.2.1 Definition of Berger-Tung Regions .............................. 131
   6.3 Computational Problems .............................................. 133
     6.3.1 Optimal Joint Distributions .................................... 135
     6.3.2 Alternating Minimization Algorithms ......................... 135
   6.4 Motivational Examples .............................................. 136
     6.4.1 Doubly Symmetric Binary Source (DSBS(p)) ................. 136
     6.4.2 Concatenation of Two DSBS Sources ............................ 137
     6.4.3 Asymmetric Sources of Different Cardinalities ............... 139
     6.4.4 Wagner and Anantharam Example ............................... 140
   6.5 Contributions and Conclusions .................................... 141

7 Conclusions and Future Work ........................................... 143
   7.1 Contributions ......................................................... 143
   7.2 Conclusions and Future Work ..................................... 145

References ................................................................. 149

Appendix A Derivation of the Dual-Energy Alternating Minimization (DE-AM) Algorithm .............................. 158

Vita ................................................................. 164
List of Figures

1.1 (a) Dose versus depth for protons. Figure is adapted from [5]. (b) Relative dose versus depth for protons of the same energy from different materials. Central axis dose, 5×5 cm² beam, 250 MeV protons, phantoms of constant water-equivalent electron density, but variable atomic compositions. Figure 1.1(b) is used with permission of Jeffrey V. Siebers and Jeffrey F. Williamson (unpublished). ................................................................. 2
1.2 Typical cylindrical brachytherapy source [12] ........................................ 4
1.3 Single-slice CT data acquisition [14][15] .................................................. 5
1.4 Typical incident spectrum corresponding to 140 kVp ............................... 6
1.5 Relative intensity of X-ray photons [20] .................................................... 7
1.6 Illustration of different hardware approaches to dual-energy CT imaging [23] 9
2.1 Structure of DE-AM algorithm, J. A. O’Sullivan, MemBIS, 11/06/2008 ..... 13
2.2 (a) Illustration of three penalty functions (b) Linear region view for linear
region matching (c) Quadratic region view for linear region matching ......... 16
2.3 (a) Incident spectra (b) Attenuation coefficients of the component materials 19
2.4 Phantom image at 75 keV with four inserts (from the top, clockwise direction)
muscle, ethanol, Teflon and substance X ............................................... 20
2.5 (a) True polystyrene \(c_1(x)\) image (b) True calcium chloride \(c_2(x)\) image . 20
2.6 Reconstructed polystyrene \(c_1(x)\) images ............................................. 21
2.7 Reconstructed calcium chloride \(c_2(x)\) images .................................... 22
2.8 Reconstructions using regularized DE-AM algorithm \((\delta = 500, \lambda = 100)\) for noisy data with scatter (total 20 realizations) ......................... 25
2.9 Incident spectra with pixel values representing relative intensity or probability
for 2D Philips data .................................................................................. 26
2.10 Phantom image with pixel values representing the relative attenuation coefficients compared to water .......................................................... 27
2.11 Reconstructed images with pixel values representing attenuation coefficients
in mm⁻¹ evaluated at 43 keV (corresponding to 90 kVp) ........................ 28
2.12 Reconstructed images with pixel values representing attenuation coefficients
in mm⁻¹ evaluated at 60 keV (corresponding to 140 kVp) ........................ 29
2.13 Profile plots of the 341ˢᵗ row through the reconstructed linear attenuation images ................................................................. 30
2.14 (Zoomed in version of the profile plots of the 341ˢᵗ row through the reconstructed linear attenuation images) .................................................. 30
2.15 (a) Incident spectrum for the 3D Philips data (b) Multislice helical CT geometry (c) Vendor’s reconstructed image at the 34th slice out of a total of 67 slices with 3D Philips data 

2.16 Reconstructed images at the 75th slice out of a total of 150 slices by using (a) the FDK and (b) the unregularized monoenergetic AM 

2.17 (a) Ratio image of reconstructed image (the 75th slice out of a total of 150 slices) by using the FDK versus vendor (the 34th slice out of a total of 67 slices) with 3D Philips data (b) Ratio image of reconstructed image (the 75th slice out of a total of 150 slices) by using unregularized monoenergetic AM algorithm (66 OS) after 100 iterations versus vendor (the 34th slice out of a total of 67 slices) with 3D Philips data 

2.18 (a) Comparison of profile plots between vendors reconstruction, the FDK and the unregularized monoenergetic AM algorithms (b) Zoomed in profile of (a), showing the transition from the polystyrene insert to the water region 

2.19 Phantom specification for singular value decomposition analysis 

2.20 Correlation coefficient matrix for phantom with 8 uniform regions 

2.21 Absolute singular vectors for every region 

2.22 Hill-valley objective function surface illustration for two estimators problem 

2.23 (a) 2000 iterations along the valley with $Z = 0.1Z$ (b) 100 iterations along the valley with $Z = 0.005Z$ 

3.1 (a) True polystyrene ($c_1(x)$) image (b) True calcium chloride ($c_2(x)$) image 

3.2 Noiseless data reconstruction for (a) polystyrene ($c_1(x)$) by using the unregularized LIAM algorithm, (b) calcium chloride ($c_2(x)$) by using the unregularized LIAM algorithm, (c) polystyrene ($c_1(x)$) by using the unregularized DE-AM algorithm and (d) calcium chloride ($c_2(x)$) by using the unregularized DE-AM algorithm 

3.3 Noisy data reconstruction for (a) polystyrene ($c_1(x)$) by using the unregularized LIAM algorithm, (b) calcium chloride ($c_2(x)$) by using the unregularized LIAM algorithm, (c) polystyrene ($c_1(x)$) by using the unregularized DE-AM algorithm and (d) calcium chloride ($c_2(x)$) by using the unregularized DE-AM algorithm 

3.4 Relative absolute errors of synthesized attenuation coefficients by using (a) the unregularized LIAM with noiseless data, (b) the unregularized DE-AM with noiseless data, (c) the unregularized LIAM with noisy data and (d) the unregularized DE-AM with noisy data 

3.5 (a) $I$-divergence vs. iteration and (b) $I$-divergence vs. time for the unregularized LIAM algorithm and the unregularized DE-AM algorithm 

3.6 Reconstructed polystyrene ($c_1(x)$) and calcium chloride ($c_2(x)$) images using the regularized LIAM algorithm ($\delta = 50$) 

3.7 Reconstructed polystyrene ($c_1(x)$) and calcium chloride ($c_2(x)$) images using the regularized LIAM algorithm ($\delta = 500$)
3.8 Reconstructed polystyrene \((c_1(x))\) images using the regularized DE-AM algorithm \((\delta = 500)\) ................................................................. 56
3.9 Reconstructed calcium chloride \((c_2(x))\) images using the regularized DE-AM algorithm \((\delta = 500)\) ................................................................. 57
3.10 Profile plots corresponding to a horizontal line through the Substance X ................................................................. 59
3.11 Total objective function, \(I\)-divergence and penalty along iterations ................................................................. 61
3.12 Reconstructed polystyrene \((c_1(x)),\) left columns) and calcium chloride \((c_2(x),\) right columns) images using the regularized LIAM algorithm by sweeping \(\lambda\) in ascending order \((\delta = 50)\) with display window \([0.8\ 1.2]\) for \(c_1(x)\) and \([0,0.15]\) for \(c_2(x)\) ........................................................................................................ 66
3.13 Reconstructed polystyrene \((c_1(x),\) left columns) and calcium chloride \((c_2(x),\) right columns) images using the regularized LIAM algorithm by sweeping \(\lambda\) in ascending order \((\delta = 500)\) with display window \([0.8\ 1.2]\) for \(c_1(x)\) and \([0,0.15]\) for \(c_2(x)\) ........................................................................................................ 70
3.14 Reconstructed polystyrene \((c_1(x),\) left columns) and calcium chloride \((c_2(x),\) right columns) images using the regularized LIAM algorithm by sweeping \(\lambda\) in ascending order \((\delta = 500)\) with display window \([0.8\ 1.2]\) for \(c_1(x)\) and \([0,0.15]\) for \(c_2(x)\) ........................................................................................................ 74
3.15 Reconstructed polystyrene \((c_1(x),\) left columns) and calcium chloride \((c_2(x),\) right columns) images using the regularized LIAM algorithm by sweeping \(\lambda\) in descending order \((\delta = 50)\) with display window \([0.8\ 1.2]\) for \(c_1(x)\) and \([0,0.15]\) for \(c_2(x)\) ........................................................................................................ 78
3.16 Noisy data reconstruction for polystyrene \((c_1(x))\) ........................................................................................................ 79
3.17 Noisy data reconstruction for calcium chloride \((c_2(x))\) ........................................................................................................ 80
3.18 Relative estimation errors of linear attenuation coefficients obtained by (a) the unregularized LIAM with \(\beta = 0\) and (b) the unregularized DE-AM ................................................................. 80
3.19 Real data reconstructions for (a) \(c_1(x)\) from the unregularized LIAM with \(\beta = 0\), (b) \(c_2(x)\) by the unregularized LIAM with \(\beta = 0\), (c) \(c_1(x)\) from the unregularized DE-AM with 33 OS and (f) \(c_2(x)\) from the unregularized DE-AM with 33 OS ........................................................................................................ 81
3.20 Real data reconstructions for (a) \(c_1(x)\) from the unregularized LIAM with \(\beta = 1000\), (b) \(c_2(x)\) by the unregularized LIAM with \(\beta = 1000\), (c) \(c_1(x)\) from the regularized LIAM with \(\beta = 1000, \lambda = 3.3\), (d) \(c_2(x)\) from the regularized LIAM with \(\beta = 1000, \lambda = 3.3\), (e) \(c_1(x)\) from the regularized DE-AM with \(\lambda = 100, 33\) OS and (f) \(c_2(x)\) from the regularized DE-AM with \(\lambda = 100, 33\) OS ........................................................................................................ 83
3.21 Relative estimation errors of linear attenuation coefficients obtained by (a) the unregularized LIAM with \(\beta = 1000\), (b) the regularized LIAM with \(\beta = 1000, \lambda = 3.3\) and (c) the regularized DE-AM with \(\lambda = 100\) and 33 OS ........................................................................................................ 84
4.1 Simulation results for a random image with different \(\lambda\) values ................................................................. 98
4.2 True (a) \(c_1(x)\) and (b) \(c_2(x)\) images ........................................................................................................ 99
4.3 Reconstructed images for the Case 1 problem ........................................................................................................ 100
List of Tables

2.1 Relationship between the four quantities ................................. 15
2.2 Component coefficients for the phantom in Figure 2.4 ...................... 23
2.3 Comparison between the unregularized and the regularized DE-AM algorithms 23
2.4 Single value for different phantom regions ................................. 37

3.1 Component coefficients of the phantom for mini CT geometry ............... 47
3.2 Relative absolute estimation errors (%) of $c_1(x)$ and $c_2(x)$ .................. 50
3.3 Component coefficients for the physical cylinder phantom .................... 66
3.4 Comparison between the unregularized LIAM ($\beta = 0$) and DE-AM algorithms 78
Acknowledgments

I want to express my great thanks first to my advisor, Professor Joseph A. O’Sullivan for his guidance and support. His profound knowledge and scope inspire and encourage me to explore more into my research area.

I am grateful to everyone in our research group, Dr. David G. Politte, Dr. Jeffrey F. Williamson, Dr. Joshua D. Evans, Dr. Bruce R. Whiting, Dong Han and Yaduo Yu. Without their support, I could never make contributions to my research.

Moreover, suggestions and advice coming from my committee members reinforce my understanding and speed up my progress on my work. Therefore, I would like to thank them for their time and effort: Dr. Joseph A. O’Sullivan, Dr. Jeffrey F. Williamson, Dr. R. Martin Arthur, Dr. Hiro Mukai and Dr. Mark A. Anastasio.

Dr. David G. Politte read every page of this dissertation multiple times and provided invaluable edits and feedback.

I also enjoy conversations with my group colleagues, Dr. Alan D. Kaplan, Dr. Ikenna Odinaka, Dr. Po-Hsiang Lai, Soysal Degirmenci and Daheng Li. I would like to thank them for their selfless help and willingness to share thoughts and ideas.

Last but not the least, I want to thank my family. My husband, Chong Dong, sacrificed a lot to make this happen. My mother, Miss Hezhi Chen, was my spiritual support and encouraged me whenever I was frustrated. My little boy, Eric, has been a great comfort ever since he was born. My aunt, Miss Hesi Chen, helped me taking care of my Eric.

This work is partially supported by NIH grant R01 CA149305.

Yaqi Chen

Washington University in Saint Louis
August 2014
Dedicated to my mother and my husband.
ABSTRACT OF THE DISSERTATION

Alternating Minimization Algorithms for
Dual-Energy X-Ray CT Imaging and Information Optimization

by

Yaqi Chen

Doctor of Philosophy in Electrical Engineering

Washington University in St. Louis, August 2014

Professor Joseph A. O’Sullivan, Chair

This dissertation contributes toward solutions to two distinct problems linked through the use of common information optimization methods. The first problem is the X-ray computed tomography (CT) imaging problem and the second is the computation of Berger-Tung bounds for the lossy distributed source coding problem. The first problem discussed through most of the dissertation is motivated by applications in radiation oncology, including dose prediction in proton therapy and brachytherapy.

In proton therapy dose prediction, the stopping power calculation is based on estimates of the electron density and mean excitation energy. In turn, the estimates of the linear attenuation coefficients or the component images from dual-energy CT image reconstruction are used to estimate the electron density and mean excitation. Therefore, the quantitative accuracy of the estimates of the linear attenuation coefficients or the component images affects the accuracy of proton therapy dose prediction.
In brachytherapy, photons with low energies (approximately 20 keV) are often used for internal treatment. Those photons are attenuated through their interactions with tissues. The dose distribution in the tissue obeys an exponential decay with the linear attenuation coefficient as the parameter in the exponential. Therefore, the accuracy of the estimates of the linear attenuation coefficients at low energy levels has strong influence on dose prediction in brachytherapy.

Numerical studies of the regularized alternating minimization (DE-AM) algorithm with different regularization parameters were performed to find ranges of the parameters that can achieve the desired image quality in terms of estimation accuracy and image smoothness. The DE-AM algorithm is an extension of the AM algorithm proposed by O’Sullivan and Benac [1]. Both simulated data and real data reconstructions, as well as system bias and variance experiments [2][3], were carried out to demonstrate that the DE-AM algorithm is incapable of reconstructing a high density material accurately with a limited number of iterations (1000 iterations with 33 ordered subsets). This slow convergence phenomenon was then studied via a toy or scaled-down problem, indicating a highly ridged objective function.

Motivated by the studies which demonstrate the slow convergence of the DE-AM algorithm, a new algorithm, the linear integral alternating minimization (LIAM) algorithm was developed, which estimates the linear integrals of the component images first; then the component images can be recovered by an expectation-maximization (EM) algorithm or linear regression methods. Both simulated and real data were reconstructed by the LIAM algorithm while varying the regularization parameters to ascertain good choices ($\delta = 500, \lambda = 50$ for $I_0 = 100000$ scenario). The results from the DE-AM algorithm applied to the same data were used for comparison. While using only 1/10 of the computation time of the DE-AM algorithm,
the LIAM algorithm achieves at least a two-fold improvement in the relative absolute error of the component images in the presence of Poisson noise.

This work also explored the reconstruction of image differences from tomographic Poisson data. An alternating minimization algorithm was developed and monotonic decrease in the objective function was achieved for each iteration. Simulations with random images and tomographic data were presented to demonstrate that the algorithm can recover the difference images with 100% accuracy in the number of and identity of pixels which differ. An extension to 4D CT with simulated tomographic data was also presented and an approach to 4D PET was described.

Different approaches for X-ray adaptive sensing were also proposed and reconstructions of simulated data were computed to test these approaches. Early simulation results show improved image reconstruction performance in terms of normalized $L_2$ norm error compared to a non-adaptive sensing method.

For the second problem, an optimization and computational approach was described for characterizing the inner and outer bounds for the achievable rate regions for distributed source coding, known as Berger-Tung inner and outer bounds. Several two-variable examples were presented to demonstrate the computational capability of the algorithm. For each problem considered that has a sum of distortions on the encoded variables, the inner and outer bound regions coincided. For a problem defined by Wagner and Anantharam [4] with a single joint distortion for the two variables, their gap was observed in our results. These boundary regions can motivate hypothesized optimal distributions which can be tested in the first order necessary conditions for the optimal distributions.
Chapter 1

Introduction

1.1 Proton Therapy

1.1.1 Proton Therapy Overview

Proton therapy is a type of radiation therapy which uses high energy beams of protons to treat tumors. A particle accelerator is used during treatment to energize protons. Typically, the energized protons have energies in the range of 70 to 250 MeV. Proton beams have many good properties that can be utilized in radiation therapy. Protons have little side scatter in the tissue with narrow beams. When a proton of a given energy is penetrating through the body, it deposits little dose along the way. At some point, it deposits almost all of its dose and this maximum dose is called Bragg peak. In most treatments, protons of different energies which have Bragg peaks at different depths are applied in order to treat the entire tumor. The thin lines in Figure 1.1a to the right show these Bragg peaks. The dose of total radiation is called the Spread-Out Bragg Peak (SOBP), which can be seen as a heavy dashed line in Figure 1.1a to the right. The depth-dose curve of an X-ray beam in solid red line is also provided for comparison. The shaded area represents the extra dose deposited by X-ray radiotherapy. The extra dose can be the source of damage to normal tissues. Proton therapy is extraordinarily expensive and the facility can cost from $20M to $200M to construct. Up to September 2012, there were 39 treatment centers all over the world and there is a new site at Washington University in Saint Louis that began treating patients in December 2013.
Figure 1.1: (a) Dose versus depth for protons. Figure is adapted from [5]. (b) Relative dose versus depth for protons of the same energy from different materials. Central axis dose, 5×5 cm² beam, 250 MeV protons, phantoms of constant water-equivalent electron density, but variable atomic compositions. Figure 1.1(b) is used with permission of Jeffrey V. Siebers and Jeffrey F. Williamson (unpublished).
1.1.2 Dose Prediction in Proton Therapy

As in other radiation oncologic applications [6], dose prediction is very important in proton therapy. Particle trajectories for computing radiation dose depend on materials. Figure 1.1b gives the relative dose versus depth for different materials for photons of the same energy. Stopping power determines energy loss per unit length, and therefore dose delivered. The uncertainty of the estimation in the Bragg peak position is due to the estimation uncertainty in the stopping power. According to [7], current Stoichiometric single energy stopping power CT mapping yields 2 to 3 mm of range uncertainty. A recent study [8] showed the advantage of using dual-energy X-ray CT for dose calculations. Based on Bethe’s formula [9], the goal is to use the linear attenuation coefficient estimation from dual-energy CT to estimate the electron density and mean excitation energy, thereby reducing the range uncertainty. Torikoshi et al. [10] used a non-separable equation for the effective atomic number and the electron density of tissues, and empirically determined the mean excitation energy value dependence on effective atomic number. However, it requires iterative numerical solution making it impractical for statistical iterative CT image reconstruction. Han et al. [11] derived a two-parameter model using the reconstructed component images from dual-energy CT. They estimate the electron density using the reconstructed component images and estimate the mean excitation energy using the obtained electron density, with an additional correction factor for the mean excitation energy.

1.2 Brachytherapy Overview

Brachytherapy is a type of radiotherapy that places radioactive seeds close to the targeted tissue or tumor. The delivered dose drops off very sharply with increasing distance from the seeds. Therefore, a relatively high dose can be delivered to a target region over a short time while at the same reducing the radiation exposure to the surrounding healthy tissues [12]. The dose delivered to tissue depends on several factors: the strength of the radioactive seeds, placement duration, tissue composition, etc. Figure 1.2 shows the four principle factors that affect the relative dose distribution for brachytherapy. According to Figure 1.2, the delivered dose is attenuated by the surrounding tissues, obeying an exponential decay. In permanent interstitial brachytherapy, low energies photons are used (approximately 20 keV) to allow
fast decay of the radiation. Therefore, the estimation accuracy of the linear attenuation coefficient of the tissues at low energies has a direct effect on the accuracy of dose prediction in brachytherapy.

1.3 Transmission Tomography

1.3.1 X-ray CT Overview

In general, an X-ray CT uses an external source of X-rays and an array of detectors on the opposite side. For helical CT geometries, the source of X-rays repeats the illumination at some evenly distributed source angles on a circle, while the patient table is moving at a constant speed at a direction that is perpendicular to the plane that the source angles form. Figure 1.3 shows single-slice CT (3rd generation) data acquisition. Modern CT systems can finish the scan of a patient from head to toe in just a few seconds with an isotropic spatial resolution better than half a millimeter [13].
Modern CT uses two types of detectors, which are energy-integrating detectors and photon-counting detectors. Energy-integrating detectors measure the energy flux of the received photons and the photon-counting detectors can give the number of received photons corresponding to some discrete energy levels.

Each CT image pixel represents the linear attenuation coefficient at that particular location and sinograms record the linear attenuation coefficient line integral along the ray connecting each source to each detector.

X-rays emitted from tubes are not monoenergetic, instead, the distribution of the photon energies obeys a spectrum [16][17]. Figure 1.3 gives a typical incident X-ray spectrum corresponding to 140 kVp. The photons at lower energies are more likely to be absorbed first because the linear attenuation coefficient of the material is higher for low photon energies. Therefore, at photons penetrate through an object, the mean photon energy coming out of the object is higher. This is referred as beam hardening phenomenon and it is the source of many image artifacts, such as cupping artifact and streaking artifact. The beam hardening [18][19] phenomenon can be illustrated in Figure 1.5. According to this figure, some very low energy photons are absorbed by the material of the anode. Assume the relative intensity of photons at energy $E$ is $I(E)$, then the intensity received after some path $l$ is governed by
Beer’s law,

\[ I_{\text{out}}(E) = I(E) \exp \left[ - \int I \mu(x, E) \, dx \right], \quad (1.1) \]

where \( \mu(x, E) \) is the linear attenuation coefficient at a particular location, \( x \), and a particular energy level, \( E \).

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{incident_spectrum.png}
\caption{Typical incident spectrum corresponding to 140 kVp}
\end{figure}

### 1.3.2 Statistical Image Reconstruction for Transmission Tomography

In transmission tomography, we are interested in finding spatial distribution of the linear attenuation coefficients, \( \mu \), in the scanning object. If we assume the object is static, the ideal transmission imaging modality would provide a complete description of \( \mu \) for a wide range of energies, at infinitesimal spatial, at a modest price, and with no harm to the subject [2].

After a transmission scan is acquired, the goal of reconstruction is to estimate the linear attenuation coefficients from measurements corresponding to source-detector pairs. Since only a finite number of measurements are available, discretization of image space is required. The conventional basis is a “pixel” for 2D images or a “voxel” for 3D images. Conventional analytical reconstruction approaches, such as the filtered backprojection (FBP) algorithm,
assume a linear relationship between attenuation coefficients and measurements. This inaccurate assumption leads to many reconstruction problems, such as streaking artifacts and noisy reconstructed images in low-count scenarios. It might result in more severe problems in some special situations, such as a missing data scenario.

With so many limitations and constraints on analytical approaches, it is natural to consider the reconstruction problem as a statistical problem, especially when noise is high enough. By applying more accurate models of the physics, statistical methods tend to significantly reduce image artifacts and achieve the same quantitative accuracy with lower dose.

Statistical reconstruction methods can fall into different categories. The maximum-likelihood (ML) estimation method [21] is a natural approach that finds $\mu$ from a particular measurement such that it maximizes the probability of having observed this measurement. A standard statistical model is that measurements are Poisson distributed with means following Beer’s law. However, maximizing the log-likelihood alone will lead to unacceptably noisy images. This is because the reconstruction problem itself is ill-conditioned, i.e., there are many solutions that can fit the log-likelihood very well. Among those solutions, many are not physically feasible. Therefore, certain regularizations must be incorporated in order to
yield a unique image. Piecewise smoothness is an image property that is good for image segmentation, meaning that pixels or voxels near each other tend to or should have the same or similar intensities. This can be treated as prior information, and in Bayesian learning [22], a maximum a posteriori (MAP) estimate of $\mu$ can be obtained by maximizing some posterior distribution. This maximization can be realized by adding regularization or a penalty to the log-likelihood function.

In practice, statistical methods are not superior to analytical methods in every aspect and there are many tradeoffs. The most important one is computation time. Most statistical methods are iterative, and computation time is the most significant drawback that limits its application in clinical use. Moreover, complexity issues, both in the physical and mathematical models and in the software with which the methods are implemented, should also be taken into account.

### 1.4 Dual-Energy X-ray CT

Dual-energy X-ray CT is a technique used for diagnostic imaging purposes which uses two sets of data corresponding to two different X-ray spectra. The two sets of measurements corresponding to two spectra are statistically independent of each other.

#### 1.4.1 Dual-Energy X-ray CT Data Acquisition

Figure 1.6 illustrates different hardware approaches to dual-energy CT imaging acquisition. In rapid kVp switching approach, the X-ray tube voltage is rapidly changed at different kVp levels, producing spectra corresponding to lower and higher energy level (left). In the layered detector approach, energy-sensitive layer detectors are superimposed on each other, allowing higher energy photons to penetrate through the top layer and being detected by the bottom layer (middle). The lower energy photons will be absorbed by the top layer. The dual-source CT approach uses two X-ray tubes and the corresponding detectors are arranged at an angular off-set (right). The X-ray tubes are operated at two different voltage levels, allowing simultaneous dual-energy data acquisition [23].
1.4.2 Dual-Energy X-ray CT Algorithms

Alvarez and Macovski proposed the basis material decomposition (BMD) algorithm in 1976 [24], and it became the most prevalent algorithm in spectral CT including dual-energy CT. Two broad categories of the BMD algorithm are projection-based BMD [24][25][26] and image-based BMD [27]. For the case of \( I = 2 \) basis materials, dual-energy CT data are acquired. In projection-based BMD, the basis material model replaces the linear attenuation coefficients in the mathematical expressions of measurements which are governed by Beer’s law. Then the line integrals of the coefficients \( c_1(x) \) and \( c_2(x) \) can be computed based on the measurements. Finally, the coefficients \( c_1(x) \) and \( c_2(x) \) can be calculated. In image-based BMD, the coefficients \( c_1(x) \) and \( c_2(x) \) can be expressed as the product of a 2 by 2 matrix \( T \) \((T = K^{-1}, K_{i,j} = \int_E S_j(E)f_i(E)dE)\) and two sets of input image data corresponding to two spectra. To be more specific, the input image data can be expressed as \( \int_E S_j(E)\mu(x,E)dE \), where \( S_j(E) \) is the normalized spectrum with \( \int_E S_j(E)dE = 1 \) and \( j \) indexes the spectrum.

