Transmission Tomography for high contrast media based on sparse data

by

Yenting Lin

A Dissertation Presented to the
FACULTY OF THE USC GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(ELECTRICAL ENGINEERING)

August 2013

Dedication

To my Mon, Hui-ing Wu, who always support me and remains on my side through this incredible journey.

To my godfather and godmother, Jhen-Chuan Gong and Yu-Ying Wu, who always keep faith on me.

Acknowledgements

I would like to thank, first and for most, to my dissertation adviser, Dr. Antonio Ortega, for his support and guidance throughout my graduate studies at the University of Southern California. I have been fortunate to work with him, who shapes my ideas about research, and moreover, refine my approach on writing and analytical thinking. I owe my gratitude to Dr. Iraj Ershaghi, for his support and many insightful suggestions from the petroleum engineering aspects to strengthen of my research. Thanks are also due to Dr. Richard Leahy for serving as members in my dissertation committee. I also needs to thank to Dr. Alex Dimikis who introduces and inspires me to work on compressed sensing, and serve as a committee member in my qualifying exam. I would like to thank to Dr. Harry Hu, who serves in my qualifying committee and valuable suggestions on medical imaging. Special thanks to Dr. Robert Scholtz, for the insightful suggestions and discussion over these years.

I would like to thank to several PhD students in Petroleum Engineering: Amir Mohammad Nejad, and Tayeb Ayatollahy Tafti. They provide valuable suggestions and do incredible work on reservoir simulations. I also need to thank to the staff of the CiSoft project for their help and continuous support for these years.

For all the colleagues in our research group Group, it is a great pressure and know and work with you. Especially, I would like to thank Kun-han Lee for unforgettable years of collaborations and for all the sharing. I would also like to thank

Sungwon Lee, Sunil Kumar, Godwin Shen and Sean McPherson, for the wonderful friendship that helps me strengthen my research and support for these years.

I would also like to thank my mother Hui-ing Wu, my godfather Jhen-Chuan Gong and godmother Yu-Ying Wu for their endless love, patience, guidance and support throughout my life. Finally, I would like to thank all my pets to accompany me for all these years.

Table of Contents

List of Tables

List of Figures

Abstract

Transmission tomography is a powerful tool to image the interior structure based on measured data on the boundary. It provides a “non-destructive” imaging and widely used in different areas. However, the reconstructed image quality degrades when lower number of data is available. In this work, we consider tomography problems involving detection of high contrast structures with limited amount of data, and propose a sparse representation for the images of our interest. The sparse model is useful to reduce the unknown variables for the tomographic reconstruction. In the first part, we propose an object-based model for high velocity structure in travel time tomography. We focus on how to speed up the path tracking procedure on high velocity object assumption. To estimate the object parameters, we use probabilistic approach to define the appearance probability of different models, and estimate the probability distribution by accelerated random walk. The result can be viewed as a “probability map”, which represent the appearance of the high velocity structure in different regions. Compared to the conventional grid-based model, our approach provides stable results and superior reconstruction quality. As an application, we use the high contrast tomography to detect fractures in a low permeability reservoir under water-flooding. In particular, we (i) propose to use the injected water as a probe signal, where we are able to measure the travel time without shutting down the usual operation, and (ii) formulate the fracture as high velocity objects and do tomographic reconstruction. The results from commercial simulator show our method can successfully identify the location of fractures, and the field experiment is consistent with the conclusion from experts in the field engineering team. In the second part, we study the X-ray transmission tomography where the object is assumed to be made from few distinct materials. This problem is called discrete tomography (DT). Such images arise in tomography problems where very high contrast is expected, e.g., in angiography medical imaging or electron tomography. We assume the image of interest can be segmented as several smooth boundary, constant intensity regions. Moreover, we design transform which can represent each region with sparse coefficients, and formulate the tomographic reconstruction as a sparse recovery. A common assumption for DT is that the set of possible intensity levels is known in advance. However, determining the prescribed intensity levels is a difficult problem, coupled with measurement calibration and the prior knowledge of image. We introduce an unsupervised DT algorithm that jointly reconstructs the image and estimates the unknown intensity levels. Our algorithm alternates between (i) an L_1 sparse recovery step with a reweighed cost function that search the sparse representation which fits the measurements, and (ii) an estimation step for the most likely intensity levels. We experimentally demonstrate that the proposed algorithm successfully estimates the unknown levels and leads to high quality reconstruction of phantom images.

Chapter 1

Introduction

1.1 Motivation

Transmission tomography aims to image the interior structure of an object based on measuring the response of the penetrating wave through it. The penetrating wave, which acts as a probe signal, can be X-ray, ultrasound or seismic wave. A set of sensors are installed on the boundary of the object to measure the signal response, which could be the attenuation or delay of the signal induced by the object. Then, we would like the reconstruct the object model, which captures the physical properties of the structure by material parameters (e.g., wave velocity, attenuation) as a function of position. In tomographic inversion, we want to match the measured data with the predicted ones from the reconstructed model. This technique provides a "non-destructive" method to image the interior of the object, and the idea can be traced back to Radon transform [21]. Although different researchers made progress on tomographic imaging [72, 82], we have to wait for the development of electronics and digital computers. In 1972, Hounsfield [38] developed the first computed tomography scanner. Since then, the transmission tomography has been widely used in many different areas, e.g., radiology [61], biology [31], geophysics [23], oceanography [55], etc.

In this dissertation, we investigate two types of transmission tomography: Travel time tomography and X-ray computed tomography. In travel time tomography, we measure the delay of signals between sensors, which give us the first arrival travel times. The travel time is the path integral on the shortest path in the slowness (reciprocal of velocity) model, where the travel path (shortest path) depends on the velocity model and may not be a straight line between sensors. Thus, with the travel time data, estimating the velocity model is a nonlinear inverse problem. This technique is very useful in modeling geophysical structures, for example, reconstruction of velocity model of the Earth and monitoring the earthquake zone. In X-ray computed tomography, series of X-ray sources are pointing to the object from different directions, and we measure the X-ray energy from the sensors on the boundary. Different from previous case, the travel path of X-ray can be assumed as a straight line, and the energy decrease is characterized by the line integral of the material attenuation coefficients. Therefore, we have a linear inverse problem when estimating the attenuation model with the X-ray energy measurements. It is widely used in medical imaging, e.g., Angiography and cancer tumor detection.

To reconstruct the object model, it is an inverse problem and might be ill-posed. But in many typical applications, for example, computed medical imaging, it is possible to increase the number of samples by putting more sensors. Thus, a sufficient number of samples can be acquired in order to reconstruct a high quality object model. Many reconstruction algorithms have been developed for this scenario, e.g., simultaneous iterative reconstruction technique (SIRT) [35], filtered back projection algorithm (FBP) [43] and total variation reconstruction [65]. It is well known that we will have better quality for reconstruction with more samples. However, in some applications, it would be too costly to increase the number of sensors significantly. In these scenarios, the measured data will be sparse and many of these reconstruction tools will not provide good quality results. For example, in seismic cross-borehole tomography [37] sensors are placed in a vertical wellbore to measure the reflection or refraction seismic waves. In such case, increasing the amount of measured data means drilling new wells to place additional sensors. Similarly, in X-ray computed tomography cardiac angiography [34], series of photon attenuations are measured by capturing X-ray that travel through the object from different angles. But taking more data means the patient is exposed to higher X-ray radiation doses, which may increase the risk of cancer. In this dissertation, the primary focus is to develop algorithms to reconstruct the object image when only sparse measured data available.

Furthermore, we consider the cases where the structure has "high contrast" property, which means that there are regions with sharp changes in the model parameters. For example, in hydraulic travel time tomography, the flow conductivity in a low permeability reservoir is very low, while the fractures provide fast pathways for the fluid. Also, in industrial computed tomography inspection, the testing material is assumed to be homogeneous and high attenuation (e.g., metal), where the cracks inside can be viewed as almost zero attenuation. The "high contrast" property and the limited availability of data makes the reconstruction more difficult and typical reconstruction algorithms may not perform well. The high contrast of the travel velocity will cause the travel paths bend severely near the high contrast regions. This violates the "almost straight travel path" assumption in most iterative linearized reconstruction algorithms and leads to lack of convergence in the solutions. For X-ray transmission tomography, the sharp changes of attenuation will cause the standard Filter-Back projection reconstruction method generates blur image near the high contrast boundary, which causes diagnostic ambiguity.

To improve the handling of this difficult situation, different from typical grid-based model, we propose an object based model to provide a "sparse" representation for high contrast regions. This is useful to (i) reduce the dimensions of unknown variables and (ii) provide a regularization for ill-posed inverse problem. In this work, we mainly focus on designing object models and the corresponding tomographic reconstruction algorithms. The algorithms in this dissertation are toward two applications: The first one is fracture estimation in a low permeability reservoir under water-flooding. In this application, estimating the location of fractures is important to reservoir characterization, production optimization, etc. We model the fractures as high velocity objects and use "injected water" as a probe signal, then estimate the fractures based on injection-production response. Compared to other methods (seismic, tracer testing), our method can be done based on current infrastructure and without additional cost. The second application is X-ray discrete tomography, where the object of interest is assumed to have several smooth boundary, constant regions. We design a dictionary to represent the object in a transformed domain and apply the "sparse recovery" technique. Thus, superior reconstructed image quality is achieved with limited amount of measured X-ray projection data. We give an overview of our work in Sections 1.2, 1.3 and 1.4 then provide an outline of this dissertation in Section 1.5.

1.2 Travel-time tomography for high contrast media with sparse data

In travel time tomography, the first arrival time between transmitters and receivers is measured and then this data is used to reconstruct the velocity model for the object, where each point in the object structure is characterized by its wave traveling velocity. The transmitted signal can be a seismic wave, an acoustic sound wave, and even a fluid pressure wave. This technique is important to characterize large scale elastic media, in applications such as seismic geophysical exploration [58] and acoustic tomography (for example, in the atmosphere's surface [77] or in the ocean layers [55]). As we mentioned before, the "high contrast media" problem means that the typical algorithms usually fail to provide a stable solution. Moreover, lack of a sufficient amount of measured data will introduce more uncertainty for the reconstructed model, because multiple velocity models may fit the measured travel time data.

Without knowing the travel path, the reconstruction of a velocity model becomes a nonlinear inverse problem. The reconstruction algorithm can be specified by (i) a *forward* step, which calculates the travel time and path, based on current estimated velocity model, (ii) an *inverse* step, which updates the velocity model based on the travel path calculated in previous step. By Fermat's principle, the travel path is the path traversed in the shortest time [11], and thus calculating the travel path is very computationally intensive. In this dissertation, we describe how to use object-based models to represent the high contrast velocity structure and propose a probabilistic approach to reconstruct the possible models with sparse data. The *novel contributions* are (i) an object representation which provides a dramatic reduction in the complexity requirement to calculate the travel path in the forward step, and (ii) development of a fast randomized algorithm to estimate the probability distribution in model parameter space by using the monotonicity property of high velocity objects.

1.3 Water-flooding Tomography

We study the fracture estimation problem in a low permeability oil reservoir under water-flooding. Due to the high demand of crude oil, enhanced oil recovery (EOR) technologies are introduced to boost the production. “Water-flooding” is one of the most widely used process to enhance the oil recovery, where water is injected in the reservoir to provide pressure support and stimulate the oil production. Water-flooding requires a careful design of the amount of injection to use at different locations because the heterogeneous structures (fracture or fault, which act as fast pathways or barriers for the fluid) will affect the recovery efficiency dramatically. Thus, to optimize the water-flood efficiency it is critical to identify these heterogeneous structures in a reservoir.

Many reservoir characterization techniques have been proposed in the past few decades, e.g., seismic cross-hole tomography [14,64], pulse testing [41,53] and tracer tests [2, 3, 24]. A common aspect of these methods is that they all require extra equipments and may disrupt normal operations. In this work, we only consider fractures in a low permeability reservoir. Thus, we can model the fractures as high velocity objects in a uniform low velocity background and apply our high contrast travel time tomography technique. Moreover, we use the injected water as a signal source and measure the change in production. By using the injection-production wells as transmitters-receivers, we can estimate the response time between them.

This will eliminate the need for the extra equipment for the reservoir characterization, therefore making possible real-time reservoir monitoring based on current infrastructure. We validate our approach by running simulations on a commercial reservoir simulator from computer modeling group LTD. [50]. A field experiment is also conducted and it shows that our result is consistent with the known fault location and the expectation from the field engineers.

1.4 Discrete X-ray Tomography

The second type of transmission tomography we investigate in this dissertation is X-ray computed tomography, which is widely used for noninvasive testing in medical imaging and industrial material inspection. During the test, a series of X-rays are sent targeting the object from different directions, and the decrease of X-ray intensity is measured around the object. It is well known that the quality of the reconstructed object image degrades if we have fewer projections angles. However, in medical imaging we want to limit the number of X-ray projections, so that it is as small as possible thus limiting the radiation dose for the patient. For example, in Angiography [39] a typical cardiac CT scan needs to put the patient under CT scan for the entire cardiac cycle, which means the patient is exposed to a high energy X-ray radiation over a long period of time (about $60 - 90$ seconds). Thus, our goal is to study the problem of improving the reconstructed image quality when a reduced amount of data is available.

In this dissertation, we consider the case where the image of interest can be segmented into several smooth boundary, high contrast regions. The testing objects are assumed to be made from few different materials, which correspond to distinct attenuation levels of X-ray radiation. For example, in an industrial CT scan for material flaw detection, we usually know the type of material compound for the object (aluminum, plastic, etc) in advance and want to test if a cavity exists. A cavity refers to empty space, which has almost zero X-ray attenuation. Thus, the reconstructed image can be approximated by a binary intensity image where the high pixel value accounts for the object and an almost zero value corresponds the cavity.

In this case, we want to recover a high contrast image where there are only few different pixel values. With a limited amount of data, the image reconstruction becomes an ill-posed integer inverse problem. This problem is called Discrete Tomography (DT) [36] and many algorithms have been proposed [36]. However, none of these algorithms has considered the sparsity in a transform domain. In recent years a significant amount of work, e.g., compressed sensing [18] has been devoted to perfectly recover a sparse signal, even though the amount of measured data is insufficient. Following this concept, we propose to use the object based model to represent the high contrast image, which leads to a sparse representation. This is done by designing a dictionary that can represent the discrete image with very few coefficients in the transformed domain. Thus, with the designed dictionary we can formulate the DT as a sparse recovery problem, and the quality of reconstruction result is superior to standard reconstruction methods.

Another challenge is that for a high contrast image, the set of possible intensity levels is not known in advance. In practice, determining the intensity levels as a prior is very challenging and affected by other practical issues, such as measurement calibration [16]. Therefore, we formulate an unknown intensity levels DT problem and propose a *completely unsupervised* algorithm to reconstruct the image. We first address the binary DT problem for known intensity levels, then extend our algorithm by representing the multi-value image with the superposition of binary images and adding a level-estimation step within each iteration. Thus, our algorithm iteratively estimates the intensity level and reconstructs the final image.

1.5 Outline

This dissertation is organized as follows. First we study the high contrast travel time tomography problem and propose an object-based reconstruction in Chapter 2. As an application, we propose the fracture estimation in a low permeability reservoir under water-flooding in Chapter 1.3. In Chapter 4 we propose a dictionary and use sparse recovery to reconstruct the object in X-ray discrete tomography. Finally, in Chapter 5 we conclude our work and point out some possible future work.

Chapter 2

Travel-time tomography for high contrast media with sparse data

2.1 Introduction

Travel time tomography aims to reconstruct an interior velocity model by using the measured first-arrival times between transmitters and receivers. The velocity model captures the physical properties of the region where the signal transmission occurs. The transmitted signal can be a seismic wave, an acoustic sound wave, and even a fluid pressure wave. This technique is important to characterize large scale elastic media, in applications such as seismic geophysical exploration [58] and acoustic tomography (for example, in the atmosphere's surface [77] or in the ocean layers [55]). However, different from many other types of transmission tomography (X-Ray CT, Positron emission tomography CT) where the straight-line travel path assumption is commonly used, in this work we consider the "high contrast" velocity model where the travel path may bend severely according to the velocity changes. This phenomenon happens in all kinds of waveform inversion applications, with one of the most studied cases being acoustic tomography. Instead of using the straight line assumption, the travel path can be characterized by Fermat's principle: the travel path observed is the path traversed in the shortest time [11], where the time can be predicted as a path integral of the slowness function (reciprocal of velocity) integrated along the travel path.

In many geophysical applications it is very common to have heterogeneous structures where the velocity contrast is high, e.g., a factor of two difference in velocity between different areas. Example of scenarios where this situation can be encountered arise in many applications: using seismic waves to find fault zones [69], monitoring the water/oil saturation in vegetable oil bio-remediation projects [47], etc. The heterogeneous structure could be either fast or slow velocity structures. In this work, we mainly focus on finding the high velocity structures in a homogeneous background. Different from the case where the velocity distribution only has small variations [77], the travel path near the boundary of high velocity structure not only bends severely but is almost dominated by them that form high transmissive channels. Because the straight line travel path approximation is not valid, the reconstruction of the velocity model becomes a nonlinear inverse problem. Note that conventional reconstruction methods are based on iterative linearized algorithms to approximate the travel path. Thus, given that in many practical situations we only have sparse measured travel time data, these methods suffer from problems due to low ray-coverage and severe path bending in the high contrast velocity medium. For a comprehensive review on this subject, we refer the reader to the overview by Berryman [12].

Our initial motivation for this work comes from the flow permeability characterization in a fractured reservoir [73]. One of the most widely used enhanced oil recovery (EOR) techniques, water-flooding, involves injecting water in a controlled manner in order to provide pressure support that can slowly sweep the oil into the production wells [75]. During this process, the permeability (measurement of ability to transmit fluid) of open fractures can be orders of magnitude higher than that of surrounding tight rocks, providing fast pathways for the flow. Thus, travel time through a fracture (which could be modeled as a line in $2D$ or a plane in $3D$) is much faster than through surrounding rocks. The flow properties of the reservoir are dominated by these highly "transmissive" fracture structures. If a fracture is close to both an injection well and a production well, most water will flow directly through the fracture and will fail to displace oil in nearby areas. This is a phenomenon known as "water cycling", which significantly reduces oil recovery efficiency. Thus, understanding the locations of fractures is critical in flow characterization and enhancement of oil recovery efficiency. The major challenge to achieve this understanding is that travel time measurements are limited by the borehole locations, which makes it difficult to reconstruct high resolution images of fracture locations. The high permeability contrast property between open fractures and surrounding rocks also increases the difficulty of reconstructing accurate velocity maps.

A similar situation also arises in bio-remediation problems, where the goal is to map the hydraulic conductivity and predict the ground water flow in the aquifer. Borehole core samples and pollutant concentration data are available only from the pumping wells, which are very sparsely located in the field. Another interesting application is network delay tomography [71], where the main goal is to identify the nodes where congestion occurs. Instead of using internal measurements at many nodes, which require high bandwidth resources, network tomography measures the end-to-end delay in the network and estimates the delay in each link. When congestion happens, the delay time may be very different from the typical values, which would make this a "high contrast" tomography problem.

Many different models have been developed for problems that involve a structured velocity distribution. Grid-based methods [10], which divide the volume into small cells and assign a constant velocity to each cell, are widely used because of the freedom to represent the structure in any degree of detail by increasing the number of cells. As an alternative, object-based methods [47] use predefined objects to represent the velocity structure. Compared to grid-based methods, if the pre-defined shape of those objects is chosen wisely, object-based methods have the advantage of parameterizing the spatial velocity distribution with a small number of objects instead of a large number of cells.

This paper focuses on finding high velocity structures (HVS) in a relatively slow homogeneous background. Specifically we consider the case where the travel velocity can only take one of several possible discrete values. We call this a "discrete" travel time tomography problem. For example, a reservoir can usually be modeled as a layered structure, where each layer has quite different hydraulic conductivity.

Only a limited amount of research has addressed the high contrast velocity problem in travel time tomography. Bai and Greenhalgh [6] proposed to add irregular nodes on the boundary of cells to improve the stability when determining nonlinear ray travel paths. Berryman [9] used Fermat's principle on the velocity model to handle the case where high contrast velocity alters the travel path severely. Both of these approaches are grid-based methods and use iterative linearized algorithms to perform the tomographic reconstruction. Because the velocity contrast is so high, grid-based models require a very fine grid to represent the velocity distribution at the boundaries between areas of different velocity, and the corresponding iterative linearized inversion algorithm will often fail to converge and determine the actual travel path. A finer grid implies we need to estimate more unknown values (the number of parameters to estimate grows linearly with the number of cells), some of cells may not even be covered by any travel path. For these cells, we cannot determine the travel velocity because no travel path passes through the corresponding cells. This is the well known “lack of ray coverage” phenomenon.