In general, the BMD algorithm assumes the linear attenuation coefficient \( \mu(x,E) \) of the scanned object can be expressed as a linear combination of some basis functions,

\[
\mu(x,E) = c_1(x)f_1(E) + c_2(x)f_2(E) + \cdots + c_i(x)f_i(E) + \cdots + c_I(x)f_I(E),
\]

(1.2)

where \( f_1, f_2, \cdots, f_i, \cdots, f_I \) are the linearly independent basis material attenuation functions and \( c_1, c_2, \cdots, c_i, \cdots, c_I \) are the corresponding coefficients. While the choice of these basis functions can be very empirical, in particular, Alvarez and Macovski recommended using a
specific choice of two basis functions. One is $1/E^3$, corresponding to photoelectric effect, and the other is $f_{KN}(E)$ (Klein-Nishina function [28]) describing the Compton scattering. However, later studies show that these particular choices of basis functions cannot achieve the desired accuracy for energies in the range of 30 keV to 140 keV [29][30][31], which limits its practical application. Of course, many other choices of basis functions were studied extensively to model the linear attenuation coefficients of different materials. Actually, the projection-based BMD algorithm can be very sensitive to the deviation between the choice of basis functions and the true composition of the object. A mismatch between them can cause systematic errors as well as image artifacts such as streaking [13]. For example, if the linear attenuation coefficients set for bone is one of the basis functions, then for fat tissue, the corresponding reconstructed coefficient for bone can be negative. Moreover, since the projection-based BMD algorithm neglects noise characteristics, it can produce very noisy estimates in the low dose scenario [32]. Therefore, Noh et al. proposed to use penalized weighted least squares and penalized likelihood methods based on statistical models in order to reduce the noise in the estimates [32].

Instead of using a linear combination of basis functions, the density $\rho(x)$ that describes the morphology of the objects and atomic number $Z(x)$ that describes the material distribution, can be used as material characterization values, known as the RZP method [33]. Heismann showed that the model mismatch for linear attenuation coefficients affects only local accuracy, thereby making it better suited for general practical applications [33].

1.5 Dissertation Organization

Chapter 2 through Chapter 5 consider reconstruction problems in transmission tomography and emission tomography. Chapter 2 and Chapter 3 focus on image reconstruction for dual-energy CT. In Chapter 2, an edge-preserving penalty function is introduced to the DE-AM algorithm. Reconstructed results are shown from simulated data and real data by using the unregularized and regularized DE-AM algorithms. The analysis for the DE-AM algorithm is to find the reason for its slow convergence. Based on the analysis, a new algorithm, called LIAM, is proposed in Chapter 3, with reconstruction results from both simulated and real data. The performances of the LIAM algorithm and the DE-AM algorithm are
compared in Chapter 3. In Chapter 4, the problem of reconstructing image differences using a linear regression model is considered, and the results from simulated tomographic data are displayed. A similar monoenergetic CT model problem is also considered and simulated data reconstructions are presented. Moreover, an algorithmic extension to 4D PET is proposed. In Chapter 5, the adaptive sensing problem for X-ray imaging is considered and simulated data reconstructions are exhibited.

Chapter 6 considers a lossy distributed source coding problem. The Berger-Tung inner and outer bounds are computed for several problems by using a proposed alternating minimization algorithm.
Chapter 2

Dual-Energy AM Algorithm Reconstruction

2.1 Dual-Energy AM (DE-AM) Algorithm

Denote the source-detector pairs by \( y \), pixels by \( x \) and X-ray spectra by \( j \in \{1, 2\} \). The transmission data \( d_j(y) \) are assumed to be Poisson distributed with mean \( g_j(y : \mu) \) \cite{34}\cite{35},

\[
g_j(y : \mu) = \sum_E I_{0j}(y, E) \exp \left( - \sum_x h(y|x)\mu(x, E) \right) + \beta_j(y),
\]

where the line integrals in the forward model are approximated using a discrete point-spread function \( h(y|x) \), with units given in mm; the attenuation function \( \mu(x, E) \), with units of \( \text{mm}^{-1} \) is modeled as a linear combination of two component materials, i.e., \( \mu(x, E) = \sum_{i=1}^2 \mu_i(E)c_i(x) \); \( I_{0j} \) represents the unattenuated photon counts or intensity and the mean background events are denoted by \( \beta_j(y) \).

Therefore, we can write down the log-likelihood function as

\[
l(d : \mu) = \sum_{j=1}^2 \sum_y \left( d_j(y) \ln g_j(y : \mu) - g_j(y : \mu) \right).
\]

Our goal is to maximize the log-likelihood function \cite{36} over \( c_i(x) \) which are nonnegative and represent the component coefficients.
Since maximizing the log-likelihood function is equivalent to minimizing $I$-divergence, our goal becomes to minimize the following $I$-divergence \[ I(d || g(y : c)) = \sum_{j=1}^{2} \sum_{y} \left( d_j(y) \ln \frac{d_j(y)}{g_j(y : c)} - d_j(y) + g_j(y : c) \right). \] (2.3)

Readers are encouraged to refer to [1] for a detailed derivation of the algorithm for a polyenergetic version of the alternating minimization algorithm which can easily be extended to a dual-energy alternating minimization (DE-AM) algorithm [46][47][27].

The update for DE-AM algorithm can be written as

\[ \hat{c}_i^{(k+1)}(x) = \max \left( 0, \hat{c}_i^{(k)}(x) - \frac{1}{Z_i(x)} \ln \frac{\sum_{j=1}^{2} \hat{b}_{ij}(x)}{\sum_{j=1}^{2} \hat{b}_{ij}^{(k)}(x)} \right), \] (2.4)

where $\hat{b}_{ij}(x)$ and $\hat{b}_{ij}^{(k)}(x)$ are the backprojections of the measured data and estimate data, respectively, and $Z_i(x)$ is a normalization constant that guarantees convergence. We illustrate the structure of the DE-AM algorithm in Figure 2.1.
2.2 Regularized DE-AM Algorithm

However, it has been observed that as AM iterations continue, the reconstructed image tends to become noisier. This motivates researchers to incorporate neighboring pixel interactions in the algorithm to explicitly trade off image smoothness and data fitting.

We used the edge-preserving penalty function

\[ \psi(t) = \frac{1}{\delta^2} \left( |\delta t| - \ln(1 + |\delta t|) \right), \]  

(2.5)

which has also been used by other researchers [48][49][50]. In this penalty, \( t \) is the intensity difference between neighboring pixels/voxels, \( \delta \) is a parameter that controls the transition (cutoff point is at \( 1/\delta \)) between a quadratic region (for small \( t \)) and a linear region (for larger \( t \)) and we will explore later the effects of \( \delta \) on the reconstructed image. This penalty function has a lot of desirable properties [48]; for example, it’s convex and twice continuously differentiable [51][52][53]. This penalty function is actually a modified version of the well-known log-cosh function [3] that is quadratic for small pixel variation and linear for larger variations, so that it can suppress image noise while at the same time preserving desired edge boundaries. To be more specific, larger \( \delta \) will result in the transition occurring at smaller pixel variation and vice versa. Therefore, the choice of \( \delta \) largely determines the reconstruction image quality, especially for sharp edge boundaries, i.e. from bone to tissue.

Now we consider four quantities associated with this penalty, which are the weight, the cutoff point (corresponding to \( t \)), the quadratic weight (the weight of the penalty in quadratic region) and the linear weight (the weight of the penalty in linear region). By specifying any two of the four quantities, we know the other two. Table 2.1 gives various cases of knowing any two of the four quantities. For example, when the weight is \( \lambda \) and the cutoff is \( \delta \), then for \( t < \delta \), the penalty function can be approximated as \( \frac{\lambda}{\delta^2} t^2 \) with the quadratic weight equal to \( \frac{\lambda}{\delta^2} \). Similarly, for \( t > \delta \), the penalty function can be approximated as \( \frac{\lambda}{\delta} t \) with the linear weight equal to \( \frac{\lambda}{\delta} \).

For better visualization and comparison, three different penalty functions are plotted in Figure 2.2a. In this figure, \( \delta = 500 \) for both our penalty function and the Huber loss function \( (\psi(t) = \frac{1}{2} t^2, |t| \leq \frac{1}{\delta}; \psi(t) = \frac{1}{\delta}(|t| - \frac{1}{\delta^2}), |t| > \frac{1}{\delta}) \) and \( \lambda = 1 \) for all the functions.
Table 2.1: Relationship between the four quantities

<table>
<thead>
<tr>
<th></th>
<th>Weight</th>
<th>Cutoff</th>
<th>Quadratic Weight</th>
<th>Linear Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$</td>
<td>$\delta$</td>
<td>$\frac{\lambda}{2\delta}$</td>
<td>$\frac{\lambda}{\delta}$</td>
<td></td>
</tr>
<tr>
<td>$\delta^2q$</td>
<td>$\delta$</td>
<td>$\frac{p}{\delta}$</td>
<td>$\delta q$</td>
<td></td>
</tr>
<tr>
<td>$\delta p$</td>
<td>$\delta$</td>
<td>$\frac{p}{\delta}$</td>
<td>$p$</td>
<td></td>
</tr>
<tr>
<td>$\frac{\mu}{q}$</td>
<td>$\frac{q}{\mu}$</td>
<td>$\frac{q}{\mu}$</td>
<td>$p$</td>
<td></td>
</tr>
<tr>
<td>$\lambda$</td>
<td>$\frac{\lambda}{\mu}$</td>
<td>$\frac{p}{\lambda}$</td>
<td>$p$</td>
<td></td>
</tr>
<tr>
<td>$\lambda$</td>
<td>$\sqrt{\frac{2}{q}}$</td>
<td>$\frac{q}{2}$</td>
<td>$\sqrt{\lambda q}$</td>
<td></td>
</tr>
</tbody>
</table>

From Figure 2.2a, it can be seen that with the same $\lambda$ value ($\lambda = 1$), the log-cosh penalty function ($\psi(t) = \log[\cosh(t)] = \log(e^t + e^{-t})$) has a much wider quadratic region compared to the other two. We can match the linear part of the three penalty functions and observe the differences in the quadratic region. The matching and comparison can be seen in Figure 2.2b and Figure 2.2c. For our penalty function, $\lambda = 400, \delta = 500$, for the Huber loss function, $\lambda = 40, \delta = 5000$ and for the log-cosh penalty function, $\lambda = \frac{8}{9.5}$. Since for linear region matching, both our penalty function and the Huber loss function will be approximated as $\frac{\lambda|d|}{\delta}$, as long as we maintain the same ratio of $\lambda$ over $\delta$, the linear region will match for those two penalty functions. Therefore, the comparison between the quadratic regions for those two penalty functions can be very flexible. However, the log-cosh penalty function has a relatively larger quadratic region.

Our complete objective function, including the regularization term, is

$$\Phi_{AM}(c) = I(d\|g(y : c)) + \lambda R(c)$$

$$= \sum_{j=1}^{2} \sum_{y} \left( d_j(y) \ln \frac{d_j(y)}{g_j(y : c)} - d_j(y) + g_j(y : c) \right)$$ (2.6)

$$+ \lambda \sum_{i=1}^{2} \sum_{x} \sum_{k \in N_x} w_{x,k} \psi \left( c_i(x) - c_i(k) \right),$$

where $\lambda$ is a scalar that reflects the amount of smoothness desired, controlling the tradeoff between data fitting ($I$-divergence) and image smoothness (regularization); $N_x$ is the 8-pixel neighborhood surrounding pixel $x$ for 2D reconstruction or the 26-voxel neighborhood.
Figure 2.2: (a) Illustration of three penalty functions (b) Linear region view for linear region matching (c) Quadratic region view for linear region matching
surrounding voxel \( x \) for 3D reconstruction. The weights \( w_{x,k} \) control the relative contribution of each neighbor and are set to be the inverse distance between pixel/voxel centers.

Readers may notice that in (2.6), the \( c_i(x) \) are coupled in the regularization term and can't be updated in parallel as in the unregularized AM algorithm. Therefore, we used the convex decomposition lemma to find a surrogate penalty function [15] shown below, in which \( c_i(x) \) are decoupled.

\[
\psi(c_i(x) - c_i(k)) = \psi \left\{ \frac{1}{2} \left[ 2(c_i(x) - \hat{c}_i(x)) + (\hat{c}_i(x) - \hat{c}_i(k)) \right] \\
+ \frac{1}{2} \left[ -2(c_i(k) - \hat{c}_i(k)) + (\hat{c}_i(x) - \hat{c}_i(k)) \right] \right\} \\
\leq \frac{1}{2} \psi \left[ 2(c_i(x) - \hat{c}_i(x)) + (\hat{c}_i(x) - \hat{c}_i(k)) \right] \\
+ \frac{1}{2} \psi \left[ -2(c_i(k) - \hat{c}_i(k)) + (\hat{c}_i(x) - \hat{c}_i(k)) \right] \\
= \frac{1}{2} \left[ \psi(2c_i(x) - \hat{c}_i(x) - \hat{c}_i(k)) + \psi(2c_i(k) - \hat{c}_i(k)) \right].
\]

(2.7)

where \( \hat{c}_i(x) \) and \( \hat{c}_i(k) \) are the previous results for the corresponding pixels.

After replacing the original regularization term with the surrogate regularization term denoted by \( \hat{R}(c) \), we can obtain the modified surrogate objective function as

\[
\hat{\Phi}_{AM}(c) = I(d\|g(y : c)) + \lambda \hat{R}(c) \\
= \sum_{j=1}^{2} \sum_{y} \left( d_j(y) \ln \frac{d_j(y)}{g_j(y : c)} - d_j(y) + g_j(y : c) \right) \\
+ \frac{1}{2} \sum_{i=1}^{2} \sum_{x} \sum_{k \in N_x} w_{x,k} \left[ \psi(2c_i(x) - \hat{c}_i(x) - \hat{c}_i(k)) + \psi(2c_i(k) - \hat{c}_i(k)) \right].
\]

(2.8)

Because our penalty function is convex and twice continuously differentiable, Newton's method can be applied to minimize the above surrogate objective function. Of course, in this surrogate objective function, we still need the corresponding “surrogate” for the first data fitting term as seen in Appendix A.
Another issue is that the ordered subsets (OS) method [54] was implemented in our regularized DE-AM algorithm. The idea of OS is to use only a subset of the measured data for each update, and sequentially go through each subset partition, which can accelerate convergence approximately by the number of subsets. However, the number of OS can not be too large, because the OS method can not guarantee convergence, and too many OS tend to make the algorithm diverge. Note that if OS are in use, \( \lambda \) should be scaled down in each subset iteration by the number of subsets and we use the scaled \( \lambda \) in the following reconstruction.

2.3 Experiments and Reconstructions

All 2D images presented here are 512×512 with pixel size 1mm×1mm. All 2D simulations use \( I_0 = 100000 \) which corresponds to the number of unattenuated photons. The two component materials used are polystyrene \( (c_1(x)) \) and calcium chloride \( (c_2(x)) \). The attenuation coefficients spectra for the two components are shown in Figure 2.3b. The initial conditions are images with all 0’s.

2.3.1 Simulated Data Reconstruction in 2D

The Siemens Somatom Plus 4 scanner geometry is used for generating simulated data. There are 1408 source angles and 768 detectors in this geometry. Figure 2.3a shows the two incident energy spectra corresponding to tube voltages of 80 kVp and 140 kVp. The cross-section of the phantom image is shown in Figure 2.4. The pixel values represent the attenuation \( \mu(x, E) \) evaluated at 75 keV. The phantom consists of a thin Lucite shell that contains a cylindrical Lucite core in a water bath. The core has four openings for housing different materials. We simulate samples of muscle, ethanol, Teflon and material X with the two component materials (polystyrene and calcium chloride) in proportions listed in Table 2.2. The true component images for polystyrene \( (c_1(x)) \) and calcium chloride \( (c_2(x)) \) are given in Figure 2.5. A more detailed description of this phantom can be found in [47].

Figure 2.6 and Figure 2.7 give the reconstructed component images obtained by using the unregularized and regularized DE-AM \( (\delta = 500, \lambda = 100) \) algorithms with noiseless data,
Figure 2.3: (a) Incident spectra (b) Attenuation coefficients of the component materials
Figure 2.4: Phantom image at 75 keV with four inserts (from the top, clockwise direction) muscle, ethanol, Teflon and substance X

Figure 2.5: (a) True polystyrene ($c_1(x)$) image (b) True calcium chloride ($c_2(x)$) image
Figure 2.6: Reconstructed polystyrene \( c_1(x) \) images
(a) Noiseless, unregularized DE-AM
(b) Noiseless, $\delta = 500, \lambda = 100$

(c) Noisy, unregularized DE-AM
(d) Noisy, $\delta = 500, \lambda = 100$

(e) Noisy with scatter, unregularized DE-AM
(f) Noisy with scatter, $\delta = 500, \lambda = 100$

Figure 2.7: Reconstructed calcium chloride ($c_2(x)$) images
Table 2.2: Component coefficients for the phantom in Figure 2.4

<table>
<thead>
<tr>
<th>Substance</th>
<th>Polystyrene $c_1(x)$</th>
<th>Calcium Chloride $c_2(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water</td>
<td>0.90357</td>
<td>0.1357</td>
</tr>
<tr>
<td>Lucite</td>
<td>1.14</td>
<td>0.05834</td>
</tr>
<tr>
<td>Muscle</td>
<td>0.93995</td>
<td>0.13904</td>
</tr>
<tr>
<td>Ethanol</td>
<td>0.79904</td>
<td>0.03369</td>
</tr>
<tr>
<td>Teflon</td>
<td>1.4194</td>
<td>0.48779</td>
</tr>
<tr>
<td>Substance X</td>
<td>0.03</td>
<td>2.8613</td>
</tr>
</tbody>
</table>

noisy (Poisson) data and noisy data with scatter, respectively, after 1000 iterations. Here, we assume the scattered photons are Poisson distributed with the same mean for each source-detector pair. The scattered to primary photons have a ratio of 0.5%, with the primary photon reading at the central source-detector pair of the transmission data. The ordered subsets (OS) technique is enabled for every individual reconstruction. The acceleration factor is approximately proportional to the number of subsets. It’s worth noting that $\lambda$ should be scaled down in each subset-iteration by the number of subsets, and the value $\lambda = 100$ is the scaled down value for a total of 44 OS. From Figure 2.6 and Figure 2.7, we can see that both unregularized and regularized DE-AM algorithms can reconstruct images with noisy data with scatter. The regularized DE-AM algorithm produces smoother images with some choices of regularization parameters ($\delta, \lambda$). However, the reconstruction of the very high density material (substance X) is still problematic. The fact is that high density material is very rarely seen in the real application since we are motivated by radiation oncology. Table 2.3 summarizes the statistics in terms of $I$-divergence which has been normalized to $I_0 = 1$, and the penalty (weight excluded) for all the simulation reconstructions. According to Table 2.3, the regularized DE-AM algorithm produces much smoother images with little sacrifice

Table 2.3: Comparison between the unregularized and the regularized DE-AM algorithms

<table>
<thead>
<tr>
<th>Method</th>
<th>Data Model</th>
<th>$I$-divergence</th>
<th>Penalty</th>
</tr>
</thead>
<tbody>
<tr>
<td>unregularized DE-AM</td>
<td>Noiseless, scatter-free</td>
<td>48.3272</td>
<td>14.9195</td>
</tr>
<tr>
<td></td>
<td>Poisson noisy, scatter-free</td>
<td>58.9775</td>
<td>30.6853</td>
</tr>
<tr>
<td></td>
<td>Poisson noisy, constant scatter</td>
<td>59.0314</td>
<td>30.6169</td>
</tr>
<tr>
<td>Regularized DE-AM</td>
<td>Noiseless, scatter-free</td>
<td>48.3450</td>
<td>12.6683</td>
</tr>
<tr>
<td></td>
<td>Poisson noisy, scatter-free</td>
<td>59.0760</td>
<td>14.6387</td>
</tr>
<tr>
<td></td>
<td>Poisson noisy, constant scatter</td>
<td>59.1294</td>
<td>14.6102</td>
</tr>
</tbody>
</table>
in the $I$-divergence (insignificant increase of $I$-divergence). The statistics are consistent with our observation from the reconstructed images. Moreover, for noisy data with scatter, the regularized DE-AM algorithm produces much smoother images.

Both unregularized DE-AM algorithm and regularized DE-AM algorithm produce images with large bias for the estimations of high density material (Substance X). This is because only limited iterations (1000 iterations with 44 OS) were performed for the DE-AM algorithm. According to the results (Figure 4.5) from [55], with many more iterations (70000 iterations without OS for Figure 4.5 in [55]), the unregularized DE-AM algorithm can produce accurate estimations for the two component images from noisy data. The introduction of regularization will add extra bias to the estimations, see Figure 2.6.

In order to evaluate the system bias and variance [56], we ran the regularized DE-AM algorithm ($\delta = 500, \lambda = 100$) for 20 realizations of noisy data with scatter (scatter to primary ratio 0.5%). Figure 2.8 shows the reconstructed mean component images, the ratio images (mean reconstruction divided by truth) and standard deviation images. Results are obtained after 1000 iterations with 44 OS. According to Figure 2.8, the regularized DE-AM algorithm significantly overestimates $c_1(x)$ for substance X. For regions around substance X, overestimation for $c_2(x)$ and underestimation for $c_1(x)$ are observed. Higher standard deviations are observed for edge regions of $c_1(x)$ and reconstructions for $c_2(x)$ tend to have more uniform standard deviations over the whole region, except for the substance X rod and Teflon, which have relatively higher attenuation coefficients.

### 2.3.2 Real Data Reconstruction in 2D

Now we consider using real 2D data (data were collected by Yaduo Yu at Virginia Commonwealth University) obtained from the Philips Brilliance scanner with 1320 source angles and 816 detectors. The detectors from the Philips Brilliance scanner are energy-integrating [55][57][58]. Figure 2.9 shows the incident energy flux spectra for 90 kVp and 140 kVp scans, across different detectors. Figure 2.10 shows the phantom image used in this reconstruction. The pixel values represent the relative attenuation coefficients compared to water. The insert (intensity value around 1.9) in the 2D phantom is a Teflon rod. It is surrounded by water (intensity value around 1.0). Outside of the water tank is an elliptical acrylic ring (intensity
Figure 2.8: Reconstructions using regularized DE-AM algorithm ($\delta = 500$, $\lambda = 100$) for noisy data with scatter (total 20 realizations)
value around 1.1). The whole phantom is surrounded by air. Also, there’s a small gap between the water tank and the ring because they were not perfectly machined. Figure 2.11 and Figure 2.12 show the reconstructed attenuation coefficient images evaluated at 43 keV (corresponding to 90 kVp) and 60 keV (corresponding to 140 kVp), respectively, by using unregularized and regularized DE-AM algorithms after 500 iterations with 33 OS. The initial conditions are images with all 0’s. Both unregularized and regularized DE-AM algorithms can produce images with real 2D data from the Philips scanner. Different regularization parameters settings are also explored in this reconstruction with \( \lambda \) ranging from 1 to 1000 and with fixed \( \delta = 400 \). As the images are not sufficient to show the performance difference of changing regularization parameters, Figure 2.13 gives the profile plots of the 341st row of Figure 2.11 and Figure 2.12. Clearly, using larger value of \( \lambda \) produces smoother images. However, using too large a value for \( \lambda \) will lose sharp edges. For example, when \( \lambda = 1000 \), the feature of the air gap is not captured. In order to view the different performances with different regularization parameters, Figure 2.14 shows the zoomed-in view of Figure 2.13, which corresponds to the area around the Teflon rod. We conclude that \( \lambda = 100 \) is the best choice when \( \delta \) is fixed at 400. Most researchers choose the regularization parameters empirically. The optimal regularization parameters depend on the problem parameters and the object. From our reconstruction results, \( \lambda \approx 100, \delta \approx 500 \) can produce smooth images in most applications.
2.3.3 Real Data Reconstruction in 3D

In this reconstruction, we use real 3D data (data were collected by Yaduo Yu at Virginia Commonwealth University) obtained from the Philips Brilliance scanner with 2640 source angles per rotation, 816 detector columns and 16 detector rows. Unlike the previous two 2D reconstructions, only one set of data from the spectrum with 140 kVp (Figure 2.15a) is used for scanning. Reconstructions were performed using the Helical CT Advanced Reconstruction Engine (HECTARE), software developed by our former colleague Daniel B. Keesing [15]. Figure 2.15b gives the multislice helical CT geometry. The focus of the x-ray source rotates at a radius of $R_f$ from the isocenter, and the detector array, described by a circular arc of radius $R_f + R_d$, rotates in the same direction at a radius of $R_d$ from the isocenter. Using continuous coordinates, $\beta$ is the angle between the positive $x$ axis and the line connecting the focal spot to the isocenter. This angle is not restricted to $[0, 2\pi)$, but rather continues increasing with subsequent gantry rotations. A particular location on the curved detector array is specified by the fan angle, $\gamma$, and the cone angle, $\eta$. Finally, $z_{feed}$ is the axial distance traveled by the patient bed (or equivalently, the gantry) in one rotation. In this section, all displayed images have unit mm$^{-1}$. Figure 2.15c shows the vendor’s reconstruction at the 34th slice out of a total of 67 slices. The shell is the acrylic water tank. The core is made of the same material. The core has 4 inserts, air (upper right), presumably teflon (lower right), presumably polystyrene (lower left), and the upper left is unknown.
Figure 2.11: Reconstructed images with pixel values representing attenuation coefficients in mm$^{-1}$ evaluated at 43 keV (corresponding to 90 kVp)
Figure 2.12: Reconstructed images with pixel values representing attenuation coefficients in mm$^{-1}$ evaluated at 60 keV (corresponding to 140 kVp)
Figure 2.13: Profile plots of the 341\textsuperscript{st} row through the reconstructed linear attenuation images

Figure 2.14: (Zoomed in version of the profile plots of the 341\textsuperscript{st} row through the reconstructed linear attenuation images
Figure 2.16 shows the reconstructed images using the Feldkamp-Davis-Kress [59] (FDK) and unregularized monoenergetic AM [47] (after 100 iterations with 66 OS) algorithms. Images shown are the axial view at the 75th slice out of a total of 150 slices with pixels representing attenuation coefficients. Figure 2.18a gives the profile plots of the 235th column across the reconstructed images from the vendor, the FDK and the unregularized monoenergetic AM algorithms. Figure 2.18b especially shows the transition from the polystyrene to water. The FDK and AM algorithms have the same transition slope while the vendor’s reconstruction has a blurrier edge. However, the vendor’s reconstruction has less noise in the water region. From Figure 2.16, Figure 2.18a and Figure 2.18b, we conclude that HECTARE can reconstruct images with real 3D data from Philips, and the unregularized monoenergetic AM algorithm produces smoother images compared to the FDK. It’s promising that with regularization or DE-AM, even smoother images will be obtained. Figure 2.17 gives the co-registration between reconstructed images from HECTARE and the vendor. Note that the co-registration only applies to the nonzero pixels of the vendor’s reconstruction. In the air region, higher ratio values result from the very small reconstruction pixel intensities from the vendor and we can ignore those. Most ratio values of the shell and central core region are close to 1, indicating fairly good reconstruction. The four inserts exhibit “shifted-halos,” indicating a geometry mismatch between HECTARE and the Philips scanner.