As an alternative, in this chapter, *we use object-based models to represent the HVS.* The objects are chosen from pre-defined fundamental convex shapes, so that the geometrical shape of HVS can be represented by a combination of these objects. This approach has two main advantages. First, the problem of approximating an arbitrary shape by multiple convex objects is well understood. Moreover, we can incorporate the prior information about the HVS into object-based models. For instance, in the fracture characterization problem we know that the geometrical shape of fractures can be well approximated by lines in $2D$ models or planes in $3D$ models. Thus, in these cases, we can define the shape of our fundamental convex object as a “line” or a “plane”, respectively. With this formulation, only small number of objects is needed to well approximate the shape of HVS, which reduces the dimensionality and uncertainty of the reconstruction problem. Second, the travel path tracking procedure can be simplified by only considering the shortest path between objects instead of all cells in spatial domain. Because the elementary objects are convex, the travel path is piecewise linear and the computational complexity for finding the shortest path between objects is reduced. Compared to the methods we mentioned before, our approach reduces the number of unknown variables (which is equal to the number of object parameters) and avoids the “lack of path coverage” problem arising in grid-based models, which leads to a more stable solution to the problem.

To estimate the velocity model from the measured travel time, we need to solve a nonlinear inverse problem. The major issue in all inverse problems is that the solution, in this case the estimated velocity model, may not be unique for the given measurements. One popular approach to handle the non-uniqueness issue is to apply regularizations to favor certain properties in the model [28]. The regularization methods can be viewed as model selection: they will lead to solutions that balance data-fitting and model-penalty. However, it is not easy to select the weight for model-penalty and this usually requires cross validation in order to avoid over-fitting [57].

An alternative approach is to estimate the probability distribution for the model space according to the data-fitting [68]. This gives a full description of the relative probabilities of the different models, so that we can explicitly consider all likely solutions. However, estimating the probability density in a high dimensional model space is very challenging and time consuming [52].

In this chapter, we choose the Bayesian approach and focus on estimating the probability density in the model parameter space. Due to the large dimension of the model parameter space, we develop an accelerated random walk algorithm to explore the model space. Based on the Hamiltonian Monte Carlo method (HMC), we add an additional friction term to the generation of sampling points along the simulated particle moving trajectory through the model parameter space. Our algorithm will sample more frequently locally in low error regions, and re-sampling will be used based on the HVS property to approximate the structure of the probability distribution. The results are presented as a probability map showing where the high velocity objects are more likely to appear in the underlying structure of the system.

To the best of our knowledge, we are the first to use an object-based approach to solve a high contrast, discrete velocity tomography problem. Our proposed algorithm uses the HVS properties to simplify the path finding step, which makes the HMC sampling process more efficient. Furthermore, we exploit the monotonicity of the travel time as a function of high velocity object size to approximate the probability distribution in the space of model parameters. The rest of the paper is organized as follows: We define the object based model to represent the structure in the velocity model in Section 2.2, and introduce the forward step and define the mathematical formulation for the travel path finding problem in Section 2.3. In Section 2.4 we give an overview of our proposed algorithm for probability density estimation. Simulation results and conclusions are presented in Section 2.5 and 2.6.

2.2 Object based model

In this chapter, we addresses the problem of detecting high velocity structures in low velocity background with sparse data. The relation between the travel time and velocity model is illustrated in Figure 2.1, where the forward step predict the travel time for a given velocity model. In the inverse step, a velocity model is estimated to match the measured travel time data. In this section, we will introduce the object based approach to represent the high contrast velocity model.

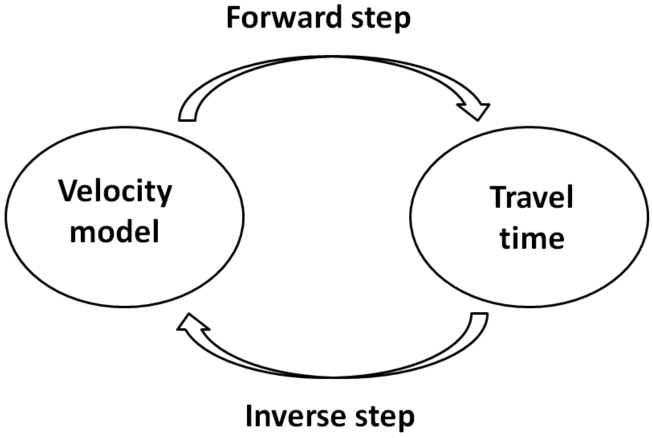
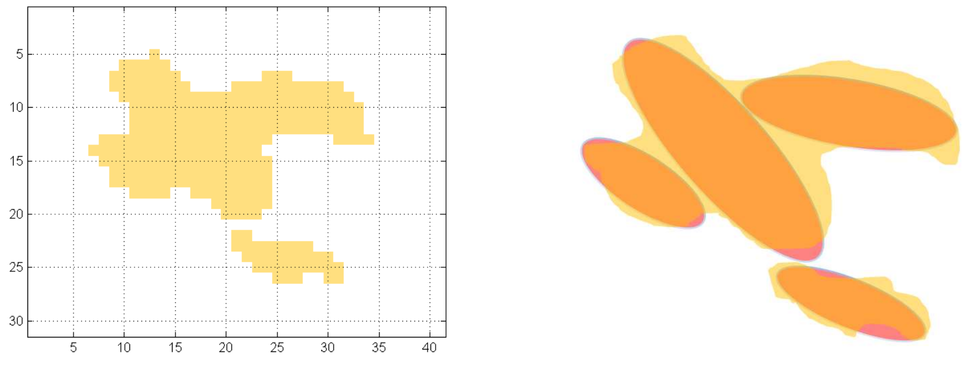


Figure 2.1: The forward-inverse step of the velocity model and travel time

To represent the high velocity structure with the object based model, we need to define the type of object that is appropriate for our application. In rendering (computer graphics), it has been well studied that we can approximate an arbitrary structure in any detail with few fundamental objects [4], for example, triangle or ellipse. In this chapter, we use convex objects as our fundamental objects, where we can define the types of convex object shape by user's choice. For example, if we only use one type convex object (ellipse) as the fundamental object, the ith object will be parameterized by the scale (size of object), center location and the orientation of the corresponding ellipse, leading to a vector of parameters $\boldsymbol{\theta}_i = \{s_i, \mathbf{c}_i, \psi_i\}$, where $s_i, \mathbf{c}_i, \psi_i$ denote the scale, center location and the orientation respectively (see Figure 2.2(b)). We denote $|\boldsymbol{\theta}_i|$ the area inside the ith object. This formulation can be easily extended to multiple type of convex shapes, for example, the object can be either a triangle or an ellipse with one additional parameter to identify the type of shape.



(a)

(b)

Figure 2.2: (a) Grid based model: the HVS structure is approximated with cells and assigns the same velocity inside each cell. (b) Object based model with ellipse as the fundamental object: the structure is approximated with objects and assigns the same velocity inside each ellipse.

Next we define the velocity model in terms of the objects. Let $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_N\}$ be the set of N high velocity objects, we denote v_h the velocity in the homogeneous background, and v_i the velocity inside object $\boldsymbol{\theta}_i$. Thus, the velocity model can be represented by the objects as:

$$V(x,y) = \begin{cases} v_i, & \text{if } (x,y) \in |\boldsymbol{\theta}_i|, \\ v_h, & \quad \text{otherwise.} \end{cases} \tag{2.1}$$

Obviously the spatial velocity distribution is determined by the objects. We use the notation $V(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N)$ to indicate the velocity distribution when we have N objects with parameters $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N$ in the model.

We use $d(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$ to represent the distance between two objects, which is defined as:

$$d(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) = \min_{\boldsymbol{\mu},\boldsymbol{\nu}} \|\boldsymbol{\mu} - \boldsymbol{\nu}\|_2,\ \boldsymbol{\mu} \in |\boldsymbol{\theta}_i|,\ \boldsymbol{\nu} \in |\boldsymbol{\theta}_j| \tag{2.2}$$

and the corresponding path is denoted by $\vec{P}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$. The same notation can be used for the distance between point and object, or point to point, i.e., $d(\boldsymbol{\alpha}, \boldsymbol{\beta})$ and $\vec{P}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ are the distance and path between points $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, etc.

2.3 Forward step

In this section, we introduce the forward step: Given the velocity model as input, the forward model predict the corresponding travel time between arbitrary transmitters and receivers. We use the mathematical formulation proposed in [12] to calculate the travel time, where the travel path is defined as the direction of wavefront propagation. Based on Fermat's principle, the actual travel path is the one with minimum time cost from all possible travel paths connecting the transmitter and the receiver. Let $V(x, y)$ represent the velocity at position (x, y). The time cost of an arbitrary path P connecting two points $\{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$ based on velocity model V is defined by the path integral:

$$\tau^P(V, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \int_P \frac{1}{V(x, y)} \, dl^P, \text{ with } \; P_{start} = \boldsymbol{\alpha}, \quad P_{end} = \boldsymbol{\beta}. \tag{2.3}$$

The travel path, P^*, is defined as the path with minimum time cost τ^*. Therefore, we can define the travel time τ^* between two points $\{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$ as:

$$\tau^*(V, \boldsymbol{\alpha}, \boldsymbol{\beta}) \;=\; \min_P \tau^P(V, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \min_P \int_P \frac{1}{V(x, y)} \, dl^P, \tag{2.4}$$

and the path P^* will be:

$$P^*(V, \boldsymbol{\alpha}, \boldsymbol{\beta}) \;=\; \arg\min_P \; \tau^P(V, \boldsymbol{\alpha}, \boldsymbol{\beta}). \tag{2.5}$$

Finding the analytical solution for the travel time τ^* and travel path P^* is a classical problem in calculus of variations [20]. Conventional methods, including the shotgun ray-tracing method [12] or the level set method [63], tend to be all very computationally expensive. In what follows, we will show that simple solutions exist if we consider the case of a homogeneous background containing high velocity convex objects.

2.3.1 Fast travel time/path finding

With the object based velocity model we proposed in Section 2.2, we now propose a fast travel path finding algorithm and prove it can be built using induction on the number of objects included in the model. For now, in order to reduce the complexity, we assume that all objects have the same velocity and consider the high contrast velocity case, i.e., $v = v_1 = \cdots = v_N$ and velocity ratio $v/v_h \to \infty$. With the high velocity contrast assumption, we can ignore the time spent passing through an object, which provides a way to find the fastest travel path by considering the path between objects recursively, thus greatly reducing the computational complexity.

We build our path-finding algorithm by induction. We start by assuming there is only one object in the velocity model, $V(\boldsymbol{\theta}_1)$, then the travel path between a transmitter $\boldsymbol{\alpha}$ and a receiver $\boldsymbol{\beta}$ will be either the direct path connecting transmitter-receiver points $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ or the one passing through the object, whichever is faster (see Fig. 2.3(a)). Thus, the travel time function for any two points $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ with a single object $\boldsymbol{\theta}_1$ in the model is the fastest of two possible travel paths:

$$\tau^*(V(\boldsymbol{\theta}_1), \boldsymbol{\alpha}, \boldsymbol{\beta}) = \min \begin{cases} 1/v_h \cdot d(\boldsymbol{\alpha}, \boldsymbol{\beta}) \\ 1/v_h \cdot (\ d(\boldsymbol{\alpha}, \boldsymbol{\theta}_1) + d(\boldsymbol{\theta}_1, \boldsymbol{\beta})\ ) + 1/v_1 \cdot d(\boldsymbol{\gamma}, \boldsymbol{\zeta}) \end{cases} \tag{2.6}$$

where $\boldsymbol{\gamma}, \boldsymbol{\zeta}$ are the closest points to $\boldsymbol{\alpha}, \boldsymbol{\beta}$ in $|\boldsymbol{\theta}_1|$, which are defined as:

$$\boldsymbol{\gamma} = \arg\min_{\boldsymbol{\gamma}} \|\alpha - \boldsymbol{\gamma}\|,\ \boldsymbol{\zeta} = \arg\min_{\boldsymbol{\zeta}} \|\boldsymbol{\beta} - \boldsymbol{\zeta}\|,\ \boldsymbol{\gamma}, \boldsymbol{\zeta} \in |\boldsymbol{\theta}_1|. \tag{2.7}$$

The travel path passing through the object is the combination of the shortest distance from $\boldsymbol{\alpha}$ to $|\boldsymbol{\theta}_1|$, a path inside $|\boldsymbol{\theta}_1|$ and the shortest distance from $|\boldsymbol{\theta}_1|$ to $\boldsymbol{\beta}$. Because of the background is homogeneous, the shortest distance from $\boldsymbol{\alpha}$ to $|\boldsymbol{\theta}_1|$ is a straight line and we use $\vec{P}(\boldsymbol{\alpha}, \boldsymbol{\gamma})$ to represent it (likewise for $\vec{P}(\boldsymbol{\zeta}, \boldsymbol{\beta})$). And the shortest path inside $|\boldsymbol{\theta}_1|$ connecting $\boldsymbol{\gamma}$ and $\boldsymbol{\zeta}$ is also a straight line, because the object is convex shape. This travel path can be viewed as $\boldsymbol{\alpha} - \boldsymbol{\theta}_1 - \boldsymbol{\beta}$ which describes the order of objects passing through. From the high velocity object assumption, the travel time inside $|\boldsymbol{\theta}_1|$ is negligible, thus, the travel time function in the one-object case becomes:

$$\tau^*(V(\boldsymbol{\theta}_1), \boldsymbol{\alpha}, \boldsymbol{\beta}) = \min \begin{cases} 1/v_h \cdot d(\boldsymbol{\alpha}, \boldsymbol{\beta}) \\ 1/v_h \cdot (\ d(\boldsymbol{\alpha}, \boldsymbol{\theta}_1) + d(\boldsymbol{\theta}_1, \boldsymbol{\beta})\ ) \end{cases} . \tag{2.8}$$

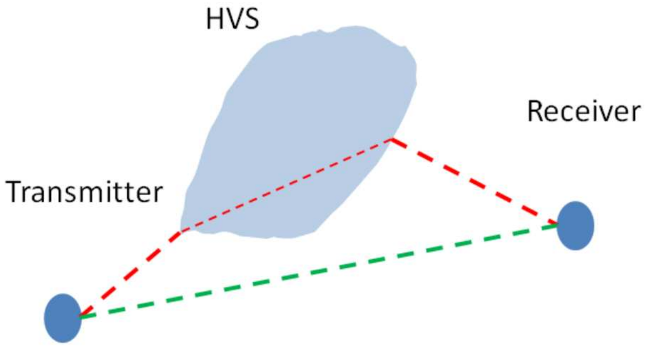


Figure 2.3: The travel path will be the faster one between the direct path or the one through high velocity object

Next, we consider the two-object case, $N = 2$. Now the fastest travel path will be either the path using less than two objects, or the path passing through both objects. From (2.8) we know how to compute the travel path/time for the one-object cases, which are $\boldsymbol{\alpha} - \boldsymbol{\theta}_1 - \boldsymbol{\beta}$ and $\boldsymbol{\alpha} - \boldsymbol{\theta}_2 - \boldsymbol{\beta}$. Thus, all we need to do now is to compute two new paths which pass through both $|\boldsymbol{\theta}_1|$ and $|\boldsymbol{\theta}_2|$, and compare them to the previous results.

We notice that the path $\boldsymbol{\alpha} - \boldsymbol{\theta}_1 - \boldsymbol{\theta}_2 - \boldsymbol{\beta}$ includes the shortest paths from $\boldsymbol{\alpha}$ to object $\boldsymbol{\theta}_1$, from object $\boldsymbol{\theta}_1$ to object $\boldsymbol{\theta}_2$, and from object $\boldsymbol{\theta}_2$ to $\boldsymbol{\beta}$. Similar to (2.8), the corresponding travel time will be $1/v_h \cdot (\ d(\boldsymbol{\alpha}, \boldsymbol{\theta}_1) + d(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) + d(\boldsymbol{\theta}_2, \boldsymbol{\beta})\ )$. But $d(\boldsymbol{\alpha}, \boldsymbol{\theta}_1)$ has been calculated in the previous step when finding the path $\boldsymbol{\alpha} - \boldsymbol{\theta}_1 - \boldsymbol{\beta}$, and likewise $d(\boldsymbol{\theta}_2, \boldsymbol{\beta})$ is also available. Thus $d(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ is the only new quantity to be calculated. Now it is clear that the travel path $\boldsymbol{\alpha} - \boldsymbol{\theta}_1 - \boldsymbol{\theta}_2 - \boldsymbol{\beta}$ can be found efficiently based on previous results. Similarly for $\boldsymbol{\alpha} - \boldsymbol{\theta}_2 - \boldsymbol{\theta}_1 - \boldsymbol{\beta}$, only $d(\boldsymbol{\theta}_2, \boldsymbol{\theta}_1)$ would be needed, but $d(\boldsymbol{\theta}_2, \boldsymbol{\theta}_1) = d(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ and thus no new distance needs to be computed.

Now the forward model for calculating the travel time given two objects in velocity model $\tau^*(V(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2), \boldsymbol{\alpha}, \boldsymbol{\beta})$ can be simplified as

$$\tau^*(V(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2), \boldsymbol{\alpha}, \boldsymbol{\beta}) = 1/v_h \cdot \min \begin{cases} d(\boldsymbol{\alpha}, \boldsymbol{\beta}) \\ d(\boldsymbol{\alpha}, \boldsymbol{\theta}_1) + d(\boldsymbol{\theta}_1, \boldsymbol{\beta}) \\ d(\boldsymbol{\alpha}, \boldsymbol{\theta}_2) + d(\boldsymbol{\theta}_2, \boldsymbol{\beta}) \\ d(\boldsymbol{\alpha}, \boldsymbol{\theta}_1) + d(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) + d(\boldsymbol{\theta}_2, \boldsymbol{\beta}) \\ d(\boldsymbol{\alpha}, \boldsymbol{\theta}_2) + d(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) + d(\boldsymbol{\theta}_1, \boldsymbol{\beta}) \end{cases} . \qquad (2.9)$$

Note that compared to the single object case in (2.8), in (2.9) we only need to compute three new terms $d(\boldsymbol{\alpha}, \boldsymbol{\theta}_2)$, $d(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ and $d(\boldsymbol{\theta}_2, \boldsymbol{\beta})$. By induction, it follows that when we add the Nth object $\boldsymbol{\theta}_N$ the new terms that need to be computed will be $d(\boldsymbol{\alpha}, \boldsymbol{\theta}_N)$, $d(\boldsymbol{\theta}_1, \boldsymbol{\theta}_N), \ldots, d(\boldsymbol{\theta}_{N-1}, \boldsymbol{\theta}_N)$ and $d(\boldsymbol{\theta}_N, \boldsymbol{\beta})$. That is, the number of new distances to be computed increases linearly with the number of objects. Therefore, the overall computational complexity of path tracking becomes $O(N^2)$.

2.3.2 Dijkstra path finding

Note that in (2.9), as well as in successive cases with additional objects, $N > 2$, the optimal result is obtained by comparing different combinations of pairwise distance (between points and/or objects). Based on this fact, we can convert the path tracking problem into a shortest distance problem on a graph.

To do so, we can define a graph where the transmitter and receiver act as source and destination vertices and the objects are intermediate vertices. The weight of edges between the vertices is the distance between the corresponding objects defined in (2.2) (sources, destinations or objects, see Figure 2.4). Thus, this undirected graph $G = (v, e)$ will be fully connected with nonnegative weights and the shortest path from source to destination can be found by running the Dijkstra algorithm [22] (see Algorithm 1).

Algorithm 1 Dijkstra algorithm for path tracking

```
for $v \in G$ do                                          ▷ Initialization
    dist[$v$] = $\infty$                      ▷ Unknown distance from source to $v$
    previous[$v$] = $\emptyset$            ▷ Previous node in optimal path from source
    dist[source] = 0                          ▷ Distance from source to source
    $\mathbf{Q} := \forall v \in G$                 ▷ Put all nodes in $\mathbf{Q}$ to be scanned
end for
while $\mathbf{Q} \neq \emptyset$ do                               ▷ The main update loop
    $u := v \in \mathbf{Q}$ with minimum dist[$v$]               ▷ Start node in first case
    $\mathbf{Q} = \mathbf{Q} \setminus u$                                    ▷ remove $u$ from $\mathbf{Q}$
    if dist[$u$] = $\infty$ then
        break ;             ▷ all remaining vertexes are inaccessible from source
    else
        for $\forall$ neighbor $v$ of $u$ do                      ▷ where $v$ is still in $\mathbf{Q}$
            alt = dist[$u$] + $e(u, v)$
            if alt $\leq$ dist[$v$] then                   ▷ update the distance for v
                dist[$v$] = alt
                previous[$v$] = $u$
            end if
        end for
    end if
end while
```

For example, in Figure 2.4b we can construct the corresponding fully connected graph for velocity model $V(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ with 4 nodes as shown in Figure 2.4(b). We show the update of the path tracking algorithm from the source α to each node

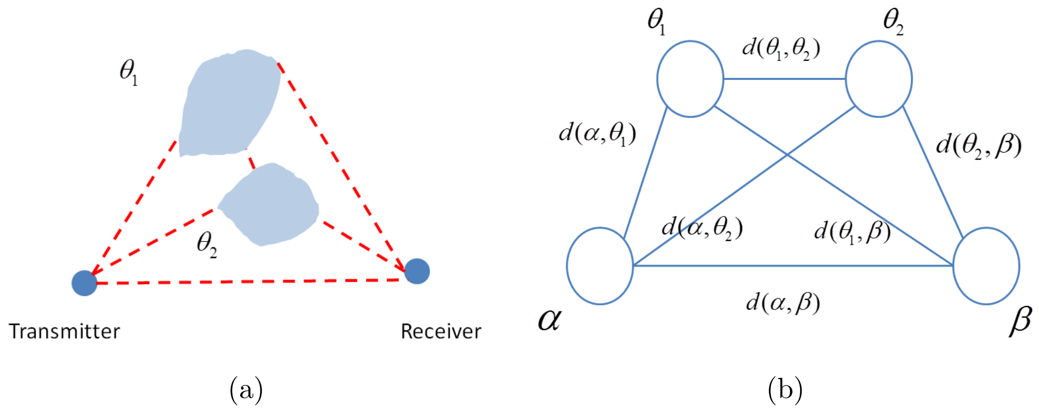


Figure 2.4: Graph representation of path tracking. (a) The distance between objects (b) Graph and distance metric

for a numerical example with the same topology as the example of Figure 2.4(b) in Figure 2.5.