2.4 Discussion

So far we have examined several reconstructions obtained using the DE-AM algorithm. While it can produce consistent, relatively smooth images by adding regularization, problems are associated with this algorithm. Overestimation and underestimation of the two components for the high density materials are observed. These two component images compensate each other and when they are synthesized to obtain the linear attenuation coefficient, low estimation error is achieved at relatively high energy levels (larger than 40 keV). However, at low energy levels (for example, 20 keV), the estimation error for the linear attenuation coefficient is very significant (as large as 40%).
Figure 2.15: (a) Incident spectrum for the 3D Philips data (b) Multislice helical CT geometry (c) Vendor’s reconstructed image at the 34th slice out of a total of 67 slices with 3D Philips data
Figure 2.16: Reconstructed images at the 75th slice out of a total of 150 slices by using (a) the FDK and (b) the unregularized monoenergetic AM algorithm.

Figure 2.17: (a) Ratio image of reconstructed image (the 75th slice out of a total of 150 slices) by using the FDK versus vendor (the 34th slice out of a total of 67 slices) with 3D Philips data (b) Ratio image of reconstructed image (the 75th slice out of a total of 150 slices) by using unregularized monoenergetic AM algorithm (66 OS) after 100 iterations versus vendor (the 34th slice out of a total of 67 slices) with 3D Philips data.
Figure 2.18: (a) Comparison of profile plots between vendors reconstruction, the FDK and the unregularized monoenergetic AM algorithms (b) Zoomed in profile of (a), showing the transition from the polystyrene insert to the water region

We now approach the explanations for these observations from two perspectives. First, we will try to depict the difficulty of the problem. Second, we will try to visualize how DE-AM approaches the solution.

The piecewise constant phantom shown in Figure 2.19 will be a useful explanation aid. Instead of estimating every pixel inside the phantom, we only look at the 8 uniform regions, assuming for every region, we only need to estimate two variable $c_1$ and $c_2$. Therefore, these 8 regions will have 16 total variables. The specifications of the regions are shown in Figure 2.19. Since the corresponding Fisher information matrix is the covariance matrix of the score function, we can visualize the corresponding correlation coefficient matrix as shown in Figure 2.20. The correlation coefficient matrix is calculated from the normalization of Fisher information matrix, i.e., every entry of the Fisher information matrix is divided by the product of the standard deviations of the corresponding variables. This procedure leads to the diagonal of the correlation coefficient matrix all 1’s. According to the obtained correlation coefficient matrix, estimators of different regions are positively correlated and, for every region, the two estimators are highly correlated (correlation coefficient is close to 1), seen as these 2 by 2 bright blocks on the diagonal of the correlation coefficient matrix.

Next, we can compute the singular value decomposition [60][61] of the Fisher information matrix associated with this phantom, in order to look at the singular values associated
Figure 2.19: Phantom specification for singular value decomposition analysis

Figure 2.20: Correlation coefficient matrix for phantom with 8 uniform regions
with the estimators for every region. This will allow us to see the correlation between the estimators of the same region. In Figure 2.21, every row of this matrix corresponds to one material/region in the order of $c_1$ and $c_2$. The columns, which correspond to the singular vectors, however, cannot easily be distinguished since they are arranged in the descending order of the singular values. By comparing the common parts between different columns, we can resolve some correspondence; see Table 2.4. For example, we can see that the air has significant value in column 6 and 14, so column 6 and 14 correspond to air region. The same rule can be applied to the other regions. According to Table 2.4, there is approximately a thousand-to-one ratio in singular values for the estimators of $c_1$ and $c_2$, indicating a huge correlation between the two estimators for the same region and a highly ridged objective function.

In order to better visualize the difficulty, we reduce our estimators to two, i.e., we assume that we know every value for every region except the left-most rod (the true $c_1$ and $c_2$ for the rod are 0.9 and 0.1, respectively) when we do reconstruction. Then we can plot the objective function surface in 3D as in Figure 2.22. The objective function surface has a hill-valley shape, i.e., it has very steep and very shallow regions. Six dotted lines on the surface correspond to the iterations of DE-AM algorithms at six different initial conditions with 1000 total iterations without OS. While DE-AM does monotonically decrease the objective
Table 2.4: Singlugar values for different phantom regions

<table>
<thead>
<tr>
<th>Region</th>
<th>Singular Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEK $c_1$</td>
<td>0.4169</td>
</tr>
<tr>
<td>MEK $c_2$</td>
<td>366.9400</td>
</tr>
<tr>
<td>Ethanol/Water $c_1$</td>
<td>0.5731</td>
</tr>
<tr>
<td>Ethanol/Water $c_2$</td>
<td>510.5024</td>
</tr>
<tr>
<td>Calcium Chloride Solution $c_1$</td>
<td>0.3933</td>
</tr>
<tr>
<td>Calcium Chloride Solution $c_2$</td>
<td>346.5505</td>
</tr>
<tr>
<td>$K_2HPO_4$ $c_1$</td>
<td>0.3305</td>
</tr>
<tr>
<td>$K_2HPO_4$ $c_2$</td>
<td>298.4948</td>
</tr>
<tr>
<td>PMMA Core $c_1$</td>
<td>0.0263</td>
</tr>
<tr>
<td>PMMA Core $c_2$</td>
<td>22.9275</td>
</tr>
<tr>
<td>Water $c_1$</td>
<td>0.0513</td>
</tr>
<tr>
<td>Water $c_2$</td>
<td>44.7430</td>
</tr>
<tr>
<td>PMMA Shell $c_1$</td>
<td>0.4535</td>
</tr>
<tr>
<td>PMMA Shell $c_2$</td>
<td>417.5676</td>
</tr>
<tr>
<td>Air $c_1$</td>
<td>0.1762</td>
</tr>
<tr>
<td>Air $c_2$</td>
<td>159.3043</td>
</tr>
</tbody>
</table>

function, and it drops down the hill very fast, however, after it gets to the valley, it is crawling very slowly along the valley, requiring many iterations to achieve the minimum. Even though the objective function is very small all along the valley, the estimators will change significantly along the value.

Figure 2.23 shows a possible speed up approach which tries to reduce the normalization factor $Z$ that controls the step size and the convergence rate when the estimator hits the valley. According to Figure 2.23, smaller $Z$ will give faster convergence under the condition that the valley has been found already. More specifically, with $Z$ value 20 times smaller, by running only $\frac{1}{20}$ of the original number of iterations (2000), similar progress along the valley can be achieved, compared to the larger $Z$ value running.
Figure 2.22: Hill-valley objective function surface illustration for two estimators problem

Figure 2.23: (a) 2000 iterations along the valley with $Z' = 0.1Z$ (b) 100 iterations along the valley with $Z' = 0.005Z$
2.5 Conclusions and Conclusions

The DE-AM algorithms were used to reconstruct 2D images from data simulated with noise and scatter for the geometry of the Siemens Somatom Plus 4 scanner. Choices for regularization parameters were explored using real 2D data from the Philips Brilliance scanner. The regularized DE-AM algorithm can produce a smoother image when appropriate regularization parameters are chosen. With the powerful tool of HECTARE, images were obtained with real 3D data from the Philips Brilliance scanner. Moreover, the unregularized polyenergetic AM algorithm produced smoother images compared to FDK, which promises even better reconstructions using regularization and DE-AM algorithms. However, a “shifted-halos” phenomenon appeared when co-registering the reconstructions from HECTARE with the vendor’s reconstruction. This indicates a geometry mismatch between HECTARE and the Philips scanner.

Although both unregularized DE-AM and regularized DE-AM algorithms can produce relatively smooth images, they both suffer from slow convergence. The reason for this slow convergence was studied using a toy problem with only two parameters to estimate. Visualization of the objective function surface shows this “hill-valley” characteristic, indicating an ill-conditioned problem setting. The DE-AM algorithm can potentially be sped up by using a more aggressive step size when it hits the “valley,” according to our simulated results for this toy problem. However, for multi-parameter problems, it is very difficult to determine if the “valley” has been found.

The implementation of unregularized DE-AM algorithm was built based on the work of Liangjun Xie [55]. I added the regularization for the DE-AM algorithm and did experimentations with a sequence of regularization parameters in order to find the appropriate range. The toy problem analysis was performed under the direction of Dr. Joseph. A. O’Sullivan.
Chapter 3

Line Integral Alternating Minimization (LIAM) Algorithm

3.1 Motivation

As seen in Chapter 2, while the polyenergetic and the regularized DE-AM algorithms can produce highly quantitatively accurate component images, the methods suffer from slow convergence. Even with the OS technique, it takes about 1000 full iterations to converge. One possible reason for slow convergence is that in the derivation of the DE-AM algorithm, for the second lifting (see Equation (A.11) and Equation (A.14)), which decouples the image voxels and the measurements, a normalization factor $Z_0$ is introduced to guarantee convergence of the algorithm, which makes the update step size small (see Equation (A.23)). Another possible reason for slow convergence is that, as shown in Chapter 2, the two component images are very highly correlated. Trying to break down this correlation might help to speed up the convergence.

In order to speed up the convergence, we first want to avoid the use of $Z_0$. One way to accomplish this is to estimate the linear integrals of the component images first, forgetting about the geometry (or point-spread function $h(y|x)$) for now. After obtaining the corresponding linear integrals for the component images, the component images can be recovered by the EM algorithm or other linear regression methods separately, thereby achieving our goal of breaking down the correlations.
However, in the first step, we totally ignore the geometry, and the resulting line integrals may not satisfy the geometry constraint, i.e., the resulting line integrals may not lie in the range space of $h(y|x)$ for feasible component images. This concern leads to the introduction of penalty which penalizes the differences between the estimated line integrals and the forward projections of the component images.

The rest of the chapter is organized as follows. We first introduce the data model, the problem definition, as well as the algorithm in Section 3.2. Section 3.3 sketches the regularized LIAM algorithm, which uses a penalty function similar to the one used in the regularized DE-AM algorithm. Sections 3.4 and 3.5 provide reconstructions with different regularization parameters.

### 3.2 Line Integral Alternating Minimization (LIAM) Algorithm

We propose to minimize $I$-divergence between the data and the estimated mean represented by line integrals $L$, subject to a soft geometry constraint $L = Hc$. The soft geometry constraint is introduced as the $I$-divergence between the line integral and forward projection of the image, with a Lagrange multiplier. Then by alternately updating $L$ and $c$ by keeping the other fixed, we can obtain converged line integrals and images. We can update integrals by using Newton’s method and update images by using an EM algorithm. If necessary, we can add a neighborhood penalty when updating images by using a trust region Newton’s method (the regularized EM algorithm).

#### 3.2.1 Data Model

Denote the source-detector pairs by $y$, pixels by $x$ and X-ray spectra by $j \in \{1, 2\}$. The transmission data $d_j(y)$ is assumed to be Poisson distributed with mean $F_j(L_1(y), L_2(y))$, where

$$F_j(L_1(y), L_2(y)) = \sum_E I_{0j}(y, E) \exp \left[ -L_1(y)\mu_1(E) - L_2(y)\mu_2(E) \right], \quad (3.1)$$
and the line integrals $L_i(y) = \sum_x h(y|x)c_i(x)$, $i \in \{1, 2\}$, and $c_i(x)$ is the component image; $I_{0j}$ represents the un-attenuated photon counts or intensity.

### 3.2.2 Problem Definitions

Similar to the approach described in Chapter 2, we use $I$-divergence to measure the discrepancy between the measured data and the estimated mean, i.e. we want to minimize

$$\min_{L_j} \sum_{j=1}^{2} I(d_j || F_j) = \min_{L_j} \sum_{j} \sum_{y} \left[ d_j(y) \log \frac{d_j(y)}{F_j(L_1(y), L_2(y))} - d_j(y) + F_j(L_1(y), L_2(y)) \right].$$

(3.2)

Next, we add the penalty to the above $I$-divergence measure in order to penalize the differences between the estimated line integrals and the forward projections of the component images,

$$\min_{L_i, c_i} \sum_{j=1}^{2} I(d_j || F_j) + \beta \sum_{i=1}^{2} I(L_i || Hc_i)$$

$$= \min_{L_i, c_i} \sum_{j} \sum_{y} \left[ d_j(y) \log \frac{d_j(y)}{F_j(L_1(y), L_2(y))} - d_j(y) + F_j(L_1(y), L_2(y)) \right]$$

$$+ \beta \sum_{i} \sum_{y} \left[ L_i(y) \log \frac{L_i(y)}{\sum_x h(y|x)c_i(x)} - L_i(y) + \sum_x h(y|x)c_i(x) \right],$$

(3.3)

where $\beta$ is the parameter that controls the tradeoff between the data fitting term and the penalty term.

Again, we use $I$-divergence as a penalty between the estimated line integrals and the forward projections of the component images. The advantage of using $I$-divergence is that if our model requires $c_i(x)$ to be nonnegative (which is the general case), then this formulation can automatically yield a nonnegative solution of $c_i(x)$ while a least squares measure or many other measures may need extra nonnegativity constraints.
3.2.3 Reformulation of Problem

By borrowing the result from Lemma A.0.1, we can reformulate our problem as

\[
\min_{L_i, c_i} \sum_{j=1}^{2} I(d_j || F_j) + \beta \sum_{i=1}^{2} I(L_i || Hc_i)
\]

\[
= \min_{p_j \in \mathcal{L}(d_j), f_j \in \mathcal{E}_j, c_i} \sum_{j=1}^{2} I(p_j || f_j) + \beta \sum_{i=1}^{2} I(L_i || Hc_i)
\]

\[
= \min_{p_j \in \mathcal{L}(d_j), f_j \in \mathcal{E}_j, c_i} \sum_{j=1}^{2} \sum_{y} \sum_{E} \left[ p_j(y, E) \log \frac{p_j(y, E)}{f_j(y, E)} - p_j(y, E) + f_j(y, E) \right]
\]

\[
+ \beta \sum_{i} \sum_{y} \left[ L_i(y) \log \frac{L_i(y)}{\sum_x h(y|x)c_i(x)} - L_i(y) + \sum_x h(y|x)c_i(x) \right],
\]

where \( \mathcal{L}(d_j) = \{ p_j(y, E) \geq 0 : \sum_{E} p_j(y, E) = d_j(y) \} \) is the linear family parameterized by \( d_j \) and \( \mathcal{E}_j = \{ f_j : f_j(y, E) = I_{0j}(y, E) \exp [-L_1(y)\mu_1(E) - L_2(y)\mu_2(E)] \} \) is the exponential family parameterized by \( L_i \).

In the above formulation, the estimation alternates between \( p_j, f_j \) and \( c_i \). Updating \( L_i \) is equivalent to updating \( f_j \) since \( f_j \) lies in the exponential family parameterized by \( L_i \).

3.2.4 Alternating Minimization Iterations

As stated before, the updating is split into three steps in every iteration, and the updating for \( p_j \) and \( c_i \) is easy.

The updating step for \( p_j \) is analogous to the updating step described for the DE-AM, which is

\[
\hat{p}_j^{(k)}(y, E) = d_j(y) \frac{\hat{f}_j^{(k)}(y, E)}{\sum_{E'} \hat{f}_j^{(k)}(y, E')}.
\]
The updating step for $c_i$ is the same as the EM algorithm update, which is

$$c^{(n+1)}_i(x) = \frac{\hat{c}_i^{(n)}(x)}{\sum_y h(y|x)} \sum_{y} h(y|x) \frac{\hat{L}^{(k+1)}_i(y)}{\sum_x h(y|x) \hat{c}_i^{(n)}(x)}.$$  \hspace{1cm} (3.6)

Note that here $k$ indexes the main iteration regarding updating $p_j, f_j$ and $c_i$, and $n$ indexes the iteration inside the EM algorithm to recover $c_i$ alone, for fixed $p_j$ and $f_j$.

In terms of updating $f_j$ or $L_i$, we use Newton’s method, since the problem is strictly convex and one extra lifting from the DE-AM algorithm can be avoided. The gradient for $[L_1(y); L_2(y)]$ is

$$\nabla_i(y) = \sum_{j} \sum_{E} \hat{p}_j(y, E) \mu_i(E) - \sum_{j} \sum_{E} \mu_i(E) I_{0j} \exp [-L_1(y)\mu_1(E) - L_2(y)\mu_2(E)]$$

$$+ \beta \left( \log \sum_x h(y|x) \hat{c}_i(x) \right).$$  \hspace{1cm} (3.7)

The corresponding Hessian matrix is $\nabla^2(y) = [\nabla^2_{11}(y), \nabla^2_{12}(y), \nabla^2_{21}(y), \nabla^2_{22}(y)]$, where

$$\nabla^2_{ii}(y) = \sum_{j} \sum_{E} \mu^2_i(E) I_{0j} \exp [-L_1(y)\mu_1(E) - L_2(y)\mu_2(E)] + \frac{\beta}{L_i(y)},$$

$$\nabla^2_{12}(y) = \nabla^2_{21}(y) = \sum_{j} \sum_{E} \mu_1(E)\mu_2(E) I_{0j} \exp [-L_1(y)\mu_1(E) - L_2(y)\mu_2(E)].$$  \hspace{1cm} (3.8)

The pseudocode for the LIAM algorithm is as follows:

A. Set $k = 0$. Select initializations for $\hat{c}_i^{(0)}(x), \hat{L}_i^{(0)}(y)$.

B. Update $\hat{p}_j^{(k)}(y, E)$ according to

$$\hat{f}_j^{(k)}(y, E) = I_{0j}(y, E) \exp \left[ -\hat{L}_1^{(k)}(y)\mu_1(E) - \hat{L}_2^{(k)}(y)\mu_2(E) \right]$$

$$, \hat{p}_j^{(k)}(y, E) = d_j(y) \frac{\hat{f}_j^{(k)}(y, E)}{\sum_{y'} \hat{f}_j^{(k)}(y', E')}.$$  

C. Update $\hat{L}_i^{(k+1)}(y)$ using Newton’s method.

a. Set $m = 0$. Let $\hat{L}_i^{(m=0)} = \hat{L}_i^{(k)}(y)$.  

44
b. $\hat{L}_{i,NEWTON}^{(m+1)}(y) = \hat{L}_{i,NEWTON}^{(m)}(y) - [\nabla^2(y)]^{-1}\nabla(y)$, where $\nabla^2(y), \nabla(y)$ are evaluated at $\hat{L}_{i,NEWTON}^{(m)}(y)$.

c. Iterate until convergence to obtain $\hat{L}_{i}^{(k+1)}(y) = \hat{L}_{i,NEWTON}^{(m+1)}(y)$.

D. Update $\hat{c}_i^{(k+1)}(x)$ using the EM algorithm,

a. Set $n = 0$. Let $\hat{c}_i^{(n=0)}(y) = \hat{c}_i^{(k)}(x)$.

b. $\hat{c}_i^{(n+1)}_{EM}(x) = \frac{\hat{c}_i^{(n)}_{EM}(x)}{\sum_y h(y|x) \sum_y h(y|x) \frac{\hat{L}_i^{(k+1)}(y)}{\sum_x h(y|x) \hat{c}_i^{(n)}_{EM}(x)}}$.

c. Iterate until convergence to obtain $\hat{c}_i^{(k+1)}(x) = \hat{c}_i^{(n+1)}_{EM}(x)$.

E. Iterate until convergence.

Note that in the above procedure, the choice of $\beta$ is not discussed. It can be incrementally changed during iterations, starting with zero.

### 3.3 Regularized LIAM Algorithm

As with the DE-AM algorithm, desirable smooth images can be achieved using regularization. In the setup of the LIAM algorithm, we cannot apply regularization to the estimated line integrals directly since we don’t have the smoothness prior regarding the line integrals for neighboring measurements. Instead, we need to apply regularization to the the EM algorithm which recovers the component images.

We use the same penalty function used by the regularized DE-AM described in Chapter 2. We also borrow the same idea to decouple the penalty function (see Equation (2.7)), and the resulting objective function (in the EM algorithm step) will be strictly convex and Newton’s method can be applied to solve for the component images.

In summary, the regularized LIAM algorithm differs from the unregularized LIAM only in the EM step which solves for the component images, where we use Newton’s method to incorporate the penalty rather than a regular EM update step. The total objective function with regularization included is
\[
\min_{L_i, c_i} \sum_{j=1}^{2} I(d_j||F_j) + \beta \sum_{i=1}^{2} [I(L_i||Hc_i) + \lambda R(c_i)]
\]

\[
= \min_{L_i, c_i} \sum_{j} \sum_{y} \left[ d_j(y) \log \frac{d_j(y)}{F_j(L_1(y), L_2(y))} - d_j(y) + F_j(L_1(y), L_2(y)) \right] + \beta \sum_{i} \sum_{y} \left[ L_i(y) \log \frac{L_i(y)}{\sum_x h(y|x)c_i(x)} - L_i(y) + \sum_x h(y|x)c_i(x) \right] + \beta \lambda \sum_{i} \sum_{x} \sum_{k \in N_x} w_{x,k} \psi'(c_i(x) - c_i(k)),
\]

where \(\psi(x)\) is the same edge-preserving penalty function as in Chapter 2.

In the implementation, \(\beta\) and \(\lambda\) can be selected jointly to achieve the desired image quality. When \(\beta\) approaches zero, the term associated with the last two double sums are negligible, and the updates of the line integrals do not depend on the reconstructed images. After the line integrals are updated, by using some value of \(\lambda\), we can recover the images.

### 3.4 Simulation Results

In this section, the simulated image size is 64×64 with pixel size 1mm×1mm. \(I_0 = 10000\) which corresponds to the mean number of unattenuated photons per detector. The two component materials used are polystyrene \((c_1)\) and calcium chloride \((c_2)\). The attenuation coefficient spectra for the two components are shown in Figure 2.3b. The initial conditions are random images with pixel intensities between 0 and 1.

A mini CT scanner geometry is used for generating simulated data. There are 360 source angles and 92 detectors in this geometry. Figure 2.3a shows the two incident energy spectra corresponding to tube voltages of 80 kVp and 140 kVp. The phantom consists of a cylindrical Lucite core in air. The core has four openings for housing different materials. We simulate samples of muscle, Teflon and substance X with the two component materials (polystyrene
and calcium chloride) in the proportions listed in Table 3.1. The true component images for polystyrene ($c_1(x)$) and calcium chloride ($c_2(x)$) are given in Figure 3.1.

Table 3.1: Component coefficients of the phantom for mini CT geometry

<table>
<thead>
<tr>
<th>Substance</th>
<th>Polystyrene $c_1(x)$</th>
<th>Calcium Chloride $c_2(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lucite</td>
<td>1.14000</td>
<td>0.05834</td>
</tr>
<tr>
<td>Muscle</td>
<td>0.93995</td>
<td>0.13904</td>
</tr>
<tr>
<td>Teflon</td>
<td>1.41940</td>
<td>0.48779</td>
</tr>
<tr>
<td>Substance X</td>
<td>0.03000</td>
<td>2.86130</td>
</tr>
</tbody>
</table>

3.4.1 Reconstruction with $\beta = 0$ and Fixed $\lambda$

We first test the unregularized version of the LIAM algorithm. We compare the reconstruction results of the unregularized LIAM after 200 full iterations (updating $p_j, L_i$ and $c_i$) and the reconstruction results of the unregularized DE-AM after 32000 iterations without OS, using noiseless data and Poisson noisy data. Note that in the reconstruction using the LIAM, we set $\beta = 0$ (it corresponds to the case when $\beta$ is closer to zero and the penalty term is negligible compared to data fitting term, so the updating for line integrals do not depend on reconstructed images) for all iterations. The reason for choosing the reconstructions corresponding to certain number of iterations (200 for the LIAM and 32000 for the DE-AM) is that if we compare the $I$-divergence terms which are the data fitting terms inside the
Figure 3.2: Noiseless data reconstruction for (a) polystyrene \( c_1(x) \) by using the unregularized LIAM algorithm, (b) calcium chloride \( c_2(x) \) by using the unregularized LIAM algorithm, (c) polystyrene \( c_1(x) \) by using the unregularized DE-AM algorithm and (d) calcium chloride \( c_2(x) \) by using the unregularized DE-AM algorithm.

objective function, the two results have similar \( I \)-divergence values. The results are shown in Figure 3.2 and Figure 3.3. We see that although the images from noisy data are grainier than noiseless data reconstructions, due to the presence of noise, results from the unregularized LIAM algorithm can achieve quite accurate estimation of the component images for every region of interest on average. However, the unregularized DE-AM algorithm seems to have difficulty separating the two components due to the high correlation between the two component images. Moreover, the LIAM requires only 200 iterations whereas the DE-AM requires 32000 iterations.
Figure 3.3: Noisy data reconstruction for (a) polystyrene \((c_1(x))\) by using the unregularized LIAM algorithm, (b) calcium chloride \((c_2(x))\) by using the unregularized LIAM algorithm, (c) polystyrene \((c_1(x))\) by using the unregularized DE-AM algorithm and (d) calcium chloride \((c_2(x))\) by using the unregularized DE-AM algorithm.
To give a quantitative assessment of the reconstructions, the relative absolute estimation errors of each component image are shown in Table 3.2. In general, the unregularized LIAM algorithm can achieve much lower relative absolute estimation errors compared to the unregularized DE-AM algorithm.

Table 3.2: Relative absolute estimation errors (%) of $c_1(x)$ and $c_2(x)$

<table>
<thead>
<tr>
<th>Material</th>
<th>Muscle(Top)</th>
<th>Muscle(Right)</th>
<th>Teflon</th>
<th>X</th>
<th>Lucite</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Noiseless Data</strong></td>
<td>LIAM</td>
<td>0.84</td>
<td>0.66</td>
<td>0.28</td>
<td>31.1</td>
</tr>
<tr>
<td></td>
<td>DE-AM</td>
<td>9.36</td>
<td>7.63</td>
<td>8.61</td>
<td>111.4</td>
</tr>
<tr>
<td></td>
<td>LIAM</td>
<td>14.7</td>
<td>16.4</td>
<td>12.9</td>
<td>160</td>
</tr>
<tr>
<td></td>
<td>DE-AM</td>
<td>39.6</td>
<td>36.9</td>
<td>14.2</td>
<td>3.92</td>
</tr>
<tr>
<td><strong>Noisy Data</strong></td>
<td>LIAM</td>
<td>21.4</td>
<td>19.3</td>
<td>17.2</td>
<td>3752</td>
</tr>
<tr>
<td></td>
<td>DE-AM</td>
<td>60.9</td>
<td>55.8</td>
<td>22.8</td>
<td>21.0</td>
</tr>
</tbody>
</table>

Similarly, relative absolute errors can be computed for synthesized attenuation coefficients (mm$^{-1}$) over an energy range. Figure 3.4 shows the relative absolute errors as a function of energy for the unregularized LIAM algorithm and the unregularized DE-AM algorithm. We can see that the unregularized LIAM can achieve lower relative absolute errors at low energy levels at which few photons are detected. Most of the materials studied behave similarly in that they have high errors at low energy levels, the errors drop down to a minimum as the energy increases and go up a little bit at higher energy levels. However, for substance X, for the unregularized LIAM algorithm with noiseless data, the error goes up monotonically as the energy level increases and for the unregularized LIAM algorithm with noisy data, the error goes down monotonically as the energy level increases.