2.3.3 Relationship between object size and travel time

For high velocity convex objects, we note that there is a monotonicity property between the travel time function and the size of the objects. This will play an important role in the inverse step to be presented later. To see this, we assume two high velocity convex objects, $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_1^{'}\}$, where $|\boldsymbol{\theta}_1^{'}|$ is a dilation of $|\boldsymbol{\theta}_1|$. In other words, $\boldsymbol{\theta}_1^{'}$ is a convex object with larger size but with the same overall shape as $\boldsymbol{\theta}_1$, thus, $|\boldsymbol{\theta}_1|$ is a subset of $|\boldsymbol{\theta}_1^{'}|$, $|\boldsymbol{\theta}_1| \subset |\boldsymbol{\theta}_1^{'}|$ (as shown in Figure 2.6).

From previous discussion, the travel path depends on the distance between the objects. If one object expands, the distance between this object and others must be shorter. Thus, all the edge weights (distance metric) in the graph will be smaller than or equal to the weights before expansion, thus, the travel time must be faster. This leads to the following lemma:

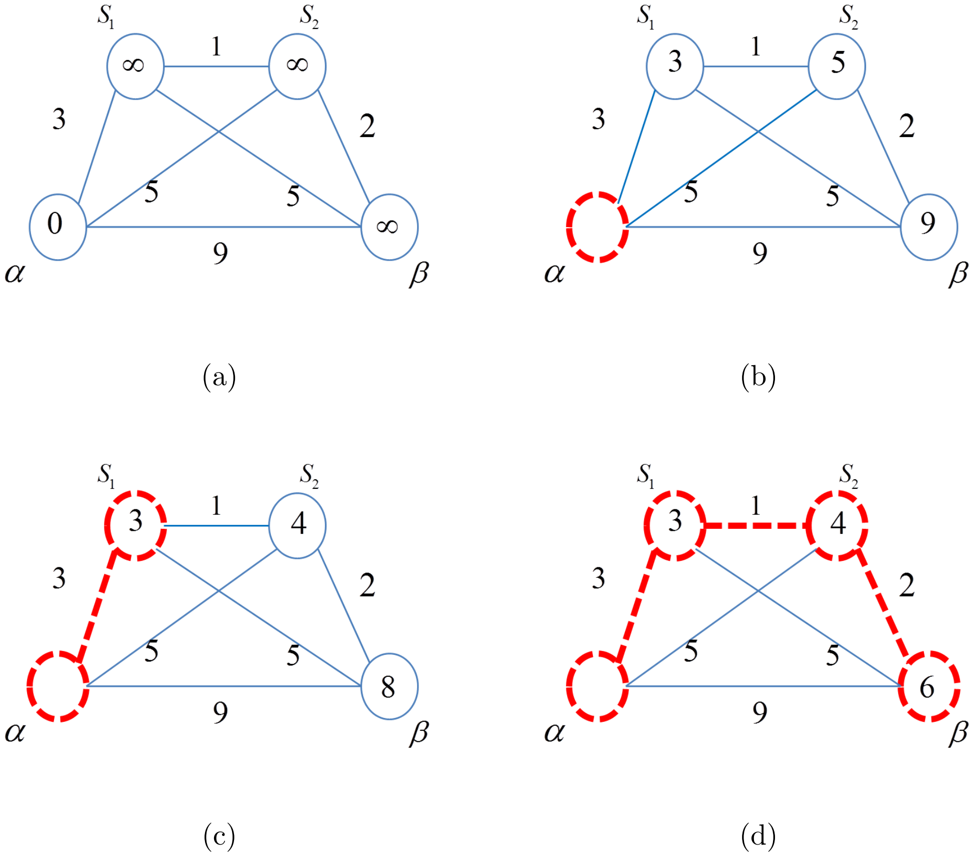


Figure 2.5: The example for Dijkstra algorithm. Note in (b) the dist[$\boldsymbol{\theta}_2$] is dist[$\boldsymbol{\alpha}$]$+$ $e(\boldsymbol{\alpha}, \boldsymbol{\theta}_2) = 5$, and in (c) after we add $\boldsymbol{\theta}_1$ the dist[$\boldsymbol{\theta}_2$] is updated to dist[$\boldsymbol{\theta}_1$] $+$ $e(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = 4$

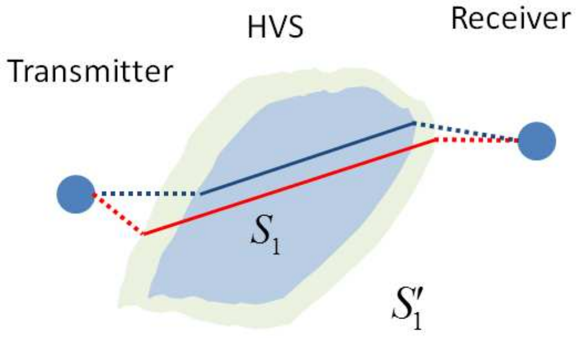


Figure 2.6: Change of the travel path with respect to the object size

Lemma 1. *Consider two high velocity objects, $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}^{'}_1\}$, where $|\boldsymbol{\theta}^{'}_1|$ is a dilation of $|\boldsymbol{\theta}_1|$, $|\boldsymbol{\theta}_1| \subset |\boldsymbol{\theta}^{'}_1|$. With the other $N-1$ objects fixed, consider two alternative velocity models $V(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_N)$ and $V(\boldsymbol{\theta}^{'}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_N)$ using $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}^{'}_1$ respectively. For the travel time function, we have $\tau^*(V(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N), \boldsymbol{\alpha}, \boldsymbol{\beta}) \geq \tau^*(V(\boldsymbol{\theta}^{'}_1, \ldots, \boldsymbol{\theta}_N), \boldsymbol{\alpha}, \boldsymbol{\beta})$ $\forall \boldsymbol{\alpha}, \boldsymbol{\beta}$. Thus, the travel time is monotonically non-increasing with respect to the size of high velocity object.* [1]

Proof. The proof is straight forward, given that weights in the graph are smaller or equal with the same topology, the shortest path will be shorter. □

2.4 Inverse step

In the forward step we just presented, we can predict the travel time when the velocity model is given. Then, in the inverse step the goal is to estimate the velocity model when the travel time data is observed. With limited travel time measured data, this inverse problem becomes ill-posed and the solution may not be unique. Thus, searching one single most possible model provides limited information for

[1] This property still holds for any even $|\boldsymbol{\theta}_1| \in |\boldsymbol{\theta}^{'}_1|$, even $|\boldsymbol{\theta}_1|$ and $|\boldsymbol{\theta}^{'}_1|$ have different shapes.

this inverse problem. Instead, we formulate it as a statistical inference problem and estimate the probability distribution of the velocity model in the parameter space [68]. We start this section by introducing some notations.

2.4.1 Notations

The input data is obtained by measuring the travel time between the set of transmitters $\mathcal{A} = \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{tx}\}$ and receivers $\mathcal{B} = \{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{rx}\}$. We denote the measured travel time for all transmitter-receiver pairs $(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_j)$ as a vector $\mathbf{t} = \{t_1, \ldots, t_n\}$, where $n = tx \cdot rx$. Assuming that there are at most N objects in the velocity model, we can cascade all object parameters and define a vector of model parameters, $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N\}$, thus the velocity model $V(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N)$ can be represented simply by $V(\boldsymbol{\Theta})$. Then we define the travel time function from the forward model, $\mathbf{T}(\boldsymbol{\Theta}, \mathcal{A}, \mathcal{B})$, as a vector function representing the travel time between each pair of transmitters and receivers based on the velocity model with parameter $\boldsymbol{\Theta}$:

$$\mathbf{T}(\boldsymbol{\Theta}, \mathcal{A}, \mathcal{B}) = (T_1(\boldsymbol{\Theta}, \mathcal{A}, \mathcal{B}), \ldots, T_n(\boldsymbol{\Theta}, \mathcal{A}, \mathcal{B})), \tag{2.10}$$

where

$$\begin{cases} T_1(\boldsymbol{\Theta}, \mathcal{A}, \mathcal{B}) & = & \tau^*(V(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N), \boldsymbol{\alpha}_1, \boldsymbol{\beta}_1) \\ & \vdots & \\ T_n(\boldsymbol{\Theta}, \mathcal{A}, \mathcal{B}) & = & \tau^*(V(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N), \boldsymbol{\alpha}_{tx}, \boldsymbol{\beta}_{rx}). \end{cases} \tag{2.11}$$

We then define an error function as a quadratic data-fitting error between the travel time predicted from the forward model and the measured travel time:

$$E(\boldsymbol{\Theta}) = \|\mathbf{t} - \mathbf{T}(\boldsymbol{\Theta}, \mathcal{A}, \mathcal{B})\|^2. \tag{2.12}$$

We use the Bayesian approach [68], which leads us to update the belief for different models after accounting for the observations. The likelihood function $L(\mathbf{\Theta})$, which is modeled as a Gaussian uncertainty of the error, measures the confidence on different models:

$$L(\mathbf{\Theta}) = \tilde{k} \cdot e^{-E(\mathbf{\Theta})}, \tag{2.13}$$

where $\tilde{k}$ is a normalization constant. From Bayes' rule, the posterior probability density function (PDF) $\sigma(\mathbf{\Theta})$ is proportional to the prior probability distribution $\rho(\mathbf{\Theta})$ multiplied by the likelihood function $L(\mathbf{\Theta})$. The prior probability $\rho(\mathbf{\Theta})$, which may come from previous experience, can provide useful information to select possible models after the data is observed:

$$\sigma(\mathbf{\Theta}) = k \cdot \rho(\mathbf{\Theta}) L(\mathbf{\Theta}), \tag{2.14}$$

where k is again a normalization constant. For the rest of this paper, we assume a uniform distribution for the prior probability $\rho(\mathbf{\Theta})$, so that the posterior PDF is equal to the likelihood function. Estimate the posterior PDF is equivalent to estimating the error function, where the most possible models correspond to the global minima in the error function $E(\mathbf{\Theta})$. In the next section, we will introduce our algorithm to estimate the error function.

2.4.2 Proposed algorithm

Because the travel time function is nonlinear, the error function $E(\mathbf{\Theta})$ will have a complex and multi-modal shape. We can "sample" the error function at any point $\mathbf{\Theta}$ by calculating the mismatch between the measured travel times and the ones predicted from the forward model based on parameter $\mathbf{\Theta}$. However, even with the fast path tracking approach of Algorithm 1, calculating the travel time is still very computationally expensive.

A naive approach to estimate the error function would be running a brute-force uniform sampling in the whole parameter space. For example, consider the 2D case where each object needs one parameter for the size, two parameters for the center location and one for the rotation angle. If we choose 3 objects in the velocity model, there will be 12 parameters, so that if we uniformly sample 10 possible values for each parameter, this will require a total of 10^{12} samples, which clearly makes this uniform sampling approach impractical.

We note that in (2.13) the high probability models correspond to the regions with low error in the parameter space. Thus, when sampling the error function, we would like to have more sample points in these low error regions. However, the error function is multi-modal and the gradient based method (steepest descend) can only search and sample near the closest local minima. Thus, in order to search and sample in the whole parameter space, we choose a random walk sampling scheme. While this is a popular approach to find multiple minima, the main drawback is that its computational time could be very high, especially when the dimension of the parameter space is high [52]. Thus, when we consider a velocity model with many objects, the dimension of the model parameter space grows with the number of objects which implies an exponential growth in the number of samples. To overcome this problem, *we propose an accelerated sampling algorithm to speed up the random walk sampling and achieve denser sampling in the low error regions.*

To further accelerate the sampling, we make use of known properties of the error function. Specifically, in the second part of our algorithm we use the monotonicity property in Lemma 1 which implies that each error function (corresponding to one measured travel time) is a unimodal function with respect to changes in the size of one object. This allows us to re-sample along the dimension corresponding to the object size (with all other parameters fixed). These re-sampling location are chosen by running golden section search on each error function (which is unimodal). Compared to other sampling methods, our method significantly reduces the computational time and provides a sufficiently good approximation of the error function. The two parts of our algorithm are described in the next section.

2.4.2.1 Accelerated random walk sampling

In the first part of our algorithm, we want to sample the error function and emphasize the sampling in the low error regions. We modify the “Hamiltonian Monte Carlo” (HMC) method [25], which is a Metropolis method but includes the gradient information to reduce the random walk behavior. HMC uses the dynamical system concept to draw samples by simulating a particle movement on the surface of the error function. We introduce a new “friction” term, which is closely related to the cooling schedule in simulated annealing [45], to draw more samples near the local minima.

In HMC, we define a dynamical system where the model parameter $\mathbf{\Theta}$ is augmented by a momentum variable $\mathbf{p}$, where $\mathbf{p}, \mathbf{\Theta}$ have the same size. The total energy $H(\mathbf{\Theta}, \mathbf{p})$ of the dynamical system is defined as the sum of “kinetic energy” and the “potential energy”, where the “potential energy” is equal to the error function $E(\mathbf{\Theta})$ and the “kinetic energy” is given by $K(\mathbf{p}) = \|\mathbf{p}\|^2/2$, i.e, $H(\mathbf{\Theta}, \mathbf{p}) = E(\mathbf{\Theta}) + K(\mathbf{p})$. The changes in $\mathbf{\Theta}$ and $\mathbf{p}$ will be determined by the following equations:

$$\dot{\mathbf{\Theta}} = \mathbf{p}, \tag{2.15}$$

$$\dot{\mathbf{p}} = -\frac{\partial E(\mathbf{\Theta})}{\partial \mathbf{\Theta}} - \epsilon \mathbf{p}. \tag{2.16}$$

To sample the error function, we simulate and record the states of particle movement. With a randomized momentum $\mathbf{p}$, we solve the Hamiltonian dynamics in (2.15) during a simulated time of duration t. In the dynamical movement, the momentum variable $\mathbf{p}$ determines where the parameter $\mathbf{\Theta}$ goes, and the gradient of potential function determines the change of momentum $\mathbf{p}$. The friction term $\epsilon\mathbf{p}$ decides the loss of total system energy.

Starting with the initial value $\mathbf{\Theta_0}$, we define the simulation time t and steps Δt, then use (2.15) to identify the successive steps in the walk through parameter space. During the simulated time t, we can record the state variables $[\mathbf{\Theta}(\Delta t), \mathbf{p}(\Delta t)]$, $[\mathbf{\Theta}(2\Delta t), \mathbf{p}(2\Delta t)]$, $\ldots$, $[\mathbf{\Theta}(t), \mathbf{p}(t)]$ which describes the variable movement on the error function. Then we take all the $\mathbf{\Theta}(\Delta t), \ldots, \mathbf{\Theta}(t)$ as new sampling points for the error function.

In Fig 2.7 we show an example of the particle movement. The initial momentum drives the particle to the high potential region, then it falls back because the momentum is changed by the gradient of the potential function. And the total energy decreases during the simulation, which causes the particle to settle down in one local minima.

Then we decide whether to accept the last sampling point $\mathbf{\Theta}(t)$ as a new starting point for the next round of simulation by the Metropolis rule [54]. This acceptance rule is based on the change of the error function, $\Delta E(\mathbf{\Theta}) = E(\mathbf{\Theta}(t)) - E(\mathbf{\Theta}(0))$, where $\mathbf{\Theta}(0)$ is the current starting point. If $\Delta E(\mathbf{\Theta}) \leq 0$, the new sample reaches a lower error state and we always accept it as a new starting point. Otherwise, we draw a random number r between $[0, 1]$ and accept it if $\exp(-\Delta E(\mathbf{\Theta})) \geq r$. We iteratively simulate this dynamic system L times, and the detail algorithm is shown in Algorithm 2.

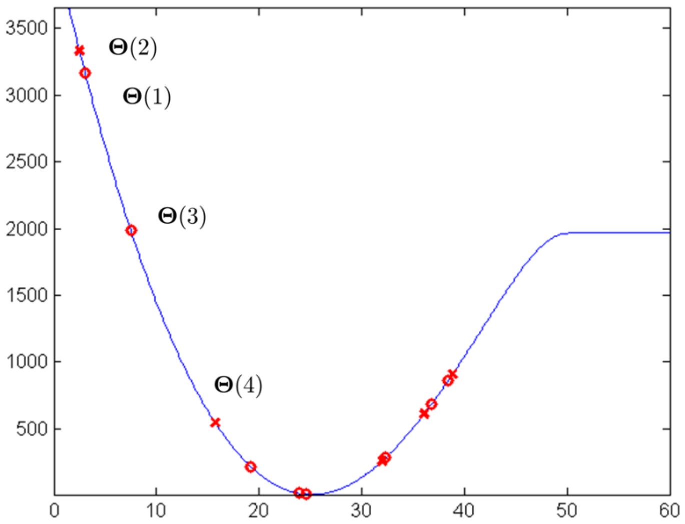


Figure 2.7: The simulated state movement. Note the initial momentum is toward left, driving to the high error region. Then it falls into low error regions, and we can see the total energy is decreasing through the simulation. Most of samples are near local minimum.

This sampling scheme provides some useful properties for exploring the parameter space in our inverse problem:

- The random momentum provides the ability to jump out of current local minima, and makes it possible to travel through all parameter space and sample in multiple low error regions.

- The additional friction term decreases the total system energy at each simulation proceeds, which "cools down" the system and allows the state to stay near a local minimum. This property leads to more samples in the low error region.

- The exploration speed is linearly related to the number of iterations, instead of being related to its square root, as in the typical random walk. This makes

Algorithm 2 Modified Hamiltonian Monte Carlo method

```
$\mathbf{\Theta}(0) = \mathbf{\Theta}_{init}$    ▷ Initialize
**for** $l = 1 : L$ **do**    ▷ loop L times
    $g = \nabla E(\mathbf{\Theta}(0))$    ▷ set gradient using initial parameter
    $E = E(\mathbf{\Theta}(0))$    ▷ set error function value
    $p \leftarrow$ randn( size($\mathbf{\Theta}(0)$) )    ▷ initialize momentum
    $\mathbf{\Theta}_{new} = \mathbf{\Theta}(0),\ \mathbf{g}_{new} = \mathbf{g}$
    **for** $t_{sim} = 0 : \Delta t : t$ **do**    ▷ Use “leapfrog” steps to simulate the dynamics
        $\mathbf{p} = \mathbf{p} - \delta \cdot \mathbf{g}_{new}/2$
        $\mathbf{\Theta}_{new} = \mathbf{\Theta}_{new} + \delta \cdot \mathbf{p}$
        Sample list $\leftarrow \mathbf{\Theta}_{new}$    ▷ Record the state samples
        $\mathbf{g}_{new} = \nabla E(\mathbf{\Theta}_{new})$    ▷ Update the gradient and momentum
        $\mathbf{p} = \mathbf{p} - \delta \cdot \mathbf{g}_{new}/2 - \epsilon \cdot \mathbf{p}$
    **end for**
    $E_{new} = E(\mathbf{\Theta}_{new})$    ▷ Find the final value of error function
    $\Delta E = E_{new} - E$    ▷ Use Metropolis rule to decide the new starting point
    **if** $\Delta E < 0$ **then**
        $\mathbf{\Theta}(0) = \mathbf{\Theta}(t)$    ▷ Accept the new starting point
    **else**
        **if** $\exp(-\Delta E) \geq$ rand() **then**
            $\mathbf{\Theta}(0) = \mathbf{\Theta}(t)$    ▷ Accept the new starting point if $\exp(-\Delta E) \leq$ a random number
        **end if**
    **end if**
**end for**
```

our sampling scheme much more efficient in a high-dimensional parameter space.

2.4.2.2 Re-sampling by the monotonicity property

In the first part of our algorithm, an accelerated random walk leads to sample points concentrated in the low error regions. However, if we want to understand the structure of the error function in the whole parameter space, it will not be sufficient if we only sample in the low error regions. In the second part our algorithm, we want to re-sample the error function and build a linear approximation for it.