We can also plot the evolution of $I$-divergence (between the data and the mean represented by the reconstructed images) in order to compare the convergence of the unregularized LIAM algorithm to the unregularized DE-AM algorithm, as seen in Figure 3.5. We see that the unregularized LIAM algorithm converges much faster than the unregularized DE-AM algorithm, in terms of both iteration and time. While this is not a perfect comparison since the $I$-divergence in the unregularized LIAM algorithm is computed in terms of component images, which is not the objective function LIAM algorithm tries to minimize, Figure 3.5 does show a much faster convergence. Note that for the same reason, $I$-divergence does not decrease monotonically. Moreover, for the unregularized LIAM algorithm (when $\beta = 0$),
Figure 3.4: Relative absolute errors of synthesized attenuation coefficients by using (a) the unregularized LIAM with noiseless data, (b) the unregularized DE-AM with noiseless data, (c) the unregularized LIAM with noisy data and (d) the unregularized DE-AM with noisy data.
since updating line integrals does not depend on the update of images, the running time of
the unregularized LIAM can be reduced significantly. In the current simulation experiment,
the subsequent EM update takes about 2000 iterations to converge.

Next, we compare the regularized LIAM algorithm and the regularized DE-AM algorithm
as well as the effect of regularization parameters. As in the unregularized case comparison,
reconstruction results of the regularized LIAM after 200 full iterations (updating $p_j$, $L_i$, and
$c_i$) and the reconstruction results of the regularized DE-AM after 32000 iterations without
OS are displayed, using the same Poisson noisy data.

Figure 3.6 and Figure 3.7 give the reconstruction results from the regularized LIAM algorithm
when $\delta = 50$ and $\delta = 500$, respectively, where $\delta$ inverse corresponds to the penalty function
transition from the quadratic region to the linear region. Clearly, with regularization weight
$\lambda$ increasing, the resulting images get smoother. However, when $\lambda$ is too large, for example
in the cases when $\delta = 50, \lambda = 500$ and $\delta = 500, \lambda = 5000$, the images are severely blurred,
and edges are lost. By jointly comparing the reconstructions under different $\lambda$’s and $\delta$’s,
$\delta = 500, \lambda = 500$ gives the best quantitative reconstruction performance in terms of image
smoothness and edge preservation.

Figure 3.8 and Figure 3.9 show the reconstruction results $c_1(x)$ and $c_2(x)$, respectively, from
the regularized DE-AM algorithm when $\delta = 500$. Clearly, as the regularization weight $\lambda$
increases, the resulting images become smoother. However, if we compare the results from
the regularized DE-AM and the regularized LIAM (not shown), the regularized LIAM can
achieve much better performance in terms of accurately estimating the component images
for every region of interest on average and image smoothness. This is consistent with the
comparison between the unregularized DE-AM algorithm and the unregularized LIAM al-
gorithm shown previously. Note that different windows were used to compare reconstructed
$c_2(x)$ images between the two algorithms because regularized DE-AM is far from convergence
and the estimation error is quite large.

In order to give a quantitative assessment of the performances of the reconstructions, profile
plots are given in Figure 3.10 to show the reconstructions through a horizontal line cutting
through Substance X. According to the profile plots, in general, the LIAM achieves much
better performance than the DE-AM, in terms of accurately estimating the component image
on average and image smoothness. The DE-AM images have severe overshooting and
Figure 3.5: (a) $I$-divergence vs. iteration and (b) $I$-divergence vs. time for the unregularized LIAM algorithm and the unregularized DE-AM algorithm.
Figure 3.6: Reconstructed polystyrene ($c_1(x)$) and calcium chloride ($c_2(x)$) images using the regularized LIAM algorithm ($\delta = 50$)
Figure 3.7: Reconstructed polystyrene \( c_1(x) \) and calcium chloride \( c_2(x) \) images using the regularized LIAM algorithm (\( \delta = 500 \))
Figure 3.8: Reconstructed polystyrene \((c_1(x))\) images using the regularized DE-AM algorithm \((\delta = 500)\)
Figure 3.9: Reconstructed calcium chloride ($c_2(x)$) images using the regularized DE-AM algorithm ($\delta = 500$)
undershooting problems for the two component images, resulting from the inability to separate the two components. As for the choice of regularization parameters, for the regularized LIAM algorithm, $\delta = 500, \lambda = 500$ appears to give the best performance, which is consistent with the observation from the reconstructed images.

### 3.4.2 Reconstruction with $\beta = 0$ and Dynamic $\lambda$

In order to adaptively select $\lambda$ in the regularized LIAM algorithm, we experimentally change $\lambda$ during the iterations and observe the effect of different $\lambda$ values.
Figure 3.10: Profile plots corresponding to a horizontal line through the Substance X
Here we consider only the regularized LIAM algorithm with $\delta = 50$ and $\delta = 500$. Again, we use the estimated line integrals obtained after 200 iterations of the LIAM algorithm with $\beta = 0$.

We can dynamically change $\lambda$ in either ascending order or descending order. We refer to the sequence of $\lambda$ values for a fixed $\delta$ as a “train.” Therefore, there are a total of four cases under consideration: for $\delta = 50$, the $\lambda$ train is selected as $[1, 2, 4, 8, 16, 32, 64, 128, 256, 512]$; for $\delta = 50$, the $\lambda$ train is selected as $[512, 256, 128, 64, 32, 16, 8, 4, 2, 1]$; for $\delta = 500$, the $\lambda$ train is selected as $[10, 20, 40, 80, 160, 320, 640, 1280, 2560, 5120]$ and for $\delta = 500$, the $\lambda$ train is selected as $[5120, 2560, 1280, 640, 320, 160, 80, 40, 20, 10]$. The reason for those choices of $\lambda$ trains of different $\delta$’s is that in the linear region (both $\delta = 50$ and $\delta = 500$ are in the linear region), the regularization term can be approximated as $\lambda/\delta \times t$, where $t$ is the neighborhood pixel difference, and maintaining the ratio of $\lambda$ and $\delta$ will give a relatively fair comparison between the results corresponding to different choices of $\delta$’s.

The experimented procedure is that we initialize $c_1(x)$ and $c_2(x)$ by the same random image for all four cases. For each case, we sequentially use the $\lambda$’s in the corresponding $\lambda$ train, and fix each $\lambda$ for 100 iterations. Reconstructed images are obtained and saved after every 100 iterations.

Figure 3.11 shows the $I$-divergence term and penalty term during the iterations for all four cases. The obvious stair-shaped jumps or declines correspond to the changes of the $\lambda$ value. For the cases with ascending choice of $\lambda$, penalty values monotonically decrease and for descending order, the penalty value monotonically decreases within the iterations corresponding to a fixed $\lambda$ choice; as $\lambda$ goes down, the penalty values gradually go up. We hope to create a flow that can achieve equilibrium for every choice of $\lambda$, and eventually achieve the global minimum.

In order to better visualize the intermediate results during the dynamic sweeping of $\lambda$, Figure 3.12 through Figure 3.15 show the reconstructed images after every 100 iterations, i.e. before the change of $\lambda$. According to the images, for both $\delta = 50$ and $\delta = 500$ cases, sweeping $\lambda$ in ascending order results in smoother images along the iterations and sweeping $\lambda$ in descending order results in rougher images along the iterations. Since Figure 3.11 shows that using 100 iterations for every choice of $\lambda$ will achieve convergence for that particular $\lambda$, it might be
Figure 3.11: Total objective function, $I$-divergence and penalty along iterations

(a) $\delta = 50$, ascending

(b) $\delta = 50$, descending

(c) $\delta = 500$, ascending

(d) $\delta = 500$, descending
possible for us to dynamically select $\lambda$ along the iterations such that equilibrium is achieved for every choice of $\lambda$.

### 3.4.3 Reconstruction with Dynamic $\beta$ and $\lambda$

The introduction of $\beta$ and $\lambda$ in the LIAM algorithm offers flexibility to control the reconstruction performance as well as the convergence rate. However, at the same time, it adds complexity in terms of choosing the correct parameters. We inherit the same spirit as in Chapter 2 and try to explore the parameter space to achieve the best image smoothness while maintaining high estimation accuracy. From the previous section, we can see that by running the unregularized LIAM with $\beta = 0$, the reconstructed images can achieve high average estimation accuracy, but are very grainy. In order to improve image smoothness, we will use nonzero $\beta$ to incorporate the geometry constraint, as well as regularization to directly add a neighborhood penalty. We compare the reconstruction results of the regularized LIAM after 200 full iterations (updating $p_j, L_i$ and $c_i$) with different $\lambda$ values, using the same Poisson noisy data as before. Note that in the reconstructions, we set $\beta = 0$ for the first 100 iterations and $\beta = 1000$ for the second 100 iterations. For the first 100 iterations, the unregularized EM algorithm is used for reconstructing images while for the second 100 iterations, a trust region Newton’s method is used. According to the results shown in Figures 3.16 and 3.17, $\lambda = 0.5$ and $\lambda = 5$ do not give a strong enough neighborhood penalty while $\lambda = 500$ is too strong and washes out some regions. Among these $\lambda$ choices, $\lambda = 50$ performs best, even though there are isolated speckles for $c_2(x)$. It is worth mentioning that the choice of $\beta$ and $\lambda$ should be jointly considered, since according to Equation 3.9, the neighborhood penalty weight is $\beta \lambda$ in the overall objective function.

### 3.5 Physical Phantom Experiment Results

In this section, real data of a cylinder phantom were collected (data were collected by Dong Han and Joshua D. Evans at Virginia Commonwealth University) from the Philips Brilliance scanner with 816 detectors, 1320 sources and collimation 4 by 0.75mm. Incident spectra are 90 kVp and 140 kVp as shown in Figure 2.9. Only the first row of data were used for
(a) $\delta = 50$, $\lambda = 1$, ascending
(b) $\delta = 50$, $\lambda = 1$, ascending
(c) $\delta = 50$, $\lambda = 2$, ascending
(d) $\delta = 50$, $\lambda = 2$, ascending
(e) $\delta = 50$, $\lambda = 4$, ascending
(f) $\delta = 50$, $\lambda = 4$, ascending
(g) $\delta = 50, \lambda = 8$, ascending

(h) $\delta = 50, \lambda = 8$, ascending

(i) $\delta = 50, \lambda = 16$, ascending

(j) $\delta = 50, \lambda = 16$, ascending

(k) $\delta = 50, \lambda = 32$, ascending

(l) $\delta = 50, \lambda = 32$, ascending
(m) $\delta = 50, \lambda = 64$, ascending  

(n) $\delta = 50, \lambda = 64$, ascending

(o) $\delta = 50, \lambda = 128$, ascending  

(p) $\delta = 50, \lambda = 128$, ascending

(q) $\delta = 50, \lambda = 256$, ascending  

(r) $\delta = 50, \lambda = 256$, ascending
Figure 3.12: Reconstructed polystyrene ($c_1(x)$, left columns) and calcium chloride ($c_2(x)$, right columns) images using the regularized LIAM algorithm by sweeping $\lambda$ in ascending order ($\delta = 50$) with display window [0.8 1.2] for $c_1(x)$ and [0,0.15] for $c_2(x)$.

reconstruction. The physical phantom is a lucite cylinder with four holes filled with ethanol, Teflon, polystyrene and PMMA. The image size is 610×610 with pixel size 1mm×1mm. The two component materials used are polystyrene ($c_1$) and calcium chloride ($c_2$). The attenuation coefficients spectra for the two components are shown in Figure 2.3b.

Since we do not have the true $c_1$ and $c_2$ for the four sample materials inside the physical phantom, we can compute the theoretical truth based on the linear attenuation map from NIST. Table 3.3 gives the true $c_1$ and $c_2$ for the four sample materials, computed with least-square fitting. NIST data corresponding to seven energy levels were used, 20, 30, 40, 50, 60, 80 and 100 keV. The nonzero coefficient (0.0020) of the calcium chloride component for the polystyrene material is due to the correlation between the attenuation coefficients of the two component materials (polystyrene and calcium chloride).

Table 3.3: Component coefficients for the physical cylinder phantom

<table>
<thead>
<tr>
<th>Substance</th>
<th>Polystyrene $c_1(x)$</th>
<th>Calcium Chloride $c_2(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethanol</td>
<td>0.7392</td>
<td>0.0327</td>
</tr>
<tr>
<td>Teflon</td>
<td>1.3055</td>
<td>0.4253</td>
</tr>
<tr>
<td>Polystyrene</td>
<td>0.9899</td>
<td>0.0020</td>
</tr>
<tr>
<td>PMMA</td>
<td>1.0532</td>
<td>0.0575</td>
</tr>
</tbody>
</table>
(a) $\delta = 50$, $\lambda = 512$, descending

(b) $\delta = 50$, $\lambda = 512$, descending

(c) $\delta = 50$, $\lambda = 256$, descending

(d) $\delta = 50$, $\lambda = 256$, descending

(e) $\delta = 50$, $\lambda = 128$, descending

(f) $\delta = 50$, $\lambda = 128$, descending
δ = 50, λ = 64, descending

δ = 50, λ = 64, descending

δ = 50, λ = 32, descending

δ = 50, λ = 32, descending

δ = 50, λ = 16, descending

δ = 50, λ = 16, descending
\(\delta = 50, \lambda = 8\), descending

\(\delta = 50, \lambda = 4\), descending

\(\delta = 50, \lambda = 2\), descending
Figure 3.13: Reconstructed polystyrene ($c_1(x)$, left columns) and calcium chloride ($c_2(x)$, right columns) images using the regularized LIAM algorithm by sweeping $\lambda$ in descending order ($\delta = 50$) with display window $[0.8 \ 1.2]$ for $c_1(x)$ and $[0,0.15]$ for $c_2(x)$

3.5.1 Reconstruction with $\beta = 0$ and Fixed $\lambda$

We first test the unregularized version of the LIAM algorithm on this set of real data. We compare the reconstruction results of the unregularized LIAM after 200 full iterations (updating $p_j$, $L_i$ and $c_i$) and the reconstruction results of the unregularized DE-AM after 1000 iterations with 33 OS (which is equivalent to 33000 iterations without OS). The initial conditions for both of the algorithms are images with all 1’s.

Table 3.4 compares some statistics for the $c_1$ and $c_2$ reconstructed by the unregularized LIAM and the unregularized DE-AM. The error percentage is calculated as the average of the difference ratio in percentage, which is the difference between the reconstructed value and true value divided by the true value. Variance is calculated as the sample variance inside every sample material region. SNR is calculated as the ratio of the mean square and variance inside the sample region. From this table, it can be seen that the unregularized LIAM gives much more accurate estimation in an average sense, but suffers from high variance and low SNR. On the other hand, the unregularized DE-AM gives low variance and high SNR, but less accurate estimation. The huge values in error percentage for polystyrene material are due to the corresponding calculated very small $c_2$ true value. Moreover, the order of magnitude
(a) $\delta = 500, \lambda = 10$, ascending

(b) $\delta = 500, \lambda = 10$, ascending

(c) $\delta = 500, \lambda = 20$, ascending

(d) $\delta = 500, \lambda = 20$, ascending

(e) $\delta = 500, \lambda = 40$, ascending

(f) $\delta = 500, \lambda = 40$, ascending
(g) $\delta = 500, \lambda = 80$, ascending

(h) $\delta = 500, \lambda = 80$, ascending

(i) $\delta = 500, \lambda = 160$, ascending

(j) $\delta = 500, \lambda = 160$, ascending

(k) $\delta = 500, \lambda = 320$, ascending

(l) $\delta = 500, \lambda = 320$, ascending
Figure 3.14: Reconstructed polystyrene \( (c_1(x), \text{left columns}) \) and calcium chloride \( (c_2(x), \text{right columns}) \) images using the regularized LIAM algorithm by sweeping \( \lambda \) in ascending order \( (\delta = 500) \) with display window [0.8 1.2] for \( c_1(x) \) and [0,0.15] for \( c_2(x) \).

SNR difference between \( c_1 \) and \( c_2 \) is due to the absolute magnitude difference between \( c_1 \) and \( c_2 \).

Next we plot the relative estimation error, which is calculated similarly to the error percentage, but for linear attenuation coefficients rather than for \( c_1 \) and \( c_2 \). From Figure 3.18, it can be seen that the unregularized DE-AM gives much higher errors at low energy levels and that the errors shrink down monotonically with increasing energy levels, within \( \pm 5\% \) for energy levels higher than 50 keV. While for the unregularized LIAM, the error is almost bounded by \( \pm 6\% \) with energy levels ranging from 20 keV to 100 keV. The behavior of the error for the unregularized LIAM is not monotonic with energy for every material.

The image reconstruction results are shown in Figure 3.19, from which we can see that although the images are all grainy due to the existence of noise, results from the unregularized LIAM algorithm can achieve quite accurate estimation of the component images for every region of interest on average. The unregularized DE-AM algorithm has difficulty separating the two components due to the high correlation between them. Moreover, the LIAM is superior to the DE-AM in terms of the number of iterations (200 iterations vs. 1000 iterations with 33 OS). These observations are consistent with those from the simulated noisy data case.
(a) $\delta = 500, \lambda = 5120$, descending

(b) $\delta = 500, \lambda = 5120$, descending

(c) $\delta = 500, \lambda = 2560$, descending

(d) $\delta = 500, \lambda = 2560$, descending

(e) $\delta = 500, \lambda = 1280$, descending

(f) $\delta = 500, \lambda = 1280$, descending
(g) $\delta = 500, \lambda = 640$, descending

(h) $\delta = 500, \lambda = 640$, descending

(i) $\delta = 500, \lambda = 320$, descending

(j) $\delta = 500, \lambda = 320$, descending

(k) $\delta = 500, \lambda = 160$, descending

(l) $\delta = 500, \lambda = 160$, descending
(m) \( \delta = 500, \lambda = 80 \), descending
(n) \( \delta = 500, \lambda = 80 \), descending
(o) \( \delta = 500, \lambda = 40 \), descending
(p) \( \delta = 500, \lambda = 40 \), descending
(q) \( \delta = 500, \lambda = 20 \), descending
(r) \( \delta = 500, \lambda = 20 \), descending
(s) $\delta = 500$, $\lambda = 10$, descending  
(t) $\delta = 500$, $\lambda = 10$, descending

Figure 3.15: Reconstructed polystyrene ($c_1(x)$, left columns) and calcium chloride ($c_2(x)$, right columns) images using the regularized LIAM algorithm by sweeping $\lambda$ in descending order ($\delta = 500$) with display window $[0.8, 1.2]$ for $c_1(x)$ and $[0, 0.15]$ for $c_2(x)$

Table 3.4: Comparison between the unregularized LIAM ($\beta = 0$) and DE-AM algorithms

<table>
<thead>
<tr>
<th>Material</th>
<th>Ethanol</th>
<th>Teflon</th>
<th>Polystyrene</th>
<th>PMMA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unregularized LIAM Error %</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c_1$</td>
<td>-0.8867</td>
<td>3.3569</td>
<td>-7.4586</td>
<td>-4.3595</td>
</tr>
<tr>
<td>$c_2$</td>
<td>5.6147</td>
<td>-1.9891</td>
<td>915.0322</td>
<td>6.7525</td>
</tr>
<tr>
<td>Unregularized DE-AM Error %</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c_1$</td>
<td>4.5249</td>
<td>-3.6676</td>
<td>-9.8442</td>
<td>-9.2281</td>
</tr>
<tr>
<td>$c_2$</td>
<td>-50.7496</td>
<td>9.7982</td>
<td>3180.9185</td>
<td>114.0799</td>
</tr>
<tr>
<td>Unregularized LIAM Variance</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c_1$</td>
<td>0.0179</td>
<td>0.0415</td>
<td>0.0327</td>
<td>0.0320</td>
</tr>
<tr>
<td>$c_2$</td>
<td>0.0007</td>
<td>0.0285</td>
<td>0.0003</td>
<td>0.0022</td>
</tr>
<tr>
<td>Unregularized DE-AM Variance</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c_1$</td>
<td>0.0002</td>
<td>0.0004</td>
<td>0.0002</td>
<td>0.0003</td>
</tr>
<tr>
<td>$c_2$</td>
<td>0.0001</td>
<td>0.0004</td>
<td>0.0002</td>
<td>0.0002</td>
</tr>
<tr>
<td>Unregularized LIAM SNR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c_1$</td>
<td>29.9950</td>
<td>43.9048</td>
<td>25.6627</td>
<td>31.7510</td>
</tr>
<tr>
<td>$c_2$</td>
<td>1.6233</td>
<td>6.1011</td>
<td>1.6055</td>
<td>1.7088</td>
</tr>
<tr>
<td>Unregularized DE-AM SNR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c_1$</td>
<td>2532.4379</td>
<td>4104.3425</td>
<td>3219.2553</td>
<td>3363.1770</td>
</tr>
<tr>
<td>$c_2$</td>
<td>1.8419</td>
<td>607.8824</td>
<td>17.4691</td>
<td>63.4820</td>
</tr>
</tbody>
</table>

78
Figure 3.16: Noisy data reconstruction for polystyrene ($c_1(x)$)
Figure 3.17: Noisy data reconstruction for calcium chloride ($c_2(x)$)

Figure 3.18: Relative estimation errors of linear attenuation coefficients obtained by (a) the unregularized LIAM with $\beta = 0$ and (b) the unregularized DE-AM
Figure 3.19: Real data reconstructions for (a) $c_1(x)$ from the unregularized LIAM with $\beta = 0$, (b) $c_2(x)$ by the unregularized LIAM with $\beta = 0$, (c) $c_1(x)$ from the unregularized DE-AM with 33 OS and (d) $c_2(x)$ from the unregularized DE-AM with 33 OS
3.5.2 Reconstruction with Dynamic $\beta$ and $\lambda$

In the following experiments, we compared the reconstruction results from the unregularized/regularized LIAM algorithm and the regularized DE-AM algorithm. For the regularized DE-AM algorithm, images with all 1’s were used as the initial condition. Reconstructed images were obtained from 1000 iterations with 33 OS and regularization parameters $\delta = 500, \lambda = 100$. For both the unregularized and the regularized LIAM algorithm, images for the initial condition were obtained by using the unregularized LIAM algorithm with $\beta = 0$ for 30 iterations. The unregularized LIAM algorithm with $\beta = 1000$ was run for 100 iterations to obtain the reconstructed images. The regularized LIAM algorithm with $\beta = 1000$ was also run for 100 iterations. The chosen regularization parameters were $\delta = 500, \lambda = 3.3$, since the equivalent neighborhood penalty weight for the regularized LIAM algorithm is $\beta\lambda = 3300$ and the equivalent $\lambda$ value for the regularized DE-AM algorithm without OS is $100 \times 33 = 3300$. Figure 3.20 show the reconstructed $c_1(x)$ and $c_2(x)$ images for the three cases described above. The corresponding relative estimation errors for the synthesized linear attenuation coefficients were computed and shown in Figure 3.21. By jointly looking at Figure 3.19(c)(d) and Figure 3.20(e)(f), the results from the regularized DE-AM algorithm do not differ significantly from those obtained from the unregularized DE-AM algorithm. The introduction of neighborhood penalty in the regularized LIAM algorithm gives some level of smoothness in the reconstructed images. However, with the equivalent choice of the regularization parameters, the results from the regularized LIAM algorithm did not achieve the same level of smoothness as the regularized DE-AM algorithm. When we look at the relative estimation errors for the synthesized linear attenuation coefficients, both the unregularized and the regularized LIAM algorithm with $\beta = 1000$ have a big jump in the error for the polystyrene estimate at 20 keV, compared to the unregularized LIAM algorithm with $\beta = 0$ (See Figure 3.18(a) and Figure 3.21(a)(b)). The regularized LIAM algorithm produces images with improved smoothness but less estimation error on average. Generally speaking, the LIAM algorithm achieves much lower estimation errors at low energy levels (around 20 keV) compared to the DE-AM algorithm.
Figure 3.20: Real data reconstructions for (a) \( c_1(x) \) from the unregularized LIAM with \( \beta = 1000 \), (b) \( c_2(x) \) by the unregularized LIAM with \( \beta = 1000 \), (c) \( c_1(x) \) from the regularized LIAM with \( \beta = 1000, \lambda = 3.3 \), (d) \( c_2(x) \) from the regularized LIAM with \( \beta = 1000, \lambda = 3.3 \), (e) \( c_1(x) \) from the regularized DE-AM with \( \lambda = 100, 33 \) OS and (f) \( c_2(x) \) from the regularized DE-AM with \( \lambda = 100, 33 \) OS.
Figure 3.21: Relative estimation errors of linear attenuation coefficients obtained by (a) the unregularized LIAM with $\beta = 1000$, (b) the regularized LIAM with $\beta = 1000, \lambda = 3.3$ and (c) the regularized DE-AM with $\lambda = 100$ and 33 OS.
3.6 Conclusions and Contributions

A new algorithm, LIAM, is proposed which estimates the line integrals of the two component images first, and then estimates the component images based on the estimated line integrals. The advantage of this two-step algorithm is that the convergence is much faster than the DE-AM algorithm and the resulting coefficient images can accurately estimate the linear attenuation coefficients even in the presence of very high density materials (higher than bone).

Simulated data reconstructions show good image smoothness as well as accurate estimation, with proper choices for the penalty between the estimated line integrals and the forward projections of the component images, and the regularization parameters for the neighborhood penalties. For real data from the Philips Brilliance scanner, accurate average estimation was achieved.

For future work, dynamic choice of the penalty for estimated line integrals and the forward projections of the component images as well as neighborhood penalty should be explored, in order to achieve an equilibrium solution at every choice stage. This procedure can minimize the bias introduced by the regularization.

The motivation for the LIAM algorithm emerged during discussions with Dr. Joseph A. O’Sullivan. I introduced the penalty that penalizes the differences between the estimated line integrals and the forward projections of the component images. I implemented the LIAM algorithm on the simulated data and the real data with different regularization parameters.
Chapter 4

Reconstructing Image Differences from Tomographic Poisson Data

4.1 Abstract

Given two measurements of an image and a modified version of the image, we seek reconstructions of both the original image and the difference of the images [62]. The data are assumed to be Poisson, with known nonnegative forward operator and nonnegative images. A penalized likelihood is minimized with the penalty equal to the sum of the absolute differences between the images.

Alternating minimization algorithms have been studied by many researchers ever since the fundamental work by Csiszar and Tusnady [63]. These algorithms are widely used in information theory [63][64][65][66][67] and in image estimation [1][68][69]. An alternating minimization algorithm is developed by reformulating the penalized maximum-likelihood problem as a double minimization of $I$-divergence plus the penalty. This polyenergetic algorithm guarantees monotonic decrease in the objective function for each iteration. Simulations with random images and tomographic data are presented to demonstrate properties of the algorithm. Convergence properties of the algorithm are studied both theoretically and in simulations. In addition, an extension of application to 4D CT with simulated tomographic data is also presented. Moreover, an approach to 4D PET is described.
4.2 Introduction

In essentially all imaging problems there are calibration measurements. The closer the calibration measurements are to later measurements, the greater their utility. In some cases, there are two measurements that are closely related, with only small changes between them. We consider general and tomographic measurements of this type, with Poisson data, where the goal is to simultaneously reconstruct the first image and the change between the images, assuming the difference image is sparse.

The general linear regression problem has attracted considerable research interest ever since it was first introduced. Least absolute shrinkage and selection operator (LASSO) casts the problem as a minimization of mean-squared error (MSE) under an $L_1$ norm constraint. By shrinking some coefficients and setting others to 0, LASSO sacrifices a little bias to reduce the variance of the predicted values and therefore may improve the overall prediction accuracy [70].

Compressed sensing (CS) exploits known (sparsity) constraints on signal structure in order to reduce the sampling rate, and subsequent demands on storage and digital signal processing [71]. The goal of CS is to have as few measurements as possible, but still preserve the significant nonzero signal components.

For both the general linear regression problem and the CS problem, nonnegativity constraints should be imposed in the context of image recovery; that is, both the measurements and the input images should be nonnegative. We consider a Poisson noise model on the measurements, which is prevalent in medical imaging, such as positron emission tomography (PET). Under such circumstances, LASSO and related CS algorithms, which are essentially based on Gaussian models, are inappropriate in two ways. First, most papers formulate the objective function as a combination of an $L_2$ norm squared data fitting term and an $L_1$ norm regularization term corresponding to the sparsity regularizer. However, due to the existence of Poisson noise, the variance of the noisy observation grows proportionally to the signal strength, and the data fitting term will result in significant overfitting in high-intensity regions and oversmoothing in low-intensity regions [72]. Second, few papers include the nonnegativity constraint imposed naturally by the image system. Willett, et al. [72] proposed a regularized Poisson log-likelihood objective function. They maximized the log-likelihood...
while minimizing a penalty function measuring the sparsity of the unknown image. The introduction of a log-likelihood function comes naturally from the Poisson distributed measurements, with the nonnegativity constraint taken care of automatically.

One application is to measurements at consecutive time steps. For this problem, the images corresponding to two adjacent time steps are different in only a small fraction of the pixels. This application is analogous to one narrow application of 4D-PET, with the extra dimension representing time. With the extra dimension comes the advantage of dynamically capturing the internal movement of organs or tumors of a patient, which allows oncologists to see how the tumor moves with breathing and other normal body motions, facilitating dose predictions in different parts of the tumor. Cheng, et al. [73] used a maximum a posteriori (MAP) reconstruction algorithm that maximizes a Poisson log-likelihood and uses the Huber potential as the regularization term. In our method, we pose a prior in the reconstruction, which is the image from the previous time step. Here, we do not assume the image itself is sparse, instead, the difference image between two adjacent time steps is sparse, allowing a variety of applications in real medical systems.

The rest of the chapter is organized as follows. We give the problem definition, reformulation and the derived alternating minimization algorithm in Section 4.3. Section 4.4 gives the convergence theory for Poisson data reconstruction. Simulation results containing a random image and point-spread function as well as a circularly symmetric geometry with tomographic data and a digital circular phantom are presented in Section 4.5 to demonstrate the capability of reconstructing a sparse difference image using our algorithm. In the meantime, different $\lambda$'s which correspond to the level of sparsity are used in the implementation, to demonstrate the quantitative performance of our algorithm. An extension of the application to 4D CT with simulated tomographic data is presented in Section 4.6 and an approach to 4D PET is described in Section 4.8.
4.3 Methods

4.3.1 Problem Definitions

Consider a desired image \( c(x) \) viewed through a non-ideal system, where \( x \in \mathcal{X} \) represent image pixels. The measured data \( d(y), y \in \mathcal{Y} \) are noisy and modeled as being Poisson distributed with mean \( \mu(y) \) equal to a blurred version of the desired image, which is

\[
\mu(y) = \sum_x h(y|x)c(x),
\]

(4.1)

where \( h(y|x) \) is the nonnegative point-spread function determining the blur, or modeling tomographic measurements.

The problem to solve for \( c(x) \) when \( d(y) \) and \( h(y|x) \) are given is called the deblurring problem \footnote{Snyder et al. proposed \( I \)-divergence as a discrepancy measure for deblurring subject to the nonnegativity constraint, i.e., \( c(x) \) and \( h(y|x) \) are nonnegative. The iterative deblurring algorithm developed by Snyder, et al. \cite{74} seeks the maximum-likelihood estimator for the desired image \( c(x) \) since minimizing \( I \)-divergence over \( c(x) \) is equivalent to maximizing the log-likelihood function over \( c(x) \) \cite{1,77,78,79}. This problem setting is the same as in PET and Lange and Carson \cite{80}, Shepp and Vardi \cite{81}, as well as Kaufmann \cite{82} sought the maximum-likelihood reconstruction.}

Now consider two images \( c_1(x) \) and \( c_2(x) \) and assume they are different only in a small subset \( \mathcal{X}_{diff} \) of \( \mathcal{X} \). Two cases associated with this assumption are given below.

A. The problem is to find \( c_2(x) \) under the nonnegativity constraint when \( c_1(x) \), \( d_2(y) \) and \( h_2(y|x) \) are given. The objective function for this problem is

\[
\min_{c_2(x) \geq 0} \sum_y \left( d_2(y) \ln \frac{d_2(y)}{\mu_2(y)} - d_2(y) + \mu_2(y) \right) + \lambda \sum_x |c_1(x) - c_2(x)|,
\]

(4.2)

where \( \mu_2(y) = \sum_x h_2(y|x)c_2(x) \). The first term is the data fitting term corresponding to the \( I \)-divergence between the measured data and the estimated data for \( c_2(x) \). The
second term is the $L_1$ norm of the difference image between $c_1(x)$ and $c_2(x)$ with a weighting factor $\lambda$ that controls the sparsity of the difference image.

B. The problem is to find $c_1(x)$ and $c_2(x)$ jointly under the nonnegativity constraint when $d_1(y), d_2(y)$ and $h_1(y|x), h_2(y|x)$ are given. The objective function for this problem is

$$\min_{c_i(x) \geq 0} \sum_{i=1}^{2} \sum_y \left( d_i(y) \ln \frac{d_i(y)}{\mu_i(y)} - d_i(y) + \mu_i(y) \right) + \lambda \sum_x |c_1(x) - c_2(x)|, \tag{4.3}$$

where $\mu_i(y) = \sum_x h_i(y|x)c_i(x), i = 1, 2$. The first term is the data fitting term corresponding to the $I$-divergence between the measured data and the estimated data for $c_1(x)$ and $c_2(x)$. The second term is the $L_1$ norm of the difference image between $c_1(x)$ and $c_2(x)$ with a weighting factor $\lambda$ that controls the sparsity of the difference image.

### 4.3.2 Reformulation of Problem Statements

A. For the case 1 problem, the first order necessary (Kuhn-Tucker) conditions for an image $c_2^*(x)$ to be a minimizer are that for each $x$ such that $c_1(x) \neq c_2^*(x)$

$$\sum_y h_2(y|x) - \sum_y h_2(y|x) \frac{d_2(y)}{\sum_x h_2(y|x)c_2^*(x)} - \lambda \text{sgn} (c_1(x) - c_2^*(x)) = 0 \tag{4.4}$$

and for each $x$ such that $c_1(x) = c_2^*(x)$

$$\sum_y h_2(y|x) \geq \sum_y h_2(y|x) \frac{d_2(y)}{\sum_x h_2(y|x)c_2^*(x)}. \tag{4.5}$$

**Lemma 4.3.1** The $I$-divergence term in (4.2) can be written in the variational form

$$I(d_2||\mu_2) = \min_{p \in \mathcal{P}} \sum_y \sum_x \left( d_2(y)p(x|y) \ln \frac{d_2(y)p(x|y)}{h_2(y|x)c_2(x)} - d_2(y)p(x|y) + h_2(y|x)c_2(x) \right), \tag{4.6}$$

where

$$\mathcal{P} = \{p(x|y) \geq 0 : \sum_x p(x|y) = 1, \forall y \in \mathcal{Y} \}. \tag{4.7}$$
Comments:

Proof of this lemma can be easily obtained by introducing Lagrange multipliers to enforce the equality constraints of $p(x|y)$. Actually, this lemma is a simple application of the convex decomposition lemma [1].

Using this form, the original objective function (4.2) can be rewritten as

$$
\begin{aligned}
\min_{c_2(x), c_2(x) \geq 0} \min_{p \in P} & \sum_y \sum_x \left( d_2(y)p(x|y) \ln \frac{d_2(y)p(x|y)}{h_2(y|x)c_2(x)} - d_2(y)p(x|y) + h_2(y|x)c_2(x) \right) \\
+ \lambda \sum_x |c_1(x) - c_2(x)|.
\end{aligned}
$$

(4.8)

This double minimization leads naturally to an alternating minimization algorithm. This is a modified version of the EM algorithm and alternating minimization algorithms in the literature [74], [1], [77], [63], [64], [65].

B. For the case 2 problem, the first order necessary (Kuhn-Tucker) conditions for images $c^*_1(x)$ and $c^*_2(x)$ to be minimizers are that for each $x$ such that $c^*_1(x) \neq c^*_2(x)$

$$
\begin{aligned}
\sum_y h_1(y|x) - \sum_y h_1(y|x) \frac{d_1(y)}{\sum_x h_1(y|x)c^*_1(x)} + \lambda \text{sgn} \left( c^*_1(x) - c^*_2(x) \right) = 0,
\end{aligned}
$$

(4.9)

$$
\begin{aligned}
\sum_y h_2(y|x) - \sum_y h_2(y|x) \frac{d_2(y)}{\sum_x h_2(y|x)c^*_2(x)} - \lambda \text{sgn} \left( c^*_1(x) - c^*_2(x) \right) = 0,
\end{aligned}
$$

(4.10)

and for each $x$ such that $c^*_1(x) = c^*_2(x)$

$$
\begin{aligned}
\sum_y h_1(y|x) - \sum_y h_1(y|x) \frac{d_1(y)}{\sum_x h_1(y|x)c^*_1(x)} = \sum_y h_2(y|x) - \sum_y h_2(y|x) \frac{d_2(y)}{\sum_x h_2(y|x)c^*_2(x)} = 0.
\end{aligned}
$$

(4.11)
4.3.3 Alternating Minimization Iterations

A. Set the iteration index, \( k = 0 \). Select the initial guess \( c_2^{(k=0)}(x) = c_1(x), \forall x \in \mathcal{X} \). Pre-compute the sensitivity factor

\[
H_{20}(x) = \sum_y h_2(y|x). \quad (4.12)
\]

Compute the forward projection

\[
\mu_{2}^{(k)}(y) = \sum_x h_2(y|x) c_2^{(k)}(x). \quad (4.13)
\]

Compute the backward projection of the ratio image

\[
f^{(k)}(x) = \sum_y h_2(y|x) \frac{d_2(y)}{\mu_{2}^{(k)}(y)}. \quad (4.14)
\]

Compute temporary images

\[
c_{21}^{(k+1)}(x) = \frac{c_2^{(k)}(x)}{H_{20}(x) + \lambda f^{(k)}(x)}, \quad (4.15)
\]

and

\[
c_{22}^{(k+1)}(x) = \frac{c_2^{(k)}(x)}{H_{20}(x) - \lambda f^{(k)}(x)}. \quad (4.16)
\]

Update the final image

\[
c_2^{(k+1)}(x) = c_{21}^{(k+1)}(x), x \in \mathcal{X}_1, \quad (4.17)
\]

\[
\mathcal{X}_1 = \{ x : c_{21}^{(k+1)}(x) > c_1(x) \};
\]

\[
c_2^{(k+1)}(x) = c_{22}^{(k+1)}(x), x \in \mathcal{X}_2, \quad (4.18)
\]

\[
\mathcal{X}_2 = \{ x : c_{22}^{(k+1)}(x) < c_1(x) \};
\]

\[
c_2^{(k+1)}(x) = c_1(x), x \in \mathcal{X}_3, \quad (4.19)
\]

\[
\mathcal{X}_3 = \{ x : x \in \mathcal{X}, x \notin \mathcal{X}_1, x \notin \mathcal{X}_2 \}.
\]

Set \( k = k + 1 \). Check for convergence and iterate if necessary.
B. Set the iteration index, $k = 0$. Select the initial guess $c_1^{(k=0)}(x) = c_2^{(k=0)}(x), \forall x \in \mathcal{X}$. Precompute the sensitivity factors

$$H_{10}(x) = \sum_y h_1(y|x), H_{20}(x) = \sum_y h_2(y|x). \quad (4.20)$$

Compute the forward projections

$$\mu_1^{(k)}(y) = \sum_x h_1(y|x)c_1^{(k)}(x), \mu_2^{(k)}(y) = \sum_x h_2(y|x)c_2^{(k)}(x). \quad (4.21)$$

Compute the backward projections of the ratio images

$$f_1^{(k)}(x) = \sum_y h_1(y|x)d_1^{(k)}(y)/\mu_1^{(k)}(y), f_2^{(k)}(x) = \sum_y h_2(y|x)d_2^{(k)}(y)/\mu_2^{(k)}(y). \quad (4.22)$$

Compute temporary images

$$c_{11}^{(k+1)}(x) = c_1^{(k)}(x) \frac{H_{10}(x)}{H_{10}(x) + \lambda f_1^{(k)}(x)}, c_{21}^{(k+1)}(x) = c_2^{(k)}(x) \frac{H_{20}(x)}{H_{20}(x) - \lambda f_2^{(k)}(x)}. \quad (4.23)$$

$$c_{12}^{(k+1)}(x) = c_1^{(k)}(x) \frac{H_{10}(x) - \lambda f_1^{(k)}(x)}{H_{10}(x) + \lambda f_2^{(k)}(x)}, c_{22}^{(k+1)}(x) = c_2^{(k)}(x) \frac{H_{20}(x) + \lambda f_2^{(k)}(x)}{H_{20}(x) - \lambda f_1^{(k)}(x)}. \quad (4.24)$$

$$c_{13}^{(k+1)}(x) = c_{23}^{(k+1)}(x) = \frac{f_1^{(k)}(x) + f_2^{(k)}(x)}{H_{10}(x) + H_{20}(x)}. \quad (4.25)$$

Update the final images

$$c_1^{(k+1)}(x) = \begin{cases} c_{11}^{(k+1)}(x), & x \in \mathcal{X}_1, \mathcal{X}_1 = \{ x : c_{11}^{(k+1)}(x) > c_{21}^{(k+1)}(x) \}, \\ c_{12}^{(k+1)}(x), & x \in \mathcal{X}_2, \mathcal{X}_2 = \{ x : c_{12}^{(k+1)}(x) < c_{22}^{(k+1)}(x) \}, \\ c_{13}^{(k+1)}(x), & x \in \mathcal{X}_3, \mathcal{X}_3 = \{ x : x \in \mathcal{X}, x \notin \mathcal{X}_1, x \notin \mathcal{X}_2 \}. \end{cases} \quad (4.26)$$

$$c_2^{(k+1)}(x) = \begin{cases} c_{12}^{(k+1)}(x), & x \in \mathcal{X}_2, \mathcal{X}_2 = \{ x : c_{12}^{(k+1)}(x) < c_{22}^{(k+1)}(x) \}, \\ c_{13}^{(k+1)}(x), & x \in \mathcal{X}_3, \mathcal{X}_3 = \{ x : x \in \mathcal{X}, x \notin \mathcal{X}_1, x \notin \mathcal{X}_2 \}. \end{cases} \quad (4.27)$$

$$c_1^{(k+1)}(x) = c_2^{(k+1)}(x) = c_{13}^{(k+1)}(x) = c_{23}^{(k+1)}(x), x \in \mathcal{X}_3, \mathcal{X}_3 = \{ x : x \in \mathcal{X}, x \notin \mathcal{X}_1, x \notin \mathcal{X}_2 \}. \quad (4.28)$$

Set $k = k + 1$. Check for convergence and iterate if necessary.


4.4 Convergence Theory

Consider a sequence of Poisson data problems indexed by a parameter $T$ that is proportional to the total number of counts (for example, the data collection time in many applications, the concentration of radioactivity in others). There are two data collection intervals, $T_1$ for the reference image $c_1(x)$ and $T_2$ for the changed image $c_2(x)$. The data $d_i^{[T_i]}(y), y \in Y$ are Poisson with mean $T_i \mu_i(y), i = 1, 2,$ where $\mu_i(y) = \sum_{x \in X} h_i(y|x)c_i^*(x)$. For finite time collections, the penalized objective function is

$$L(c_1, c_2, \lambda, T_1, T_2) = I(d_1||T_1H_1c_1) + I(d_2||T_2H_2c_2) + \lambda \|c_1 - c_2\|_1.$$  

(4.29)

Factoring out the parameter $T_2$ gives

$$L(c_1, c_2, \lambda, T_1, T_2) = T_2 \left( \frac{T_1}{T_2} I \left( \frac{d_1}{T_1} \|H_1c_1 \right) + I \left( \frac{d_2}{T_2} \|H_2c_2 \right) + \lambda \left( \frac{T_1}{T_2} \|c_1 - c_2\|_1 \right) \right).$$  

(4.30)

The normalized data converge as parameters increase, $\frac{d_i^{[T_i]}(y)}{T_i} \to \mu_i(y)$ as $T_i \to \infty$ (in a mean square sense because the variances are proportional to $1/T_i$).

Minimizing the objective function (4.30) is the general problem considered. The objective function in (4.2) corresponds to the limiting case $T_1 \to \infty$ for which the minimizing estimator of $c_1(x)$ equals the truth $c_1^*(x)$ as long as the null space of the forward operator (matrix) $h_1(y|x)$ is empty. In this limiting case, the first term on the right side of (4.30) becomes irrelevant for the optimization, leaving only the other two terms. If both $T_1$ and $T_2$ get large, then the penalty is asymptotically negligible. For finite $T_1$ and $T_2$, the convex decomposition lemma can be applied to each of the first two terms in (4.30), and an expanded alternating minimization algorithm derived.

**Lemma 4.4.1** For nonnegative $\lambda$, the cost function (4.29) is convex in the pair $(c_1, c_2)$. Fixed points of the alternating minimization algorithm satisfy the Kuhn-Tucker conditions.
Lemma 4.4.2 Denote the estimators of $c_i(x), i = 1, 2, x \in X$ that minimize $L(c_1, c_2, \lambda, T_1, T_2)$ for finite fixed $\lambda$, with data $d_i^{(T_i)}(y)$ and parameter $T_i$, by $\hat{c}_i^{(T_i)}(x)$. Then for all $y \in Y$

$$
\lim_{T_1, T_2 \to \infty} \sum_{x \in X} h_i(y|x)\hat{c}_i^{(T_i)}(x) = \sum_{x \in X} h_i(y|x)c_i^*(x), i = 1, 2.
$$

(4.31)

Comments:

a. If the null spaces of both forward operators are empty, this lemma implies convergence of the estimators to the truth,

$$
\lim_{T_1, T_2 \to \infty} \hat{c}_i^{(T_i)}(x) = c_i^*(x), i = 1, 2, x \in X.
$$

(4.32)

b. If only $h_1(y|x)$ has an empty null space, then the lemma only implies the convergence of the estimator of the mean data $\mu_2(y)$ to the truth.

c. For any finite $T_2$, the penalty forces the estimator of $c_2$ to be close to $c_1$ while the I-divergence forces the estimator of the mean $\mu_2(y)$ to be close to the truth.

d. Even though the penalty is asymptotically negligible, it provides a selection rule for the limiting estimator.

Lemma 4.4.3 Suppose that $T_1/T_2 \to \infty$ and $T_2 \to \infty$, and that $h_1(y|x)$ has an empty null space. Then

$$
\lim \hat{c}_1^{(T_1)}(x) = c_1^*(x), \ x \in X
$$

$$
\lim \hat{c}_2^{(T_2)}(x) = \arg\min_{c_2 \in \mathcal{L}} \|c_1^* - c_2\|_1.
$$

(4.33)

where

$$
\mathcal{L} = \left\{ c_2(x) \geq 0 : \mu_2(y) = \sum_{x \in X} h_2(y|x)c_2(x), y \in Y \right\}
$$

(4.34)

Comments:

a. The asymptotic estimator of $c_2$ minimizes the $L_1$ distance to $c_1^*$ subject to matching the mean of the data.

b. If $c_1^*(x)$ and $c_2^*(x)$ differ at only a finite number of points, the minimization (4.33) does not guarantee that the estimator of our algorithm for $c_2$ differs from $c_1^*(x)$ at only a finite
number of points. The simulations show that for finite values of $T_2$ and for a wide range of values of $\lambda$, the number of locations where these estimators differ decreases monotonically as $\lambda$ increases.

c. Additional conditions would be needed for the solution to this $L_1$ penalized problem to yield the same components where the estimators differ as an $L_0$ penalized problem would (where the penalty equals the number of locations where the estimators differ). Our tomographic imaging problem is an example of this, where the difference values are large enough and the forward operator diverse enough that our algorithm can recover all image difference pixels.

4.5 Simulation Results

4.5.1 Simulation on Random Images

The simulation on random images presented in this section assumes perfect knowledge of the previous image $c_1(x)$. In real applications, this may not be available. However, one can still make inferences from the previous image, or jointly reconstruct the two time-adjacent images with the prior that they are different only in a few pixels.

The underlying images $c_1(x)$ and $c_2(x)$ are uniformly generated in the range of $[0, 1]$ with $|\mathcal{X}| = 64 \times 64$, independently for all pixels. They differ only in 63 randomly selected pixels, i.e., the difference image is sparse.

For the point-spread function, $h_2(y|x)$, we consider two scenarios. In scenario 1, $h_2(y|x)$ is uniformly generated in the range of $[0, 1]$ with $|\mathcal{Y}| = 2 \times 64 \times 64, |\mathcal{X}| = 64 \times 64$, i.e., there are twice as many measurements as unknown pixels. In scenario 2, $h_2(y|x)$ is uniformly generated in the range of $[0, 1]$ with $|\mathcal{Y}| = 0.5 \times 64 \times 64, |\mathcal{X}| = 64 \times 64$, i.e., there are twice as many unknown pixels as measurements. This setting is analogous to the CS problem in which one wants as small a number of measurements as possible. For both scenarios, the alternating minimization algorithm is implemented for noiseless data and terminates when successive objective function values differ by a relative amount less than $10^{-6}$.
Figure 4.1 gives the simulation results with different $\lambda$ values in the alternating minimization algorithm. This figure shows the tradeoff between the $I$-divergence term and the $L_1$ penalty term in the objective function. The smaller $\lambda$ values correspond to larger $L_1$ penalty values and smaller $I$-divergence values. Compared to scenario 1, scenario 2 has a “steeper” tradeoff trend, due to the lower number of measurements. Actually, it can be observed that for the same $L_1$ penalty value, the $I$-divergence value corresponding to scenario 2 is almost one fourth of the value corresponding to scenario 1, which matches the fact that scenario 2 has one fourth the number of measurements of scenario 1. Another measure of the quality of the reconstruction is the number of nonzero pixel values in the difference image $|c_1(x) - \hat{c}_2(x)|$ which match the true difference. For both scenarios, the number of nonzero pixel values and the number of correct nonzero pixel values in the reconstructed images are decreasing with increasing $\lambda$. Remember that in the true difference image, there are 63 nonzero pixel values. Therefore, for scenario 1, $\lambda = 0.02$ gives the best reconstruction and for scenario 2, $\lambda = 0.01$ gives the best reconstruction within the range of $\lambda$’s in this simulation. Moreover, by using only one fourth of the original measurements, only 4 fewer nonzero pixels are identified in scenario 2.

4.5.2 Simulation with Circular Geometry and Tomographic Data

A circularly symmetric geometry is used in this simulation. The geometry does not match PET geometries, but captures the essence of tomographic imaging problems. Lines through the image are determined as follows. There are 360 uniformly distributed sources over the full circle, and 92 detectors corresponding to each source on an arc of a circle centered at the source. The forward operator projects a $64 \times 64$ image with pixel size 1 mm to a detector or measurement space with $360 \times 92$ dimensions. That is, $|\mathcal{Y}| = 360 \times 92$, $|\mathcal{X}| = 64 \times 64$. The phantom images are shown in Figure 4.2. The true $c_1(x)$ and $c_2(x)$ images are different only in two inner circular regions with constant intensity 2 in the 6 o’clock region (region 1) and constant intensity 0.5 in the 10 o’clock region (region 2) for $c_2(x)$. For the rest of the circular phantom, both $c_1(x)$ and $c_2(x)$ have constant intensity 1. The reconstruction was performed only for pixels within a circle of radius 31 mm. Compared to the random point-spread function case in Section 4.5.1, this circularly symmetric tomographic geometry has a much sparser point-spread function.
Figure 4.1: Simulation results for a random image with different $\lambda$ values
In the first simulation, both noiseless data and Poisson noisy data are used to reconstruct \( c_2(x) \) for the Case 1 problem, as can be seen in Figure 4.3. In the noiseless case, the \( \lambda = 500 \) trial selects all 240 nonzero pixel values in the difference image. Also, one can visually see that pixel values in the two circular regions are quite uniform. However, for noisy data, one can observe rough regions in the reconstructed \( c_2(x) \) image even though the \( \lambda = 10000 \) trial selects all 240 nonzero pixel values in the difference image.

In the second simulation, both noiseless data and Poisson noisy data are used to reconstruct \( c_1(x) \) and \( c_2(x) \) jointly for the Case 2 problem, as can be seen in Figure 4.4. In the noiseless case, the \( \lambda = 100 \) trial selects all 240 nonzero pixel values in the difference image, also, one can visually see that pixel values in the two circular regions of \( c_2(x) \) and the one circular region in \( c_1(x) \) are quite uniform. However, for noisy data, one can observe rough regions in the reconstructed \( c_1(x) \) and \( c_2(x) \) images even though the \( \lambda = 1000 \) trial selects all 240 nonzero pixel values in the difference image.