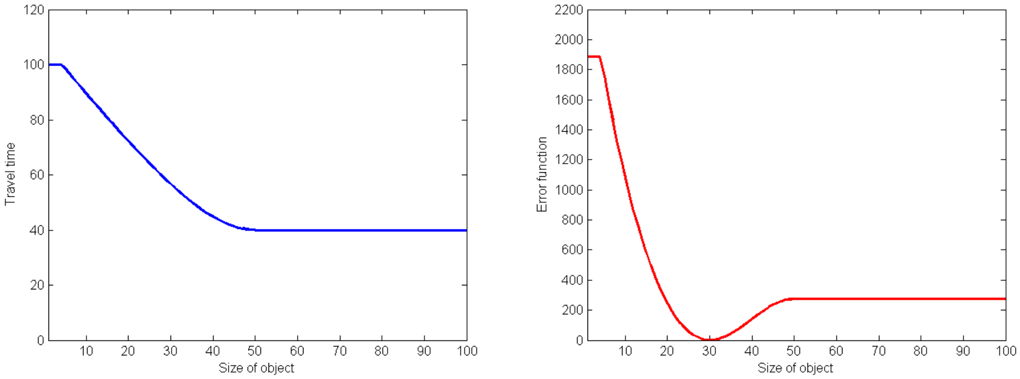
Because the shape of error function is complicated and multi-modal, without any prior information there is no easy way to efficiently choose "good" sampling locations to approximate it. One possible approach would be using the samples from HMC step as starting points to run a pure random walk to explore and re-sample the error function. However, the exploration speed in a pure random walk would be roughly equal to the Nth root of the number of samples, where N is the dimension of parameter space. This would be too slow and we would require exponential samples to cover the whole parameter space. We now show how to use some properties of the error function in order to choose the re-sampling locations efficiently.

We note that the error function is defined by the sum of mismatches corresponding to each measured travel time:

$$\begin{aligned} E(\mathbf{\Theta}) &= \|\mathbf{t} - T(\mathbf{\Theta}, \mathcal{A}, \mathcal{B})\|^2 && (2.17) \\ &= \|t_1 - T_1(\mathbf{\Theta}, \mathcal{A}, \mathcal{B})\|^2 + \|t_2 - T_2(\mathbf{\Theta}, \mathcal{A}, \mathcal{B})\|^2 + \dots && (2.18) \\ &= E_1(\mathbf{\Theta}) + E_2(\mathbf{\Theta}) + \dots, && (2.19) \end{aligned}$$

where each separate error function has a quadratic form $E_i(\mathbf{\Theta}) = \|t_i - T_i(\mathbf{\Theta}, \mathcal{A}, \mathcal{B})\|^2$. Consider the change in the travel time function $T_i(\mathbf{\Theta}, \mathcal{A}, \mathcal{B})$ with respect to the size s_j of object j, while all other parameters are left unchanged. By Lemma 1, the travel time between two arbitrary points will always be non-increasing as the size of a high velocity object increases. Thus, given its quadratic form the change of each individual error function, $E_i(\mathbf{\Theta})$, will be a weakly unimodal function with respect to the change of a single object size(see Figure 2.8).

We make use of this property to choose the re-sampling locations in the parameter space. The basic idea is to re-sample only along the dimensions corresponding



(a) (b)

Figure 2.8: Change of travel time and error function with respect to the size of one high velocity object. (a) The travel time function $T_i(\mathbf{\Theta}, \mathcal{A}, \mathcal{B})$ (b) The error function $E_i(\mathbf{\Theta} = \|t_i - T_i(\mathbf{\Theta}, \mathcal{A}, \mathcal{B})\|^2$. The travel time is monotonically non-increasing, therefore the error function is weakly unimodal.

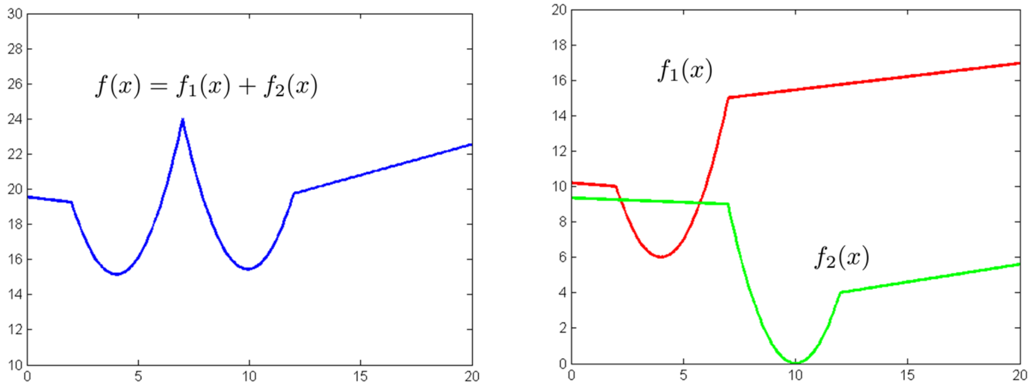
to size parameters, $s_1, s_2, \ldots$. We choose one size parameter each time and use the above property. Because $E_i(\mathbf{\Theta})$ is unimodal with respect to the change of s_j, it can be well represented by linear interpolation if we sample more frequently in large curvature regions. Thus, we select one axis s_j at a time corresponding to the size of object j and use the golden section search [44] to perform selection on re-sampling locations along that axis, while leaving parameters along the other dimensions unchanged. We use these probing points as re-sampling locations, with more locations chosen in the large curvature regions (minima). Thus, compared to the random or uniform sampling, the golden section search sampling is more efficient because it puts few samples in the almost constant regions and focuses on the large curvature regions in the parameter space.

To illustrate our approach, consider an example where we have two objects in the velocity model $\mathbf{\Theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$ and two measured data points $\mathbf{t} = \{t_1, t_2\}$, the error function is defined as $E(\mathbf{\Theta}) = E_1(\mathbf{\Theta}) + E_2(\mathbf{\Theta})$. Given an initial sample $\mathbf{\Theta}(0)$, the parameters can be re-ordered as $\mathbf{\Theta}(0) = \{s_1(0), s_2(0), \tilde{\mathbf{\Theta}}(0)\}$. To find the re-sampling locations on the s_1 axis, we run the golden section search k times along the s_1 dimension on $E_1(\mathbf{\Theta})$ and $E_2(\mathbf{\Theta})$. The re-sampling locations will be $\{s_1(1), s_2(0), \tilde{\mathbf{\Theta}}(0)\}$, $\ldots$, $\{s_1(k), s_2(0), \tilde{\mathbf{\Theta}}(0)\}$ and $\{s_1(k+1), s_2(0), \tilde{\mathbf{\Theta}}(0)\}$, $\ldots$, $\{s_1(2k), s_2(0), \tilde{\mathbf{\Theta}}(0)\}$ where the first k points are chosen by running the golden section search on E_1 and the next k points are on E_2. Likewise, we run the same procedure to choose the re-sampling locations on the s_2 axis.

Note that to re-sample on the s_1 axis, we choose $2k$ re-sampling locations for the error function based on E_1 and E_2 separately. If we run the golden section search directly on the error function $E(\mathbf{\Theta})$, which is the sum of each separate error function $E(\mathbf{\Theta}) = E_1(\mathbf{\Theta}) + E_2(\mathbf{\Theta})$ and may not be unimodal (sum of unimodal functions is not necessary unimodal, see Figure 2.9), the direct search is unlikely to give us good sampling locations. Since each separate function is unimodal and we know how to efficiently sample it, we divide the sampling "budget" and choose the sampling locations based on each separate function. Because the error function is the sum of each separate error function, the large curvature regions for the error function should belong to the union of large curvature regions of each separate error function and our approach selects more samples in the regions where one of the two error functions has large curvature. In Figure 2.10 we show an example of how this approach works better than choosing sampling locations directly on the total function.

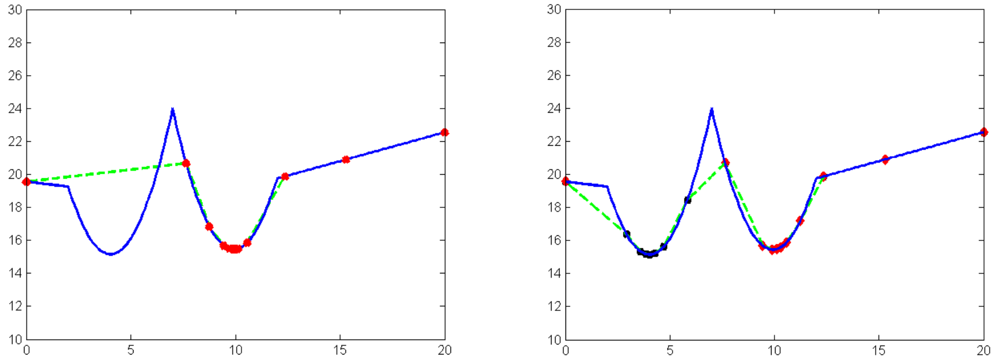
2.5 Simulation Results

Following our previous assumptions, in our simulations we use a velocity ratio $v/v_h \rightarrow \infty$ and choose "line" as the pre-defined convex geometrical object to model the HVS. The model parameters $\boldsymbol{\theta}_j = \{s_j, \mathbf{c}_j, \psi_j\}$ will be the length, line center and



(a) (b)

Figure 2.9: Sum of two unimodal functions. Note the sum of two unimodal functions is not necessary a unimodal function.



(a) (b)

Figure 2.10: Re-sampling comparison. (a) Re-sampling locations chosen by running the golden section search on $f(x)$ (b) Locations chosen separately on $f_1(x)$ and $f_2(x)$. The red "squares" are sampling locations chosen from $f_2(x)$, and black "dots" are from $f_1(x)$. Note that choosing from separate functions gives much better approximation because each one is unimodal.

angle for the line object. The scenario we choose for our experiments is motivated by the problem of modeling a fractured reservoir, where the fractures are usually represented by lines in 2D or planes in 3D.

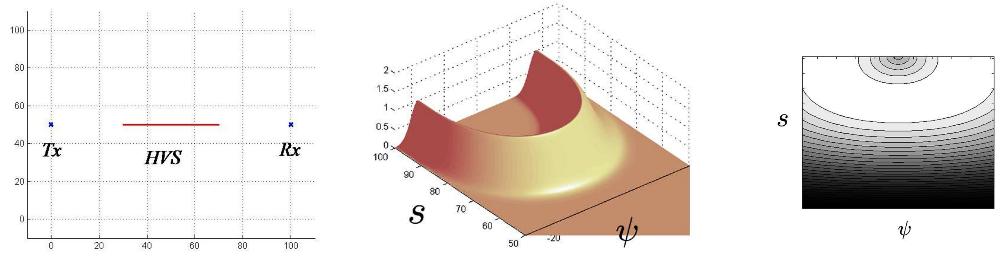
To visualize the object based model in spatial domain, after we sample $\{\boldsymbol{\Theta_1}, \ldots, \boldsymbol{\Theta_N}\}$ N points in the parameter space, we define a mapping function $f(\boldsymbol{\Theta})$ which maps the object parameters into object shapes in spatial domain. Then we calculate the average:

$$\mathcal{M}_f = \int \sigma(\boldsymbol{\Theta}) f(\boldsymbol{\Theta}) d\boldsymbol{\Theta} \approx \sum_{i=1}^{N} \sigma(\boldsymbol{\Theta_i}) f(\boldsymbol{\Theta_i}). \tag{2.20}$$

Because $f(\boldsymbol{\Theta})$ represents the high velocity structure in spatial domain, $\mathcal{M}_f$ can be viewed as the "appearance probability map" of the high velocity in different region.