Figure 4.5 gives the simulation results for the Case 1 problem with different \( \lambda \) values in the alternating minimization algorithm for both noiseless data and Poisson noisy data. This figure exhibits similar trends as in the random image and random point-spread function case, except that for noiseless data, with \( \lambda \)'s ranging from 500 to 800000, our algorithm can select all 240 nonzero pixel values in the difference image with 100% of them correct, since there are only 240 nonzero pixel values in the reconstructed difference image. For noisy data, with
\( \lambda \)'s ranging from 10000 to 1000000, our algorithm can select all 240 nonzero pixel values in the difference image with 100% of them correct. It seems that the introduction of noise in the dataset requires a larger weight for regularization or the penalty to enforce the algorithm to correctly select the nonzero pixels in the difference images.

Figure 4.6 gives the simulation results for the Case 2 problem with different \( \lambda \) values in the alternating minimization algorithm for both noiseless data and Poisson noisy data. This figure exhibits similar trends as in the Case 1 problem results, except that for noiseless data, with \( \lambda \)'s ranging from 100 to 400000, our algorithm can select all 240 nonzero pixel values in the difference image with 100% of them correct, since there are only 240 nonzero pixel values in the reconstructed difference image. For noisy data, with \( \lambda \)'s ranging from 1000 to 200000, our algorithm can select all 240 nonzero pixel values in the difference image with 100% of them correct.

For the Case 1 problem, in order to compare the reconstruction performance for reconstructed difference images with the same number of correct nonzero pixel values, we choose the two circular regions in the true difference image, i.e. the 6 o’clock region 1, the 10 o’clock region 2, and calculate the standard deviation of the corresponding reconstructed pixels. This can be seen in Figure 4.7. It can be observed that for noiseless data, the standard deviations for the two regions share the same trend as a function of \( \lambda \), which is a linear relationship. Moreover, smaller \( \lambda \) gives a more uniform performance in the two regions. For noisy data, the
trend is similar for relatively large $\lambda$’s. For the choices of $\lambda$’s in these simulations, $\lambda = 500$ gives the best reconstruction for noiseless data and $\lambda = 10000$ gives the best reconstruction for noisy data, in terms of number of nonzero correct pixel values and standard deviation in the two regions. The best scenario for noiseless data results gives much more uniform reconstruction in the two regions compared to the noisy case, and this can be seen in Figure 4.3 and Figure 4.7.

For the Case 2 problem, we pick three regions to examine the standard deviations, i.e., the center region of $c_1(x)$ and regions 1 and 2 in $c_2(x)$. Compared to the results in the Case 1
problem, the trends for the regions in $c_2(x)$ are very similar, while for the center region in $c_1(x)$, the standard deviation is smaller compared to that of $c_2(x)$. Moreover, for the choices of $\lambda$’s in these simulations, $\lambda = 100$ gives the best reconstruction for noiseless data and $\lambda = 1000$ gives the best reconstruction for noisy data, in terms of number of correct nonzero pixel values and the standard deviation in the three regions. This can be seen in Figure 4.4 and Figure 4.8.

4.6 Extension to 4D CT

While we have shown that the algorithm is able to reconstruct image differences when the data are Poisson distributed and the corresponding forward operator is known, we will show in this section that the idea of this algorithm can be easily extended to the 4D CT problem. Specifically, we consider the case when the data model is monoenergetic and the first image is known.

4.6.1 Problem Definitions

We will adopt a similar notation system as in Chapter 2. Denote the source-detector pairs by $y$, pixels by $x$. The transmission data $d(y)$ are assumed to be Poisson distributed with mean $g(y : c)$,

$$
g(y : c) = I_0(y) \exp \left( - \sum_x h(y|x)c(x) \right),
$$

where the line integrals in the forward model are approximated using a point-spread function $h(y|x)$, with units given in mm; the attenuation function $c(x)$ has units of mm$^{-1}$, and $I_0(y)$ represents the unattenuated photon counts or intensity. This data model is monoenergetic and the log-likelihood function is

$$
l(d : c) = \sum_y (d(y) \ln g(y : c) - g(y : c)).
$$
We have shown in Chapter 2 that maximizing log-likelihood function is equivalent to minimizing \( I \)-divergence as below,

\[
I(d||g(y : c)) = \sum_y \left( d(y) \ln \frac{d(y)}{g(y : c)} - d(y) + g(y : c) \right).
\]  (4.37)

Now consider two images \( c_1(x) \) and \( c_2(x) \) and assume they are different only in a small subset \( X_{\text{diff}} \) of \( X \) and the problem is to find \( c_2(x) \) under the nonnegativity constraint when \( c_1(x), d_2(y) \) and \( h_2(y|x) \) are given. The objective function for this problem is

\[
\min_{c_2(x) \geq 0} \sum_y \left( d_2(y) \ln \frac{d_2(y)}{g_2(y : c)} - d_2(y) + g_2(y : c) \right) + \lambda \sum_x |c_1(x) - c_2(x)|,
\]  (4.38)

where \( g_2(y : c) = I_{02}(y) \exp(-\sum_x h_2(y|x)c_2(x)) \). The first term is the data fitting term corresponding to the \( I \)-divergence between the measured data and the estimated data for \( c_2(x) \). The second term is the \( L_1 \) norm of the difference image between \( c_1(x) \) and \( c_2(x) \) with a weighting factor \( \lambda \) that controls the sparsity of the difference image.

### 4.6.2 Reformulation of Problem Statements

By using the convex decomposition lemma in a similar fashion as in [1], the original objective function (4.38) after iteration step \( k \) can be rewritten as

\[
\min_{c_2(x) \geq 0} \sum_y \left[ d_2(y) \ln \frac{d_2(y)}{I_0(y)} - d_2(y) \right] + \sum_{x,y} d_2(y)h_2(y|x)c_2(x)
\]

\[
+ \sum_x \left[ \sum_y I_0(y) \exp(-\sum_{x'} h_2(y|x')c_2^{(k)}(x'))h_2(y|x') \right] \exp\left(-\frac{Z_2(x)\triangle c_2(x)}{Z_2(x)}\right)
\]

\[
+ \lambda \sum_x |c_1(x) - c_2^{(k)}(x) - \triangle c_2(x)|,
\]  (4.39)

where \( \triangle c_2(x) = c_2(x) - c_2^{(k)}(x) \) and \( Z_2(x) = \max_y \sum_x h_2(y|x) \) is the normalization factor introduced from the convex decomposition lemma. A detailed derivation and justification can be also found in [1].
4.6.3 Sketch of the Alternating Minimization Algorithm

The first order necessary condition for \( \triangle c_2(x) \) after iteration \( k \) is

\[
\sum_y h_2(y|x) \left[ d_2(y) - I_0(y) \exp(-\sum_{x'} h_2(y|x') c_2^{(k)}(x')) \exp(-Z_2(x) \triangle c_2(x)) \right] - \lambda \text{sgn}(c_1(x) - c_2^{(k)}(x) - \triangle c_2(x)) = 0.
\]

(4.40)

Based on the sign of \( c_1(x) - c_2^{(k)}(x) - \triangle c_2(x) \), there are three cases to consider which forms our alternating minimization algorithm,

\[
\triangle c_2^{(k+1)}(x) = \triangle c_{21}^{(k+1)}(x) = -\frac{1}{Z_2(x)} \ln \frac{\sum_y h_2(y|x) d_2(y) + \lambda}{\sum_y h_2(y|x) I_0(y) \exp(-\sum_{x'} h_2(y|x') c_2^{(k)}(x'))}, \quad \text{for } x \in X_1, X_1 = \{ x : \triangle c_{21}^{(k+1)}(x) > c_1(x) - c_2^{(k)}(x) \};
\]

(4.41)

\[
\triangle c_2^{(k+1)}(x) = \triangle c_{22}^{(k+1)}(x) = -\frac{1}{Z_2(x)} \ln \frac{\sum_y h_2(y|x) d_2(y) - \lambda}{\sum_y h_2(y|x) I_0(y) \exp(-\sum_{x'} h_2(y|x') c_2^{(k)}(x'))}, \quad \text{for } x \in X_2, X_2 = \{ x : \triangle c_{22}^{(k+1)}(x) < c_1(x) - c_2^{(k)}(x) \};
\]

(4.42)

\[
\triangle c_2^{(k+1)}(x) = \triangle c_{23}^{(k+1)}(x) = 0, \quad \text{for } x \in X_3, X_3 = \{ x : \triangle c_{23}^{(k+1)}(x) = c_1(x) - c_2^{(k)}(x) \}.
\]

(4.43)

4.7 Simulation Results

4.7.1 Simulation with Circular Geometry and Tomographic Data

A circularly symmetric geometry which is the same as in Section 4.5.2 is used in this simulation. \( I_0 \) is selected to be 100000 and remains constant for every source-detector pair to simulate a Poisson noisy dataset with an approximate signal to noise ratio (SNR) of 0.32%. The true \( c_1(x) \) and \( c_2(x) \) images are different only in two inner circular regions with a Teflon insert (linear attenuation coefficient is 0.0398mm\(^{-1}\) at 75 keV) in the 6 o’clock region (region 1) and an ethanol insert (linear attenuation coefficient is 0.0158mm\(^{-1}\) at 75 keV) in the 10
o’clock region (region 2) for $c_2(x)$. The rest of the phantoms are both made of Lucite with linear attenuation coefficient of 0.0228mm$^{-1}$ at 75 keV. The reconstruction was performed only for the pixels within a circle of radius 31 mm. Figure 4.9 shows the true $c_1(x)$ and $c_2(x)$ images.

Both noiseless data and Poisson noisy data are used to reconstruct $c_2(x)$. Figure 4.10 shows the simulation results in terms of the number of correct reconstructed nonzero pixels in the difference images and the $I$-divergence value vs. $L_1$ penalty value plot with different $\lambda$ values in the alternating minimization algorithm for both noiseless data and Poisson noisy data. For noiseless data, with $\lambda$’s ranging from 200 to 200000, our algorithm can select all 240 nonzero pixel values in the difference image with 100% accurate, since there are only 240 nonzero pixel values in the true difference image. For noisy data, with $\lambda$’s ranging from 50000 to 200000, our algorithm can select all 240 nonzero pixel values in the difference image with 100% accurate. The introduction of noise in the dataset required use of larger weight for the regularization or the penalty to ensure the algorithm correctly selects the nonzero pixels in the difference images.

To get a better visualization of the reconstructed images, Figure 4.11 shows the reconstructed $c_2(x)$ for Poisson data with different $\lambda$ values. In Figure 4.11, all three choices of $\lambda$ yield images which perfectly select the nonzero pixels in the difference image. However, according to the reconstructed $c_2(x)$ images, the smallest $\lambda$ ($\lambda = 50000$) gives the best image performance in terms of uniformity inside the regions and the disk. Figure 4.12 shows profile plots through the 15th column of the reconstructed $c_2(x)$ images for different $\lambda$ values. According to the profile plot, smaller $\lambda$ values give better performance in terms of accuracy of the estimation.

### 4.8 An Approach to 4D PET

We now cast the 4D PET problem as an extension of our method for reconstructing the difference between two images. Given a sequence of time-bin measurements of arbitrary but known duration, and with no prior model on the image changes, reconstruct the sequence of images assuming the changes between time bins are sparse.
4.8.1 Problem Definitions

Consider a sequence of images $c_s(x)$ viewed through a non-ideal system, where $x \in \mathcal{X}$ represent image voxels and $s = 1, 2, \cdots, S$ indexes the time step or time frame. The measured data, $d_s(y)$, are noisy and modeled as being Poisson distributed with mean equal to a blurred version of the desired image, which is $\sum_x h(y|x)c(x)$, where $H = h(y|x)$ is the nonnegative point-spread function or forward operator determining the blur, or modeling tomographic measurements.

One way to estimate $c_s(x)$, $s = 1, 2, \cdots, S$ is to simply cast it as $S$ independent deblurring problems, which has been described before in [74]. However, in reality, since the measurement corresponding to each time step is usually of very short duration, the SNR in the measurement is very low, resulting in very noisy reconstructed images.

We propose to use an objective function with a sum of log-likelihoods for each time bin plus a sum of sparsifying penalties ($L_1$ norm) of all changes. Mathematically,

$$\min_{c_s(x), s=1, 2, \cdots, S} \sum_{s=1}^{S} I(d_s||Hc_s) + \sum_{s=1}^{S-1} \lambda_s \sum_x |c_s(x) - c_{s+1}(x)|. \quad (4.44)$$

Comments:

a. This is a nonparametric approach in the sense that there is no prior model on the dynamics.

b. The weights on the penalties ($\lambda_s$) may vary if the time bins are of different durations.

c. There is noise in the data, so that the sparsity penalty is needed to enforce some bins being equal across time.

d. If a voxel value is unchanged over two time bins, there essentially is integration over a longer time interval with a resulting improved image.
By using Lemma 4.3.1 for each of the $I$-divergence terms corresponding to different time steps in (4.44), the new lifted objective function can be written as

$$
\min_{c_s(x), s=1,2,\ldots,S} \left[ \sum_{s=1}^{S} \min_{p_s \in P_s} I(p_s \circ d_s || H \circ c_s) + \sum_{s=1}^{S-1} \lambda_s \sum_{x} |c_s(x) - c_{s+1}(x)| \right], \quad (4.45)
$$

where

$$
I(p_s \circ d_s || H \circ c_s) = \sum_y \sum_x \left[ p_s(x|y)d_s(y) \ln \frac{p_s(x|y)d_s(y)}{h(y|x)c_s(x)} - p_s(x|y)d_s(y) + h(y|x)c_s(x) \right], \quad (4.46)
$$

$$
P_s = \{p_s(x|y) \geq 0 : \sum_x p_s(x|y) = 1, \forall y \in \mathcal{Y} \}. \quad (4.47)
$$

Comments:

a. The objective function is decoupled across voxels which enables parallel updates.
b. The objective function is in the form of a chain and a forward-backward algorithm [83][84][85][86] can be applied.
c. The $I$-divergence terms do not involve interaction across time steps and this can simplify the updates.

### 4.8.2 Forward-backward Algorithm

To sum up how the forward-backward algorithm works, consider the minimization of a function of several variables that can be written as a sum of functions of consecutive variables. The forward-backward algorithm sequentially minimizes in a forward sweep the cost-to-get-there plus the next cost and then saves that total cost for each variable. Next, it sequentially minimizes in a backward sweep the cost-to-go plus the present cost and saves that total cost for each variable. At the end, it minimizes the sum of the forward and backward costs for each variable and this gives the global minimum for all the variables.
To describe the above statement in a mathematical way, consider the objective function as

\[
\min_{c_s, s=1,2,\ldots,S} f_1(c_1) + \sum_{s=1}^{S-1} f_s(c_s, c_{s+1}) + f_S(c_S). \tag{4.48}
\]

For the forward sweep, compute

\[
\alpha_s(c_s) = \min_{c_{s-1}} [f_{s-1}(c_{s-1}, c_s) + \alpha_{s-1}(c_{s-1})], \tag{4.49}
\]

with initial condition \(\alpha_1(c_1) = f_1(c_1)\).

For the backward sweeping, compute

\[
\beta_s(c_s) = \min_{c_{s+1}} [f_s(c_s, c_{s+1}) + \beta_{s+1}(c_{s+1})], \tag{4.50}
\]

with initial condition \(\beta_S(c_S) = f_S(c_S)\).

Then the minimizer \(c_s^* = \arg \min [\alpha_s(c_s) + \beta_s(c_s)]\). Here,

\[
\min [\alpha_s(c_s) + \beta_s(c_s)] = \min [\alpha_1(c_1) + \beta_1(c_1)], \tag{4.51}
\]

for all \(s\).

### 4.8.3 Algorithm Derivation

Before mapping the forward-backward algorithm summarized in Section 4.8.2, let us revisit the lifted objective function (4.45) first. As depicted in Section 4.3.2, if we only consider the \(I\)-divergence terms inside the objective function, at iteration \(k+1\), given the image estimator at iteration \(k\), the optimal variational operators are

\[
p_s^{(k+1)}(x|y) = \frac{h(y|x)c_s^{(k)}(x)}{\sum_{x'} h(y|x')c_s^{(k)}(x')}. \tag{4.52}
\]
If we define \( H_0(x) = \sum_y h(y|x) \), then the image estimator at iteration \( k + 1 \) by using the iterative deblurring method or the EM algorithm can be expressed as

\[
    c_{s,EM}^{(k+1)}(x) = \frac{1}{H_0(x)} \sum_y p_s^{(k+1)} d_s(y) = \frac{c_s^{(k)}(x)}{H_0(x)} \sum_y h(y|x) \frac{d_s(y)}{\sum_x h(y|x')} c_s^{(k)}(x).
\] (4.53)

If we keep only the terms that depend on the image, the objective function at iteration \( k + 1 \) for voxel \( x \) is

\[
    \min_{c_s(x)} \sum_{s=1}^S \left[ H_0(x)c_{s,EM}^{(k)}(x) \ln \frac{1}{c_s(x)} + H_0(x)c_s(x) \right] + \sum_{s=1}^{S-1} \lambda_s |c_s(x) - c_{s+1}(x)|. \tag{4.54}
\]

Now we can map our problem to the forward-backward problem as

\[
    f_1(c_1) = H_0(x)c_{1,EM}^{(k)}(x) \ln \frac{1}{c_1(x)} + H_0(x)c_1(x),
\]

\[
    f_s(c_s, c_{s+1}) = H_0(x)c_{s+1,EM}^{(k)}(x) \ln \frac{1}{c_{s+1}(x)} + H_0(x)c_{s+1}(x) + \lambda_s |c_s(x) - c_{s+1}(x)|, \tag{4.55}
\]

\[
    f_S(c_S) = 0.
\]

In terms of forward iterations, we start with the case \( s = 1 \). Then

\[
    \alpha_1(c_1) = H_0(x)c_{1,EM}^{(k)}(x) \ln \frac{1}{c_1(x)} + H_0(x)c_1(x),
\]

\[
    \alpha_2(c_2) = \min_{c_1} [f_1(c_1, c_2) + \alpha_1(c_1)] \tag{4.56}
\]

\[
    = \min_{c_1} \left[ - H_0(x)c_{2,EM}^{(k)}(x) \ln c_2(x) + H_0(x)c_2(x) + \lambda_1 |c_1(x) - c_2(x)| \\
    - H_0(x)c_{1,EM}^{(k)}(x) \ln c_1(x) + H_0(x)c_1(x) \right]
\]

We define \( c^*_s(x) \) as the minimizer that achieves the above minimum. It can be expressed as

\[
    c^*_s(x) = \begin{cases} 
    \frac{H_0(x)}{H_0(x)+\lambda_1} c_{1,EM}^{(k)}(x), & c_2(x) < \frac{H_0(x)}{H_0(x)+\lambda_1} c_{1,EM}^{(k)}(x) \\
    \frac{H_0(x)}{H_0(x)-\lambda_1} c_{1,EM}^{(k)}(x), & c_2(x) > \frac{H_0(x)}{H_0(x)-\lambda_1} c_{1,EM}^{(k)}(x) \\
    c_2(x), & \frac{H_0(x)}{H_0(x)+\lambda_1} c_{1,EM}^{(k)}(x) \leq c_2(x) \leq \frac{H_0(x)}{H_0(x)-\lambda_1} c_{1,EM}^{(k)}(x). \tag{4.57}
\end{cases}
\]

109
Therefore, the expression for $\alpha_2(c_2)$ can be found by substituting $c_1^*(x)$ for $c_1(x)$, with the decision order the same as in equation (4.57) yielding

$$
\alpha_2(c_2) = \begin{cases} 
-H_0(x)c_{2,EM}^{(k)}(x) \ln c_2(x) + H_0(x)c_2(x) - \lambda_1c_2(x) \\
-H_0(x)c_{1,EM}^{(k)}(x) \ln \frac{H_0(x)}{H_0(x)+\lambda_2}c_1^{(k)}_1(x) + H_0(x)c_{1,EM}^{(k)}(x), \\
-H_0(x)c_{2,EM}^{(k)}(x) \ln \frac{H_0(x)}{H_0(x)+\lambda_2}c_2^{(k)}(x) + H_0(x)c_{2,EM}^{(k)}(x), \\
-H_0(x)c_{1,EM}^{(k)}(x) \ln \frac{H_0(x)}{H_0(x)-\lambda_2}c_{1,EM}^{(k)}(x) + H_0(x)c_{1,EM}^{(k)}(x), \\
-H_0(x)\left[c_{1,EM}^{(k)}(x) + c_{2,EM}^{(k)}(x)\right] \ln c_2(x) + 2H_0(x)c_2(x).
\end{cases}
$$

(4.58)

By carefully examining the structure of $\alpha_2(c_2)$, we can form a condensed version of $\alpha_2(c_2)$ in which only $c_2(x)$ is involved and other constant terms are organized together,

$$
\alpha_2(c_2) = -H_0(x)c_2^{(k)}(x) \ln c_2(x) + \tilde{H}_{0,2}(x)c_2(x) + \text{other terms.}
$$

(4.59)

Similarly, $\alpha_{s-1}(c_{s-1})$ also has such a condensed version expressed as

$$
\alpha_{s-1}(c_{s-1}) = -H_0(x)c_{s-1}^{(k)}(x) \ln c_{s-1}(x) + \tilde{H}_{0,s-1}(x)c_{s-1}(x) + \text{other terms.}
$$

(4.60)

Therefore, the expression of $\alpha_s(c_s)$ can be found,

$$
\alpha_s(c_s) = \min_{c_{s-1}} \left[ -H_0(x)c_{s-1}^{(k)}(x) \ln c_{s-1}(x) + \tilde{H}_{0,s-1}(x)c_{s-1}(x) + \text{other terms} \right] \\
= \min_{c_{s-1}} \left[ -H_0(x)c_{s,EM}^{(k)}(x) \ln c_s(x) + H_0(x)c_s(x) + \lambda_{s-1}|c_{s-1}(x) - c_s(x)| \\
- H_0(x)c_{s-1}^{(k)}(x) \ln c_{s-1}(x) + \tilde{H}_{0,s-1}(x)c_{s-1}(x) + \text{other terms} \right].
$$

(4.61)

Then the minimizer $c_{s-1}^*(x)$ that achieves the above minimum can be expressed as

$$
c_{s-1}^*(x) = \begin{cases} 
\frac{H_0(x)}{H_{0,s-1}(x) + \lambda_{s-1}}c_{s-1}^{(k)}(x), & c_s(x) < \frac{H_0(x)}{H_{0,s-1}(x) + \lambda_{s-1}}c_{s-1}^{(k)}(x) \\
\frac{H_0(x)}{H_{0,s-1}(x) - \lambda_{s-1}}c_{s-1}^{(k)}(x), & c_s(x) > \frac{H_0(x)}{H_{0,s-1}(x) - \lambda_{s-1}}c_{s-1}^{(k)}(x) \\
c_s(x), & \frac{H_0(x)}{H_{0,s-1}(x) + \lambda_{s-1}}c_{s-1}^{(k)}(x) \leq c_s(x) \leq c_2(x) < \frac{H_0(x)}{H_{0,s-1}(x) - \lambda_{s-1}}c_{s-1}^{(k)}(x).
\end{cases}
$$

(4.62)
Therefore, the expression for \( \alpha_s(c_s) \) can be found by substituting \( c_{s-1}^*(x) \) for \( c_{s-1}(x) \), with the decision order the same as in equation (4.62),

\[
\alpha_s(c_s) = \begin{cases} 
-H_0(x)c_{s,EM}^{(k)}(x) \ln c_2(x) + H_0(x)c_s(x) - \lambda_{s-1}c_s(x) \\
-H_0(x)c_{s-1}^{(k)}(x) \ln \frac{H_0(x)}{H_{0,s-1}(x) + \lambda_{s-1}} c_{s-1}^{(k)}(x) + \tilde{H}_{0,s-1}(x)c_{s-1}^{(k)}(x), \\
-H_0(x)c_{s,EM}^{(k)}(x) \ln c_2(x) + H_0(x)c_s(x) + \lambda_{s-1}c_s(x) \\
-H_0(x)c_{s-1}^{(k)}(x) \ln \frac{H_0(x)}{H_{0,s-1}(x) - \lambda_{s-1}} c_{s-1}^{(k)}(x) + \tilde{H}_{0,s-1}(x)c_{s-1}^{(k)}(x), \\
-H_0(x) \left[ c_{s,EM}^{(k)}(x) + \tilde{c}_s^{(k)}(x) \right] \ln c_s(x) + H_0(x) + \tilde{H}_{0,s-1}c_s(x).
\end{cases}
\] (4.63)

As a verification, we can see that \( \alpha_s(c_s) \) also obeys the condensed form,

\[
\alpha_s(c_s) = -H_0(x)\tilde{c}_s^{(k)}(x) \ln c_s(x) + \tilde{H}_{0,s}(x)c_s(x) + \text{other terms}. \] (4.64)

The forward iteration can then be carried on by using this condensed form.

In terms of backward iterations, we start with a simplified case assuming we have computed \( c_1^*(x) \), \( c_2^*(x) \) and \( \tilde{c}_2^{(k)}(x) \), given by

\[
c_1^*(x) = \begin{cases} 
\frac{H_0(x)}{H_0(x) + \lambda_1} c_{1,EM}^{(k)}(x), & c_2(x) < \frac{H_0(x)}{H_0(x) + \lambda_1} c_{1,EM}^{(k)}(x) \\
\frac{H_0(x)}{H_0(x) - \lambda_1} c_{1,EM}^{(k)}(x), & c_2(x) > \frac{H_0(x)}{H_0(x) - \lambda_1} c_{1,EM}^{(k)}(x) \\
c_2(x), & \frac{H_0(x)}{H_0(x) + \lambda_1} c_{1,EM}^{(k)}(x) \leq c_2(x) \leq \frac{H_0(x)}{H_0(x) - \lambda_1} c_{1,EM}^{(k)}(x),
\end{cases}
\] (4.65)

\[
c_2^*(x) = \begin{cases} 
\frac{H_0(x)}{H_{0,2}(x) + \lambda_2} \tilde{c}_2^{(k)}(x), & c_2(x) < \frac{H_0(x)}{H_{0,2}(x) + \lambda_2} \tilde{c}_2^{(k)}(x) \\
\frac{H_0(x)}{H_{0,2}(x) - \lambda_2} \tilde{c}_2^{(k)}(x), & c_2(x) > \frac{H_0(x)}{H_{0,2}(x) - \lambda_2} \tilde{c}_2^{(k)}(x) \\
c_3(x), & \frac{H_0(x)}{H_{0,2}(x) + \lambda_2} \tilde{c}_2^{(k)}(x) \leq c_3(x) \leq \frac{H_0(x)}{H_{0,2}(x) - \lambda_2} \tilde{c}_2^{(k)}(x),
\end{cases}
\] (4.66)

\[
\tilde{c}_2^{(k)}(x) = \begin{cases} 
c_2^{(k)}(x) \\
\tilde{c}_2^{(k)}(x) \\
c_1^{(k)}(x) + c_2^{(k)}(x)
\end{cases}
\] (4.67)

111
Therefore, the expression for $\beta$ the decision order the same as in equation (4.70), yielding $c$ the decision regarding $c$. We index the decision regarding $c$. The algorithm needs to keep track of a small set of the potentially modified EM estimators firm up prior steps. If successive values do not change, the unknown value carries forward. Each time step, after the first time step, that successive values change, go back and A potential strategy based on the above simplified backward iterations is to back-track on the fly. Now we consider the backward iterations in general. For $s = S$, $\beta_S(c_S) = 0$, and

$$
\beta_{S-1}(c_{S-1}) = \min_{c_S} \left[ -H_0(x)c_{S,EM}^{(k)} \ln c_S(x) + H_0(x)c_S(x) + \lambda_{S-1}|c_{S-1}(x) - c_S(x)| \right].
$$

(4.69)

Then the minimizer $c_S^*(x)$ that achieves the above minimum can be expressed as

$$
c_S^*(x) = \begin{cases} 
\frac{H_0(x)}{H_0(x) + \lambda_{S-1}} c_{S,EM}^{(k)}, & c_{S-1}(x) < \frac{H_0(x)}{H_0(x) + \lambda_{S-1}} c_{S,EM}^{(k)} \\
\frac{H_0(x)}{H_0(x) - \lambda_{S-1}} c_{S,EM}^{(k)}, & c_{S-1}(x) > \frac{H_0(x)}{H_0(x) - \lambda_{S-1}} c_{S,EM}^{(k)} \\
c_{S-1}(x), & \frac{H_0(x)}{H_0(x) + \lambda_{S-1}} c_{S,EM}^{(k)} \leq c_{S-1}(x) \leq \frac{H_0(x)}{H_0(x) - \lambda_{S-1}} c_{S,EM}^{(k)}. 
\end{cases}
$$

(4.70)

Therefore, the expression for $\beta_{S-1}(c_{S-1})$ can be found by substituting $c_S^*(x)$ for $c_S(x)$, with the decision order the same as in equation (4.70), yielding

$$
\beta_{S-1}(c_{S-1}) = \begin{cases} 
-H_0(x)c_{S,EM}^{(k)} \ln \frac{H_0(x)}{H_0(x) + \lambda_{S-1}} c_{S,EM}^{(k)} \\
+H_0(x)c_{S,EM}^{(k)} - \lambda_{S-1}c_{S-1}(x), \\
-H_0(x)c_{S,EM}^{(k)} \ln \frac{H_0(x)}{H_0(x) - \lambda_{S-1}} c_{S,EM}^{(k)} \\
+H_0(x)c_{S,EM}^{(k)} + \lambda_{S-1}c_{S-1}(x), \\
-H_0(x)c_{S,EM}^{(k)} \ln c_{S-1}(x) + H_0(x)c_{S-1}(x).
\end{cases}
$$

(4.71)

A potential strategy based on the above simplified backward iterations is to back-track on the fly. Each time step, after the first time step, that successive values change, go back and firm up prior steps. If successive values do not change, the unknown value carries forward. The algorithm needs to keep track of a small set of the potentially modified EM estimators and sensitivity functions ($\hat{H}_{0,s}$).
4.9 Discussion

Throughout the simulations for 4D PET or 4D CT, $\lambda$ is fixed for each independent trial or simulation. One can adaptively select $\lambda$, or shrink $\lambda$ to achieve a desirable sparsity and accuracy. Our results with fixed $\lambda$ in each trial can be inferred in $\lambda$ adaption.

4.10 Conclusions andContributions

In this work, we considered a nonnegative linear regression problem with prior information. An alternating minimization algorithm was developed to minimize an objective function which is a combination of an $I$-divergence and a $L_1$ norm. Our algorithm exhibits monotonic decrease of the $I$-divergence and the objective function.

Both noiseless data and noisy data simulations were carried out to demonstrate the capability of our algorithm to reconstruct a sparse difference image. A random image with a random point-spread function, as well as a circularly symmetric geometry with tomographic data were considered. The algorithm was performed well for the tomographic case, where many pixels are different between the two images. The extension of 4D CT was straightforward and promising, leading to potential applications in 4D CT, even though only a monoenergetic data model for CT was considered. The proposed approach to 4D PET can jointly estimate the images corresponding to different time steps/frames, resulting in potential dose reduction in measurement and more accurate dynamic reconstruction.

Many more simulations need to be run to completely characterize the algorithm. While not reported here, preliminary results indicate better performance in the presence of Poisson noise than with LASSO. A more detailed comparison needs to be performed. We have seen perfect recovery of the difference image in some cases where the differences are large (many pixels are different between the two images). More theoretical work is needed to quantify the threshold at which such performance is obtained. Performance as a function of the number and type of measurements of the second image needs to be quantified.

The AM algorithm that reconstructs the image differences from tomographic Poisson data was developed by Dr. Joseph A. O’Sullivan. I implemented this algorithm on the simulated
data with a sequence of regularization parameters. I also carried out the experiments with the extension to the CT model. The extension algorithm for 4D PET by using the forward-backward algorithm was proposed by Dr. Joseph A. O’Sullivan.
Figure 4.5: Case 1 problem simulation results for the circular phantom image with different \( \lambda \) values in the alternating minimization algorithm
Figure 4.6: Case 2 problem simulation results for the circular phantom image with different \( \lambda \) values in the alternating minimization algorithm
Figure 4.7: Standard deviations as a function of $\lambda$ in region 1 and 2 for (a) Noiseless and (b) Noisy data for the Case 1 problem

Figure 4.8: Standard deviations as a function of $\lambda$ in center region of $c_1(x)$ and region 1 and 2 in $c_2(x)$ for (a) Noiseless and (b) Noisy data for the Case 2 problem
Figure 4.9: True (a) $c_1(x)$ and (b) $c_2(x)$ images
Figure 4.10: Simulation results for the circular phantom image with different $\lambda$ values in the alternating minimization algorithm.
Figure 4.11: Reconstructed $c_2(x)$ images with Poisson data
Figure 4.12: Profile plot at column 15 of reconstructed $c_2(x)$ image
Chapter 5

Adaptive X-Ray Sensing Study for a Poisson Model

5.1 Motivations

Compressed sensing problems for linear models have been studied extensively [87][88]. Great progress has been made in recent years by exploring intrinsic low-dimensional structure in high-dimensional objects. Based on the prior assumption that the object of interest can be represented as a linear combination of a few basis functions, sparse recovery can be carried out. The specific basis functions have to belong to a very large dictionary in order to enable sparse estimation. However, most of the existing theories and methods for the sparse recovery problem are based on nonadaptive measurements. In this chapter we investigate the possibility and advantages of sequential sampling schemes that adapt to intermediate estimation using information gathered throughout the sampling process. Specifically, we consider adaptive sensing for an X-ray Poisson model which is governed by Beer’s law.

5.2 General Physical Configuration and Data Model

Consider a measurement system with separate rings for detectors and sources. The source ring is the inner ring and the detector ring is the outer ring. The two rings are rotated together. Sources emit fan beams. We stored the used sources and choose which ring
orientation in \([0.5^\circ, 1.5^\circ, \ldots, 355.5^\circ]\) to use for the next measurement. We consider single and multiple source illuminations, which are illustrated in Figure 5.1.

The system matrix is denoted as \(h(\alpha, \beta, x)\) where \(\alpha\) denotes which detector, \(\beta\) denotes which source and \(x\) denotes an image pixel. We consider the monoenergetic CT data model which can be expressed as \(d_k(\alpha, \beta) : \text{Poisson}(q(\alpha, \beta))\), where \(k\) denotes acquisition step and \(q(\alpha, \beta)\) is

\[
q(\alpha, \beta) = I_0 e^{-\sum_x h(\alpha, \beta, x)c(x)\mu_{\text{water}}}.
\]  

Therefore, the log-likelihood function can be expressed as

\[
l(d) = \sum_k \sum_\alpha [d_k(\alpha, \beta_k) \log(I_0 e^{-\sum_x h(\alpha, \beta, x)c(x)\mu_{\text{water}}}) - (I_0 e^{-\sum_x h(\alpha, \beta, x)c(x)\mu_{\text{water}}})].
\]  

\[ \tag{5.2} \]

### 5.3 Adaptive Sensing for Single and Multiple Sources Illumination

The key step in an adaptive sensing strategy is to determine the optimal source position for the next measurement. For single source illumination, we find the next optimal source direction by finding the maximum log-determinant of the new Fisher information matrix
after adding another measurement from some source direction. At step \( k \), the calculated Fisher information matrix is

\[
J_k = J_0 - E \left[ \frac{\partial^2 l(d)}{\partial c(x)c(x')} \right] = \lambda I + I_0 \mu_{water}^2 \sum_k \sum_\alpha h(\alpha, \beta_k, x)h(\alpha, \beta_k, x')e^{-\sum x h(\alpha, \beta_k, x) c(x') \mu_{water}}. \tag{5.3}
\]

To choose the next optimal measurement source position, based on the current image estimate \( c^{(k)}(x) \), we sweep all the possible source locations, and compute the corresponding new Fisher information matrix,

\[
\tilde{J}_{k+1} \approx J_k + \Delta \tilde{J}_{k, \tilde{\beta}_k} = J_k + I_0 \mu_{water}^2 \sum_\alpha h(\alpha, \tilde{\beta}_k, x)h(\alpha, \tilde{\beta}_k, x')e^{-\sum x h(\alpha, \tilde{\beta}_k, x) c(x') \mu_{water}}. \tag{5.4}
\]

Then the next optimal source position \( \beta_{k+1} \) is chosen such that

\[
\beta_{k+1} = \arg \max_{\beta_k} \left[ \log |\tilde{J}_{k+1}| - \log |J_k| \right] = \arg \max_{\beta_k} \left[ \log |\tilde{J}_{k+1} J_k^{-1}| \right]. \tag{5.5}
\]

To test whether if the above algorithm works, we set up a simulation with inner ring radius 400 mm and outer ring radius 500 mm. The maximum fan angle is 90° and the image size is 64 by 64 with pixel size 4 mm by 4 mm. Poisson noisy data were generated with \( I_o = 100000 \). The forward projection is constructed such that the ray is defined from each source location to the center of each detector. We used a modified Shepp-Logan phantom as the true image shown in Figure 5.2.

The initialization image is obtained by using measurements from 18 evenly distributed source positions across the ring and running the monoenergetic AM algorithm for 360 iterations. Since this is equivalent to adding 18 source positions, we set \( k = 19 \) for the first adaptive sensing step. When the next source direction is selected, the monoenergetic AM update is computed using data corresponding to the new source direction for 10 iterations and then with 10 more iterations using all the data acquired so far. The initialization image, the image obtained by the non-adaptive sensing single-direction method (for every adaptive sensing
step, choose one optimal direction) after 32 acquisition steps, and the image obtained by
the adaptive sensing single-direction method after 32 acquisition steps are shown in Figure
5.3. Quantitatively, there seem to be no significant difference between results from the
non-adaptive and adaptive single-direction methods.

When multiple sources are used for illumination, for example, \( n \) sources, we have to sweep
all the \( n \) sources combinations in our source position pool, which is a much large number
compared to the single source case and is a much more complicated problem. Therefore,
for simulation purposes, we consider only selecting two sources. We use the same data
and same initialization as in the single source acquisition case, but we set \( k = 10 \) for the
first two sources adaptive sensing step. When the next two source directions are selected,
the monoenergetic AM update is computed using data corresponding to the new source
directions for 20 iterations and then with 20 more iterations using all the data acquired so
far. Figure 5.4 shows the reconstructed image from adaptive sensing with the two-direction
method after 32 acquisition steps.

Joint comparison between Figure 5.3 and Figure 5.4 does not show a significant difference
between the different methods considered here. We switch our performance measure to
normalized L\(_2\) norm error (L\(_2\) norm of the difference between the truth and the reconstructed
images over all pixels). Figure 5.5 gives the normalized L\(_2\) norm error vs. iteration number
for the different methods. The adaptive single-direction method starts to perform best at
iteration 26, while the non-adaptive two-direction method has almost the same performance
as the non-adaptive single-direction method. Due to computational limitations, the adaptive
Figure 5.3: Single source illumination reconstruction results

Figure 5.4: Two sources illumination reconstruction result
two-direction method runs for only 32 steps, and yields performance no better than the single-direction method. The superior performance of adaptive methods promises an even better trend as more source directions are adaptively selected.

### 5.4 Another Approach for Adaptive Sensing

Rather than comparing the log-determinant of the Fisher information matrix, one can look at the trace of $H_{\tilde{\beta}} J_k H_{\tilde{\beta}}^T$, where $H_{\tilde{\beta}} = \{h(\alpha, \beta, x), \beta = \tilde{\beta}\}$. Then the next optimal source direction $\beta_K = \arg\min_{\beta} \text{trace}(H_{\beta} J_k H_{\beta}^T)$. Of course, this strategy is only for single source selection.

To test whether the approach works, we used the same simulated data as before. The true image, the initialization image, and images obtained by the non-adaptive and adaptive methods are shown in Figure 5.6. The initial image is obtained by using the data from the 11 evenly distributed source positions across the ring. The adaptive image is obtained after 60 adaptive sensing steps. The non-adaptive image is obtained after 60 non-adaptive sensing steps by sweeping every 4 source positions. The image obtained after 1000 iterations using all the data without regularization is also displayed for comparison purpose. According
to these reconstruction results, the adaptive single-direction method using minimum trace selection rules does not perform as well as the uniform sampling strategy.

### 5.5 Conclusions and Contributions

Different approaches for X-ray adaptive sensing are proposed and simulation data reconstructions were performed. Early simulation results show better image reconstruction performance for adaptive sensing methods compared to the non-adaptive sensing method in terms of normalized L$_2$ norm error. However, due to computational limitations, the multiple-directions method cannot be carried out for more iterations. A more computational efficient method needs to be developed for adaptively selecting multiple directions. The better performance of adaptive sensing promises its future application in dose reduction and many other areas.

The selection rule used for the adaptive sensing step was proposed by Dr. Joseph A. O’Sullivan. I implemented the different adaptive sensing approaches and designed simulations to test these approaches. We discussed results and approaches with and received feedback from Dr. Yan Kaganovsky at Duke University.
Figure 5.6: True and reconstructed images by using the non-adaptive and adaptive single source method with minimum trace selection rule
Chapter 6

Computation of Berger-Tung Bounds for Lossy Distributed Source Coding

6.1 Abstract

Inner and outer bounds for the achievable rate regions for distributed source coding have been derived by several authors [89][90][91][92], building on the work by Berger and Tung [93][94]. We describe an optimization and computational approach for characterizing these regions. Optimization is based on a Lagrangian that incorporates the mutual information and distortion terms in one cost functional. Computation results from lifting the Lagrangian to a higher dimensional space through the introduction of auxiliary variables corresponding to posterior probability distributions. This relies on two sets of new variational equalities for mutual information, extending those introduced by Blahut and Arimoto [95] to multi-variable probability distributions [96][97][98][99]. A probability distribution is introduced for each decoding function, and included in the iterations with a damping term. Alternately minimizing this lifted functional over primary and auxiliary variables yields a sequence of probability distributions that converge to fixed points. Known convergence properties for the algorithms are derived. Several two-variable examples are presented. For each problem considered that has a sum of distortions on the encoded variables, the inner and outer bound regions coincide. For a problem defined by Wagner and Anantharam [4] with a single joint distortion for the two variables, their gap is observed in our results. These boundary regions can motivate hypothesized optimal distributions which can be tested in the first order necessary conditions for the optimal distributions.
6.2 Introduction

The problem studied in this chapter is the computation of inner and outer bounds on the achievable rate region for lossy distributed source coding. Alternating minimization algorithms are derived, implemented, and tested on example problems. These algorithms have at their core the philosophy described by Blahut in his 1972 paper [100]. The idea is to write each mutual information in the problem using a variational representation [101][64][102]. That is, the mutual information equals the maximum or the minimum of an auxiliary functional, where the optimization is over auxiliary variables. These additional variables effectively lift the problem to a higher dimensional space where the optimization is more easily understood. An alternating minimization algorithm results, where the primary and the auxiliary variables are alternately updated.

6.2.1 Definition of Berger-Tung Regions

Suppose that the joint distribution on a pair of random variables $(X,Y)$ taking values in $\mathcal{X} \times \mathcal{Y}$ is denoted by $p(x,y)$. This distribution is known and fixed. Define the sets of probability distributions

\[ P_i = \{ p(x,y,u,v) \in \mathcal{P} : p(x,y,u,v) = p(x,y)p(u|x)p(v|y) \} \]

\[ P_o = \{ p(x,y,u,v) \in \mathcal{P} : p(x,y,u) = p(x,y)p(u|x) \text{ and } p(x,y,v) = p(x,y)p(v|y) \} \]

\[ P_{mix} = \{ p(x,y,u,v) \in \mathcal{P} : p(x,y,u,v) = p(x,y) \sum_k p(u|x,k)p(v|y,k)p(k) \}. \]

The random variables $U$, $V$, and $K$ take values in finite sets $\mathcal{U}$, $\mathcal{V}$, and $\mathcal{K}$, respectively; $k \in \mathcal{K}$ is a hidden random variable known to both encoders, independent of $(X,Y)$, and unknown at the decoder. The set $P_i$ defines a length four Markov chain $U - X - Y - V$ while $P_o$ and $P_{mix}$ have distributions that satisfy two length three Markov chains $U - X - Y$ and $X - Y - V$. 

131
For reproduction alphabets $\hat{X}$ and $\hat{Y}$, any functions $g : \mathcal{U} \times \mathcal{V} \to \hat{X}$ and $h : \mathcal{U} \times \mathcal{V} \to \hat{Y}$ are estimators of the pair $(X, Y)$. The pair of distortions achieved by estimators $g$ and $h$ is

$$D_X = E[d_X(X, g(U, V))]$$
$$D_Y = E[d_Y(Y, h(U, V))],$$

where the distortion measures $d_X$ and $d_Y$ (or just $d$) are nonnegative and assumed here to be finite-valued; these distortion measures can be extended to vectors by taking the average per symbol distortion.

For a given joint distribution $p(x, y, u, v)$, define the regions of possible quadruples of rates and distortions as

$$\mathcal{R}(p) = \{(R_X, R_Y, D_X, D_Y) : I(U; X, Y | V) \leq R_X, I(V; X, Y | U) \leq R_Y, I(U, V; X, Y) \leq R_X + R_Y, E[d_X(X, g(U, V))] \leq D_X, E[d_Y(Y, h(U, V))] \leq D_Y\}.$$  (6.3)

The Berger-Tung inner region $\mathcal{R}_i$ and outer region $\mathcal{R}_o$ are

$$\mathcal{R}_i = \text{co}\{\mathcal{R}(p) : p \in \mathcal{P}_i\}$$
$$\mathcal{R}_o = \{\mathcal{R}(p) : p \in \mathcal{P}_o\},$$

where $\text{co}\{\cdot\}$ denotes convex hull. Define the region corresponding to mixtures of length four Markov chains

$$\mathcal{R}_{\text{mix}} = \{\mathcal{R}(p) : p \in \mathcal{P}_{\text{mix}}\}.$$  (6.6)

We note that $\mathcal{R}_o$ and $\mathcal{R}_{\text{mix}}$ are convex regions. Berger and Gibson [103] give a historical tour of source coding with many additional references. See Westover and O’Sullivan [104] for a motivation for the mixture distributions.
6.3 Computational Problems

We consider the computation of $\mathcal{R}_i$ and $\mathcal{R}_{\text{mix}}$. For $\mathcal{R}_i$, $I(U;V|X,Y) = 0$ so the rate bounds become

$$
I(U;X,Y|V) = I(U;X) - I(U;V) \leq R_X
\quad (6.7)
$$

$$
I(V;X,Y|U) = I(V;Y) - I(U;V) \leq R_Y
\quad (6.8)
$$

$$
I(U,V;X,Y) = I(U;X) + I(V;Y) - I(U;V) \leq R_X + R_Y.
\quad (6.9)
$$

The three values of mutual information $I(U;X)$, $I(V;Y)$, and $I(U;V)$ determine the left sides of the rate bounds. Our approach is to characterize the achievable sets of these mutual informations subject to the distortion constraints. Subject to upper bounds on $I(U;X)$ and $I(V;Y)$, we seek to maximize $I(U;V)$. For every admissible joint distribution, each pair of constraints (6.7)-(6.8), (6.7)-(6.9), and (6.8)-(6.9) determines a rate-distortion quadruple $(R_X, R_Y, D_X, D_Y)$ in the region. The convex hull of all such points determines a boundary of the region. For the inner bound, the admissible distributions are in $\mathcal{P}_i$. For the outer bound, they are in $\mathcal{P}_{\text{mix}}$.

For any joint distribution there are corresponding distributions

$$
\mathcal{Q}_i = \{q(u), q(v), \text{and } q(u,v)\}
\quad (6.10)
$$

for the inner bound and

$$
\mathcal{Q}_o = \{q(u), q(v), q(k|x,u), q(k|y,v), q(u,v), q(k|x,y,u,v)\}
\quad (6.11)
$$

for the outer bound. These may be thought of as being determined directly from the joint distribution or as the result of optimization problems

$$
\min_J(p,q,\lambda,\mu,\nu)
\quad (6.12)
$$
where \( J \) is the Lagrangian. For the inner bound, the Lagrangian is of the form

\[
J(p, q, \lambda, \mu, \nu) = \sum_{(x, y, u, v)} p(x, y) p(u| x) p(v| y) \log \frac{p(u| x) p(v| y)}{q(u, v)}
\]

\[
+ \lambda_1 \sum_{(x, u)} p(x) p(u| x) \log \frac{p(u| x)}{q(u)}
\]

\[
+ \lambda_2 \sum_{(y, v)} p(y) p(v| y) \log \frac{p(v| y)}{q(v)}
\]

\[
+ \mu_1 \sum_{(x, y, u, v)} p(x, y) p(u| x) p(v| y) d_X(x, g_X(u, v))
\]

\[
+ \mu_2 \sum_{(x, y, u, v)} p(x, y) p(u| x) p(v| y) d_Y(y, g_Y(u, v))
\]

\[
+ \sum_{(x, u)} \nu_1(x) p(u| x) + \sum_{(y, v)} \nu_2(y) p(v| y)
\]

\[
+ \sum_{(u)} \nu_3 q(u) + \sum_{(v)} \nu_4 q(v) + \sum_{(u, v)} \nu_5 q(u, v).
\] (6.13)

The \( \nu \) Lagrange multipliers enforce the probability constraints.

For the outer bound, the Lagrangian is of the form

\[
J(p, q, \lambda, \mu, \nu) = \sum_{(x, y, u, v, k)} p(x, y) p(u| x, k) p(v| y, k) p(k)
\]

\[
\times \log \frac{p(u| x, k) p(v| y, k) p(k)}{q(u, v) q(k| x, y, u, v)}
\]

\[134\]
\[
+ \lambda_1 \sum_{(x,u,k)} p(x)p(u|x,k)p(k) \log \frac{p(u|x,k)p(k)}{q(u)q(k|x,u)} \\
+ \lambda_2 \sum_{(y,v,k)} p(y)p(v|y,k)p(k) \log \frac{p(v|y,k)p(k)}{q(v)q(k|y,v)} \\
+ \mu_1 \sum_{(x,y,u,v,k)} p(x,y)p(u|x,k)p(v|y,k)p(k)d_X(x, g_X(u,v)) \\
+ \mu_2 \sum_{(x,y,u,v,k)} p(x,y)p(u|x,k)p(v|y,k)p(k)d_Y(y, g_Y(u,v)) \\
+ \text{other terms,}
\]

where the other terms have Lagrange multipliers that enforce the probability constraints.

### 6.3.1 Optimal Joint Distributions

Optimal distributions satisfy the first order necessary (KKT) conditions that are determined from these Lagrangians. These conditions can be used to check candidate optimal distributions.

### 6.3.2 Alternating Minimization Algorithms

For either the inner or outer bound, an alternating minimization algorithm alternately computes a candidate joint distribution and then the corresponding auxiliary distributions in either \( Q_i \) or \( Q_o \). Given our Lagrangians, these equations are derived in a straightforward manner. Because the derivative of the equations with respect to \( p(u|x,k) \) for the outer region depend on \( p(v|y,k) \), the updates for the joint distribution are performed in parts, alternating between an updating \( p(u|x,k) \) at one step (followed by an update of \( Q_o \)) and then updating \( p(v|y,k) \) (again followed by an update of \( Q_o \)).

To determine the decoding functions \( g_X \) and \( g_Y \), we introduce probability distributions \( p_X(\hat{x}|u,v) \) and \( p_Y(\hat{y}|u,v) \) and weighted relative entropy penalties with respect to the last values of these distributions \( D(p^{k+1}_X||p^k_X) \) and \( D(p^{k+1}_Y||p^k_Y) \). These probability distributions are used to compute the average distortion. The relative entropy penalties damp (slow) the
convergence of these probability distributions to functions (probability one associated with
the decoded values).

The factors $\lambda_1, \lambda_2, \mu_1,$ and $\mu_2$ are used to sweep out the surface of the rate region. Every
convergence point determines three potential points on the surface (but all in the region) as
described above.

### 6.4 Motivational Examples

#### 6.4.1 Doubly Symmetric Binary Source (DSBS($p$))

Let $(\bar{X}, \bar{Y})$ be a DSBS($p$), where $\bar{X}$ and $\bar{Y}$ are binary random variables with $p_{\bar{X}, \bar{Y}}(0, 0) = p_{\bar{X}, \bar{Y}}(1, 1) = (1 - p)/2$ and $p_{\bar{X}, \bar{Y}}(0, 1) = p_{\bar{X}, \bar{Y}}(1, 0) = p/2$, $p \in (0, 1/2)$. Thus, $\bar{X} \sim$ Bernoulli$(1/2)$, $\bar{Y} \sim$ Bernoulli$(1/2)$. Let $X = (\bar{X}, \bar{Y})$, $Y = \bar{Y}$, $d_X(X, \hat{X}) = d_{\bar{X}}(\bar{X}, \hat{X})$ be a Hamming distortion measure, and $d_Y(Y, \hat{Y}) \equiv 0$. We consider the minimum achievable rate for distortion $D_X$. By taking $p(v|\bar{y})$ to be a binary symmetric channel (BSC) ($\alpha$), the rate distortion region for distributed lossy source coding is the set of rate pairs $(R_X, R_Y)$ such that

$$
R_X > H(\alpha \ast p) - H(D_X), \\
R_Y > 1 - H(\alpha),
$$

for some $\alpha \in [0, 1/2]$ that satisfies the constraint $H(\alpha \ast p) - H(D_X) \geq 0$. The operator $\ast$ is defined as $a \ast b = a(1 - b) + (1 - a)b$, $a \in [0, 1], b \in [0, 1]$.

This example is borrowed from [105], and we used our algorithm to compute the inner and outer bound regions in terms of tripletuples $R_X, R_Y$ and $D_X$ for $p = 0.2$.

In this example, due to the common part or side information of the input [106][107][108], the encoder $X$ can encode $\bar{Y}$ the same way encoder $Y$ does, i.e., encoder $X$ has full access to encoder $Y$. Therefore, there is no gap between the inner and outer bound surfaces, which can be shown in Figure 6.1, and this surface matches the surface described in (6.15).
6.4.2 Concatenation of Two DSBS Sources

Consider concatenating two DSBS sources, DSBS\((p)\) and DSBS\((q)\), respectively. That is, 
\[
p_{(X,Y)}(0,0) = p_{(X,Y)}(1,1) = (1 - p)/4, \quad p_{(X,Y)}(1,0) = p_{(X,Y)}(0,1) = p/4, \quad p_{(X,Y)}(2,2) = p_{(X,Y)}(3,3) = (1 - q)/4, \quad p_{(X,Y)}(3,2) = p_{(X,Y)}(2,3) = q/4.
\] The rest of the joint probabilities are zero. Thus, \(X \sim \text{Uniform}(1/4)\), \(Y \sim \text{Uniform}(1/4)\). The distortion measure is Hamming distortion.

We used our algorithm to compute the inner and outer bound regions in terms of quadtuples \(R_X, R_Y, D_X\) and \(D_Y\) for \(p = 0.1, q = 0.2\).
Figure 6.2 shows how the objective function value varies with iterations for some of inner and outer bounds computations, with computation parameters $|\mathcal{U}| = |\mathcal{X}| = |\mathcal{Y}| = |\mathcal{V}| = 4, |\mathcal{K}| = 2$. Monotonic decrease of the objective function is achieved in the algorithm for both inner and outer bound computations, and convergence is achieved within 40 iterations in general. The objective function values drop fast in the first ten iterations with random initial conditions. In terms of the computational complexity for this example, for the outer bound computation, each iteration requires approximately $80,000$ operations (including $+$, $\times$, $\exp$) and for the inner bound, each iteration requires approximately $20,000$ such operations.

Figure 6.3 shows the computed Berger-Tung inner and outer bound points and convex hull surface corresponding to $D_Y < D_{Y_{\text{max}}}$ for concatenation of two DSBS sources example. The maximum distortion for $X$ is $0.75$ since $X \sim \text{Uniform}(1/4)$. The maximum rate pair $(R_X, R_Y)$ on the surface, which is the line formed by the surface hitting the X-Y plane, corresponds to Slepian-Wolf coding \cite{109}.

Figure 6.4 gives the computed Berger-Tung inner and outer bound points and convex hull surface with Z-axis the sum distortion and X-Y plane the rate pair $(R_X, R_Y)$. There is no gap between inner and outer bound surfaces. The maximum sum distortion is $1.5$ since $X \sim \text{Uniform}(1/4)$ and $Y \sim \text{Uniform}(1/4)$. The maximum rate pair $(R_X, R_Y)$ on the surface which is the line formed by the surface hitting X-Y plane corresponds to Slepian-Wolf coding.
Figure 6.4: (a) Computed Berger-Tung inner and outer bound points and (b) convex hull surface for concatenation of two DSBS sources example

Figure 6.5: (a) Computed Berger-Tung inner and outer bound convex hull surface corresponding to $D_X < D_{X_{\text{max}}}$ and (b) convex hull surface for asymmetric sources of different cardinalities example

6.4.3 Asymmetric Sources of Different Cardinalities

Consider $X, Y_1, Y_2$ being Bernoulli(1/2) with joint distribution of $(X, Y_1)$ and $(X, Y_2)$ modeled as one Z-channel and one inverse Z-channel, where $X$ is the common input and $Y_1, Y_2$ is the output of the two channels, respectively. For $(X, Y_1)$, the Z-channel is defined as $p(Y_1|X)(0,0) = 1$, $p(Y_1|X)(0,1) = p(Y_1|X)(1,1) = 1/2$. For $(X, Y_2)$, the inverse Z-channel is defined as $p(Y_2|X)(1,1) = 1$, $p(Y_2|X)(0,0) = p(Y_2|X)(1,0) = 1/2$. Let $Y = (Y_1, Y_2)$, and the distortion measure for $X$ be Hamming distortion and the distortion measure for $Y$ be the sum of Hamming distortions for $Y_1$ and $Y_2$. 
Figure 6.5(a) shows the computed Berger-Tung inner and outer bound convex hull surface corresponding to $D_X < D_{X_{max}}$. There is no gap between the inner and outer bound surfaces. The maximum distortion for $Y$ is 0.5. The maximum rate pair $(R_X, R_Y)$ on the surface, which is the folding line formed by the surface hitting X-Y plane, corresponds to Slepian-Wolf coding and a perfectly encoding $Y$, i.e., $R_Y = H(Y)$.

Figure 6.5(b) shows the computed Berger-Tung inner and outer bound convex hull surface with the Z-axis representing the sum distortion and the X-Y plane the rate pair $(R_X, R_Y)$. There is no gap between the inner and outer bound surfaces. The maximum sum distortion is 1. The maximum rate pair $(R_X, R_Y)$ on the surface, which is the line formed by the surface hitting X-Y plane, corresponds to Slepian-Wolf coding. The curve which has maximum $R_Y$ corresponds to rate-distortion quaduples,

$$
R_X = 0.5(1 - h(2D)), R_Y = 1.5,
$$

$$
D_X = D, D_Y = 0, 0 \leq D \leq 0.25. \tag{6.16}
$$

### 6.4.4 Wagner and Anantharam Example

Let $X_1, X_2, Y_1, Y_2$ be i.i.d. random and Uniform(1/2). Consider $X = (X_1, X_2), Y = (Y_1, Y_2)$, and the decoder yields single $Z = \{0, 1\}^2$ with distortion

$$
d((X, Y), Z) = \begin{cases} 
0 & \text{if } Z_1 = (X_1, Y_1) \text{ or } Z_2 = (X_2, Y_2), \\
1 & \text{otherwise.}
\end{cases}
$$

This example was first introduced by Wagner and Anantharam [4]. This problem is symmetric in $X$ and $Y$; therefore, the analyses for the encoders of $X$ and $Y$ are the same and an analysis on the sum rate and distortion will suffice. Figure 6.6 shows the inner and outer bounds for the sum rate vs. the distortion as well as the gap between the inner and outer bounds. In particular, the sum rate and distortion pair $(1.5, 0)$ matches the point gap in Wagner and Anantharam’s description [4].
For the outer bound, the analytical form of the sum rate and distortion pair is,

\[
R_X + R_Y = 1.5 - h(\alpha) - 0.5h(2\alpha(1 - \alpha)),
\]

\[
D_Z = 0.75(2\alpha - \alpha^2), 0 \leq \alpha \leq 0.5. \quad (6.17)
\]

### 6.5 Contributions and Conclusions

We described an optimization and computational approach for characterizing the inner and outer bounds for the achievable rate regions for Berger-Tung coding. The optimization is based on a Lagrangian that incorporates the mutual information and distortion constraints in one cost functional. Computation can be performed by lifting the Lagrangian to a higher dimensional space through the introduction of auxiliary variables corresponding to posterior probability distributions. In addition, a probability distribution corresponding to each decoding function is included in the objective function with a damping term. Through alternately minimizing this lifted objective function over primary and auxiliary variables, a sequence of probability distributions that converge to fixed points can be obtained. Moreover, monotonic decrease of the objective function is achieved through such alternating minimization. Four two-variable examples are presented with brief interpretations in terms of encoding strategy that can achieve the bound. Each problem is presented through the sum of distortions on the encoded variables, and for three out of the four examples, the inner and outer bound
regions coincide. For the problem defined by Wagner and Anantharam with a single joint distortion for the two variables, the gap they described is observed in our results. For one of the problems, computational complexity is analyzed in terms of the number of operations needed for every alternating minimization iteration. These computed boundary regions can motivate hypothetical optimal distributions which can be tested in the first order necessary conditions for the optimal distributions.

The characterization and analysis for the Berger-Tung inner and outer bounds were developed by Dr. Joseph A. O’Sullivan. I implemented the computation tools and tested our algorithms on several examples.
Chapter 7

Conclusions and Future Work

7.1 Contributions

This work first explored the effect of regularization on existing DE-AM algorithm reconstruction performance by using noiseless data, noisy data, and noisy data with scatter generated, based on the Siemens Somatom Plus 4 scanner geometry. At the same time, system bias and variance were studied by using noisy and simulated data with scatter. An edge-preserved penalty function was incorporated in the DE-AM algorithm and different regularization parameters were used in a sequence of simulated data reconstructions, in order to find a good range for those parameters. A similar approach was used on real 2D data from the Philips Brilliance scanner. Moreover, real 3D data from the Philips Brilliance scanner were reconstructed with the utilization of HECTARE, demonstrating the ability to reconstruct 3D data from other vendors, too.

Next, studies were performed to understand the reason for the slow convergence of the AM algorithms, specifically, the DE-AM algorithms. Visualization of a toy problem with only two parameters to be estimated shows a “hill-valley” phenomenon, indicating an ill-conditioned problem setting, which is the high correlation between the estimates of the two component images.

In order to cope with the ill-conditioned problem, the line integral AM algorithm was developed, the idea of which is to estimate the line integrals of the two component images first, and then estimate the component images based on the estimated line integrals. A penalty on the estimated line integrals and the forward projections of the component images, as well
as an image neighborhood penalty, can be included in the LIAM, in order to achieve certain image smoothness while maintaining relatively accurate estimation. Simulated noiseless and noisy data were used for reconstructions, and similarly, a sequence of noisy data reconstructions were performed in order to find appropriate regularization parameters. A much faster convergence rate was observed by using the LIAM compared to the DE-AM algorithms and more accurate estimation can be obtained, even at low energy levels where few photons are detected. Real axial data from the Philips Brilliance scanner were also reconstructed and results show accurate average estimation. However, the images obtained were very noisy.

A nonnegative linear regression problem with prior information that the image difference is sparse, was considered and an alternating minimization algorithm was developed to minimize an objective function which is a combination of $I$-divergence (in the data space) and $L_1$ norm (in the image space). Both noiseless data and noisy data simulations were carried out to demonstrate the capability of our algorithm to reconstruct sparse difference images. The extension of 4D CT was carried out for monoenergetic data. The proposed approach to 4D PET can jointly estimate the images corresponding to different time steps/frames, resulting in potential dose reduction in measurements and more accurate dynamic reconstructions. Moreover, analysis of the convergence rate for the EM reconstruction was carried out and validated by 2D and 3D simulated data reconstructions.

Different approaches for X-ray adaptive sensing were proposed and simulation data reconstructions were performed. Early simulation results show better image reconstruction performance in terms of normalized $L_2$ norm error in the image space compared to the non-adaptive sensing method, promising its future application in dose reduction and many other areas.

An optimization and computational approach was developed for characterizing the inner and outer bounds for the achievable rate regions for Berger-Tung coding. Some interesting problems were tested by using this computational approach and monotonic decrease of the objective function was achieved. These computed boundary regions can motivate hypothetical optimal distributions which can be tested in the first order necessary conditions for the optimal distributions.
7.2 Conclusions and Future Work

This work first explored the effect of regularization to the existing DE-AM algorithm reconstruction performances by using noiseless data, noisy data, and noisy data with scatter, generated based on the Siemens Somatom Plus 4 scanner geometry. At the same time, system bias and variance were studied by using simulated noisy data with scatter, which included 20 realizations of random data. A sequence of simulated data reconstructions were performed by using different combinations of regularization parameters, in order to find empirically good ranges for those parameters. A similar exploration of regularization parameters was also applied on real 2D data from the Philips Brilliance scanner. The regularized DE-AM algorithm can produce smoother images with appropriate choices for the regularization parameters. We observed no faster convergence when regularization was enabled. With the powerful tool of HECTARE, images were obtained with real 3D data from the Philips Brilliance scanner. Moreover, the unregularized polyenergetic AM algorithm produced smoother image compared to FDK, which promises even better reconstructions using regularization and the DE-AM algorithms. While both the unregularized DE-AM and the regularized DE-AM algorithms can produce relatively smooth images, they both suffer from slow convergence. The reason for this slow convergence was studied using a toy problem with only two parameters to estimate. Visualization of the objective function surface shows a “hill-valley” phenomenon, indicating an ill-conditioned problem setting, which is the high correlation between the estimates of two component images. The DE-AM algorithm can be potentially sped up by using a more aggressive step size when it hits the “valley,” according to our simulated results for this toy problem. However, for multi-parameter problems, it is very difficult to determine if the “valley” has been found.

The prior study of slow convergence of the DE-AM algorithm motivates the new algorithm which tries to break down the high correlations between two estimated component images. Also, it tries to avoid using normalization factors resulting from the lifting which decouples the image voxels and measurements. The idea of this new proposed algorithm, the line integral AM algorithm, is to estimate the line integrals of the two component images first, and then estimates the component images based on the estimated line integrals. One argument for doing so is to obtain fast convergence on a larger exponential family to get the line integrals first and then use the EM or other linear regression methods to obtain the images. In
order to take into account the geometry, we introduced a penalty to penalize the difference between the estimated line integrals and the forward projections of the component images. To validate the LIAM algorithm, simulated data with and without Poisson noise were used for reconstruction. We first set the penalty of the line integral and the forward projections of the component images to zero, and simply did a two-step reconstruction by using the EM algorithm after the line integrals were obtained. Fast convergence was observed and improved average estimation accuracy was achieved. In order to improve image smoothness, a neighborhood penalty was applied and different regularization parameters were studied to explore the effect on the reconstructed images. Moreover, a nonzero penalty on the line integral and the forward projections of the component images was used in conjunction with a neighborhood penalty to achieve better image smoothness while maintaining quantitatively accurate estimates. Real data from the Philips Brilliance scanner were reconstructed following the same procedure. They again showed much faster convergence compared to the DE-AM algorithms, even in the presence of very high density materials (higher than bone). For future work, a dynamic choice for the penalty of the estimated line integrals and the forward projections of the component images as well as neighborhood penalty should be explored, in order to achieve equilibrium solution at every choice stage. This procedure can minimize the bias introduced by regularization. Also, current results from real data show high noise and this can either result from inappropriate post-processing of the raw data or from lack of robustness of the algorithm itself.

Next, we considered a nonnegative linear regression problem with prior information. An alternating minimization algorithm was developed to minimize an objective function which is a combination of $I$-divergence and $L_1$ norm. There are several general convergence properties of our algorithm, including monotonic decrease of the $I$-divergence and the objective function. Simulations with both noiseless data and noisy data were carried out to demonstrate the capability of our algorithm to reconstruct a sparse difference image. A random image with a random point spread function, as well as a circularly symmetric geometry with tomographic data were considered. The algorithm performed well for the tomographic case, where the difference images were large compared to the first image. The extension of 4D CT was straightforward and promising, leading to potential applications in 4D CT, even though only monoenergetic data model for CT was considered. The proposed approach to 4D PET can jointly estimate the images corresponding to different time steps/frames, resulting in potential dose reduction in measurement and more accurate dynamic reconstruction.
Many more simulations need to be run to completely characterize this algorithm. While not reported here, preliminary results indicated better performance in the presence of Poisson noise than is achieved by LASSO. A more detailed comparison needs to be performed. We have seen perfect recovery of the difference image in some cases where the differences are large. More theoretical work is needed to quantify the threshold at which such performance is obtained. Performance as a function of the number and type of measurements of the second image needs to be quantified.

Different approaches for X-ray adaptive sensing were proposed and simulation data reconstructions were performed. Early simulation results show better image reconstruction performance in terms of normalized $L_2$ norm error, compared to non-adaptive sensing method. However, due to computational limitations, the multiple-directions method cannot be carried out for more iterations. A more computationally efficient method needs to be developed for adaptively selecting multiple directions. The better performance of adaptive sensing promises its future application in dose reduction and many other areas.

We developed an optimization and computational approach for characterizing the inner and outer bounds for the achievable rate regions for Berger-Tung coding. The optimization is based on a Lagrangian that incorporates the mutual information and distortion constraints in one cost functional. Computation can be performed by lifting the Lagrangian to a higher dimensional space through the introduction of auxiliary variables corresponding to posterior probability distributions. In addition, a probability distribution corresponding to each decoding function is included in the objective function with a damping term. Through alternately minimizing this lifted objective function over primary and auxiliary variables, a sequence of probability distributions that converge to fixed points can be obtained. Moreover, monotonic decrease of the objective function is achieved through such alternating minimization. Four two-variable examples are presented with brief interpretations in terms of encoding strategy that can achieve the bound. Each problem is presented through the sum of distortions on the encoded variables, and for three out of the four examples, the inner and outer bound regions coincide. For the problem defined by Wagner and Anantharam with a single joint distortion for the two variables, the gap they described is observed in our results. For one of the problems, computational complexity is analyzed in terms of the number of operations needed for every alternating minimization iteration. These computed boundary regions can
motivate hypothetical optimal distributions which can be tested in the first order necessary conditions for the optimal distributions.
References


Appendix A

Derivation of the Dual-Energy Alternating Minimization (DE-AM) Algorithm

The derivation presented here is an extension of the one in [1] and follows the one in [46]. As stated in Chapter 2, our goal is to minimize the following $I$-divergence over $c_i(x) \geq 0$,

$$I(d||g(y : c)) = \sum_{j=1}^{2} \sum_{y} \left( d_j(y) \ln \frac{d_j(y)}{g_j(y : c)} - d_j(y) + g_j(y : c) \right), \quad (A.1)$$

where

$$g_j(y : c) = \sum_E q_j(y, E), \quad (A.2)$$

$$\mathcal{E}_j = \left\{ q_j(y, E) = I_{0j}(y, E) \exp \left( - \sum_x \sum_i h(y|x) \mu_i(E)c_i(x) \right) , E \neq 0, \quad q_j(y, 0) = \beta_j(y) \right\}. \quad (A.3)$$

The exponential family $\mathcal{E}_j$ defines the model used for the data.

The first difficulty in solving the original objective function is that there is a summation over energies inside the “ln” denominator. We need to decouple that summation over energies and move them outside the “ln” part.
Lemma A.0.1 The $I$-divergence (A.1) may be written in the variational form

$$I(d||g(y : c)) = \min_{p_j \in L(d_j)} \sum_{j=1}^{2} I(p_j||q_j), \quad \text{(A.4)}$$

where

$$I(p_j||q_j) = \sum_y \sum_E \left( p_j(y, E) \ln \frac{p_j(y, E)}{q_j(y, E)} - p_j(y, E) + q_j(y, E) \right), \quad \text{(A.5)}$$

$$L(d_j) = \left\{ p_j(y, E) \geq 0 : \sum_E p_j(y, E) = d_j(y) \right\}. \quad \text{(A.6)}$$

The proof of this lemma is straightforward by introducing two Lagrange multipliers to enforce the equality in the definition of $L(d_j)$. Because the minimization over the two linear families are independent of each other, we can write

$$L_j = \sum_y \sum_E \left( p_j(y, E) \ln \frac{p_j(y, E)}{q_j(y, E)} - p_j(y, E) + q_j(y, E) \right) + \lambda_j(y) \left( \sum_E p_j(y, E) - d_j(y) \right) \quad \text{(A.7)}$$

Minimizing over $p_j(y, E)$ and solving for $\lambda_j(y)$ to enforce the equality in Equation (A.6) yields $p_j(y, E) = 0$ if $q_j(y, E) = 0$ (defining $I(0||0) = 0$) and if $q_j(y, E) \neq 0$

$$p_j(y, E) = d_j(y) \frac{q_j(y, E)}{\sum_{E'} q_j(y, E')}. \quad \text{(A.8)}$$

Substituting this back into the $I$-divergence yields the equality in the Lemma statement. The nonnegativity of $p_j(y, E)$ is inherent.

Therefore, we can rewrite the original maximum-likelihood estimation problem as a double minimization problem,

$$\min_{q_j \in E_j} \min_{p_j \in L(d_j)} \sum_{j=1}^{2} I(p_j||q_j), \quad \text{(A.9)}$$

subject to the inequality constraints $c_i(x) \geq 0$, for all $(i, x)$. This double minimization leads to an alternating minimization algorithm, where the iterations alternate between estimating
\( \mathbf{p}_j \in \mathcal{L}(d_j) \) and \( \mathbf{q}_j \in \mathcal{E}_j \). Note that given \( \hat{q}^{(k)}_j \), the update for \( \mathbf{p}_j \) is straightforward based on Lemma A.0.1. However, when \( \hat{p}^{(k)}_j \) is given, the update for \( \mathbf{q}_j \) is not trivial.

This first lifting just described achieved the first goal, which was to move the summation over energies outside the “\( \ln \)” in Equation (A.5).

However, there is still another difficulty lying inside the exponential term in Equation (A.3) which is a summation over all \((i,x)\). In order to tackle this, we introduce the following convex decomposition lemma.

**Lemma A.0.2** Suppose that \( f \) is a convex function defined on a convex cone \( \mathcal{D} \subset \mathbb{R}^n \). Given \( x_i \in \mathcal{D}, i = 1, 2, \ldots, \)

\[
f \left( \sum_i x_i \right) \leq \sum_i r_i f \left( \frac{1}{r_i} x_i \right) \tag{A.10}
\]

for all \( r \in \mathcal{P}, \) with \( r_i > 0 \) for all \( i \). If \( f \) is strictly convex, equality holds if and only if \((1/r_i)x_i = x\) is independent of \( i \).

By applying Lemma A.0.2 to our objective function when \( \hat{p}_j \) is given, we have

\[
\sum_j \sum_y \sum_E \sum_x \sum_i \hat{p}_j(y, E) h(y|x) \mu_i(E) c_i(x) \\
+ \sum_j \sum_y \sum_E \sum_i I_{0j}(y, E) \exp \left( - \sum_i \sum_x h(y|x) \mu_i(E) c_i(x) \right) \\
= \sum_j \sum_y \sum_E \sum_x \sum_i \hat{p}_j(y, E) h(y|x) \mu_i(E) c_i(x) \\
+ \sum_j \sum_y \sum_E \sum_i \hat{q}_j(y, E) \exp \left[ \sum_i \sum_x h(y|x) \mu_i(E) (\hat{c}_i(x) - c_i(x)) \right] \tag{A.11}
\]

\[
\leq \sum_j \sum_y \sum_E \sum_x \sum_i \left\{ \hat{p}_j(y, E) h(y|x) \mu_i(E) c_i(x) \\
+ r(x, i|y, E) \hat{q}_j(y, E) \exp \left[ \frac{h(y|x) \mu_i(E)}{r(x, i|y, E)} (\hat{c}_i(x) - c_i(x)) \right] \right\}
\]
for all \( r(x, i|y, E) > 0 \) such that
\[
\sum_x \sum_i r(x, i|y, E) \leq 1, \forall (y, E). \tag{A.12}
\]

Equality is achieved in (A.11) if
\[
\frac{h(y|x)\mu_i(E)}{r(x, i|y, E)} (\hat{c}_i(x) - c_i(x)) \tag{A.13}
\]
is only a function of \((y, E)\).

In order to choose a universal normalization factor that does not depend on \((x, i)\), we set
\[
r(x, i|y, E) = \frac{h(y|x)\mu_i(E)}{Z_i(x)}, \tag{A.14}
\]
where \(Z_i(x)\) are chosen to satisfy the constraint (A.12).

One choice of \(Z_i(x)\) can be
\[
Z_i(x) = Z_0 = \max_{y, E} \sum_x \sum_i \mu_i(E) h(y|x). \tag{A.15}
\]

After performing the second lifting on the summation of the exponential, we get a decoupled version of the objective function
\[
\sum_j \sum_y \sum_E \sum_i \sum_x \left\{ \hat{p}_j(y, E)h(y|x)\mu_i(E)c_i(x) \right. \\
+ \hat{q}_j(y, E)h(y|x)\mu_i(E) \frac{1}{Z_i(x)} \exp \left[ Z_i(x) (\hat{c}_i(x) - c_i(x)) \right] \right\}, \tag{A.16}
\]
Setting the derivative of Equation (A.16) with respect to $c_i(x)$ to 0, solving for $c_i(x)$ and imposing the nonnegativity constraint yields the update equation for the DE-AM algorithm

$$
\hat{c}_i^{(k+1)}(x) = \max \left(0, \hat{c}_i^{(k)}(x) - \frac{1}{Z_i(x)} \ln \frac{\sum_{j=1}^{2} \tilde{b}_{ij}^{(k)}(x)}{\sum_{j=1}^{2} \hat{b}_{ij}^{(k)}(x)} \right),
$$

(A.17)

where

$$
\tilde{b}_{ij}^{(k)}(x) = \sum_{y} \sum_{E} \mu_i(E) h(y|x) \hat{p}_j^{(k)}(y, E), \hat{b}_{ij}^{(k)}(x) = \sum_{y} \sum_{E} \mu_i(E) h(y|x) \hat{q}_j^{(k)}(y, E).
$$

(A.18)

We now summarize the DE-AM algorithm.

Set $k = 0$. Select an initial condition for $\hat{c}_i^{(0)}(x)$.

Compute the current estimate for the $q_j$

$$
\hat{q}_j^{(k)}(y, E) = I_{0j}(y, E) \exp \left( - \sum_x \sum_i h(y|x) \mu_i(E) \hat{c}_i^{(k)}(x) \right).
$$

(A.19)

Compute the current estimate for the $p_j$

$$
\hat{p}_j^{(k)}(y, E) = \frac{d_j(y)}{\sum_{E'} \hat{q}_j^{(k)}(y, E')},
$$

(A.20)

Compute the backprojection of the current estimate of $p_j$ and $q_j$

$$
\tilde{b}_{ij}^{(k)}(x) = \sum_{y} \sum_{E} \mu_i(E) h(y|x) \hat{p}_j^{(k)}(y, E),
$$

(A.21)

$$
\hat{b}_{ij}^{(k)}(x) = \sum_{y} \sum_{E} \mu_i(E) h(y|x) \hat{q}_j^{(k)}(y, E).
$$

(A.22)
Update the estimate of the relative partial densities

\[ \hat{c}_i^{(k+1)}(x) = \max \left( 0, \hat{c}_i^{(k)}(x) - \frac{1}{Z_i(x)} \ln \frac{\sum_{j=1}^2 \tilde{b}_{ij}^{(k)}(x)}{\sum_{j=1}^2 \bar{b}_{ij}^{(k)}(x)} \right). \] (A.23)

Set \( k = k + 1 \). Check for convergence and iterate if necessary.
Vita

Yaqi Chen

Degrees

Ph.D., Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, MO, U.S.A., August 2014
M.S., Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, MO, U.S.A., May 2012
B.S., Electrical Engineering, Tsinghua University, Beijing, China, July 2009

Publications


Publications in Preparation


August 2014