In simulations, we need to choose the random momentum $\mathbf{p}$ and friction $\epsilon \cdot \mathbf{p}$. We draw the momentum from a normal distribution $N(0, \rho)$, where ρ is equal to the 10% of the maximum possible value of $\boldsymbol{\Theta}$. The friction coefficient ϵ is chosen to be $1/t$ in our experiments .

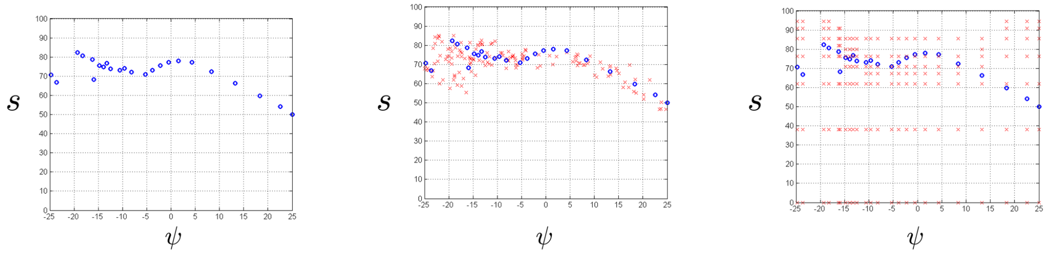
In **Experiment 1**, we illustrate the sampling process. Assuming there is only one measured data point (one travel time between transmitter-receiver pair), and the center of "line" object is fixed. Thus, in this case we only have two parameters $\{s, \psi\}$ which represent the length and angle of the line object. We show the geometry of the line object, sensor locations and the ground truth PDF (which is multi-modal because only one measured data point) in Figure 2.11. Now we apply our algorithm to sample and estimate the PDF: The samples from the HMC step are listed as blue circles, which are concentrated in high probability regions. The results of randomized re-sampling and golden section search are shown in Figure 2.12, where the randomized re-sampling only explores a very narrow region near the starting points and golden section search re-samples along the whole s axis.



(a) (b) (c)

Figure 2.11: The (a) geometry of sensor settings and (b) (c) the PDF for experiment 1.

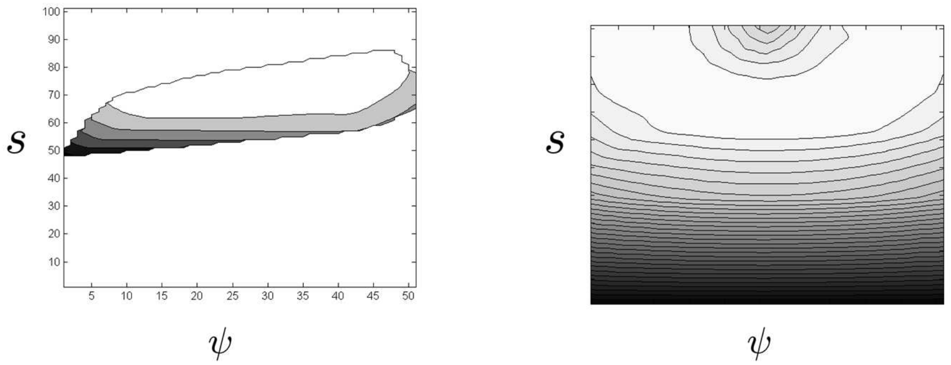
The estimated PDF from the two different approaches are shown in Figure 2.13, which shows that our proposed method has much better estimated PDF than the result from randomized re-sampling. Because the randomized re-sampling only explores a small region near the initial samples, most of the structure of the PDF remains unknown. Our approach choose the re-sampling locations which lead to better approximation of PDF structure.



(a) (b) (c)

Figure 2.12: The HMC sampling and randomized re-sampling in experiment 1. (a) The blue "dots" are samples from HMC. (b) The red "x" are re-samples from random walk. Note it can only explores a small region in parameter space. (c) The red "x" are re-samples from golden section search. The re-sampling locations have the same ψ but cover the whole s axis.

In **Experiment 2**, we increase the number of measured data (2 transmitters and 2 receivers, total 4 measured travel time between transmitter-receiver pairs). The



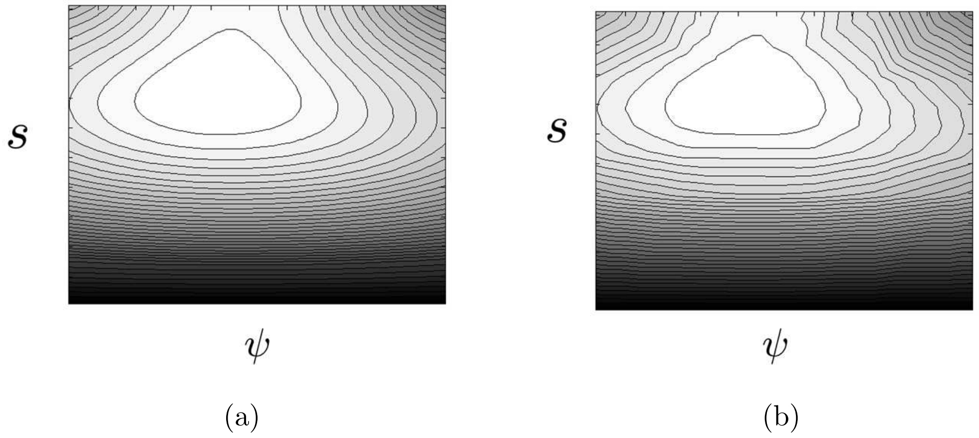
(a) (b)

Figure 2.13: The estimated PDF by (a) Randomized re-sampling (b) Golden section search re-sampling. Note the randomized re-sampling only explores a small region of parameter space and most of PDF is unknown.

PDF and our estimation are shown in Figure 2.14. Comparing to the Experiment 1, the change of PDF is much sharper which implies the model uncertainty is less if we have more data. For example, if we define the acceptable model as $E(\mathbf{\Theta}) \leq \delta$, then the corresponding region in the parameter space is much narrow for the large data case.

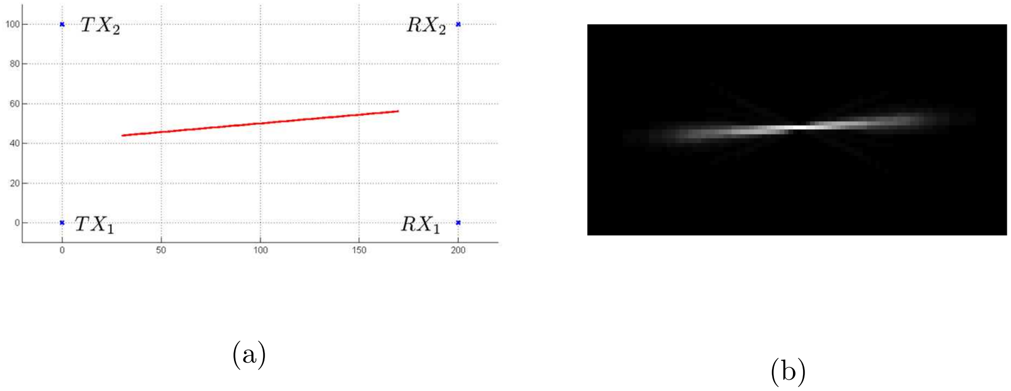
We show the ground truth model and the estimated probability map in Figure 2.15. The result shows that we have high probability areas near the ground truth. Note the center has highest probability - the reason is that in this simple example we fix the center for all HVS models. Thus, when we calculate the appearance probability, all models have a common center point and we will have the highest probability near the center area.

In **Experiment 3**, the transmitters and receivers are placed on the boundary and the HVS is near the center lower area. In the inverse step, we use one line object in the HVS model and all object parameters (center position, length, angle) can be changed freely, so that the parameter space has 4 dimensions.



(a) (b)

Figure 2.14: The PDF and our estimation in experiment 2. Note comparing to experiment 1, the PDF has sharper changes which implies higher model resolution.



(a) (b)

Figure 2.15: (a) The ground truth and (b) the appearance probability map. It shows several different models closed to ground truth all have high probability.

To quantify the reconstructed model error, we define the error metric as the difference between each sampled HVS model and the ground truth, weighted by the probability of each estimated HVS model. We use the Hausdorff distance to characterize the difference of two HVS models. The Hausdorff distance for two objects $\mathbf{X}, \mathbf{Y}$ is defined by:

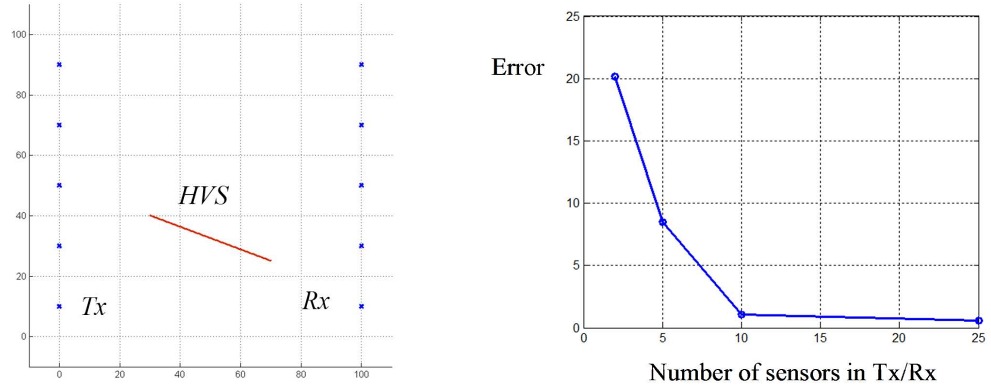
$$d_H(\mathbf{X}, \mathbf{Y}) = \max \left\{ \sup_{\boldsymbol{\alpha} \in \mathbf{X}} \inf_{\boldsymbol{\beta} \in \mathbf{Y}} d(\boldsymbol{\alpha}, \boldsymbol{\beta}), \quad \sup_{\boldsymbol{\beta} \in \mathbf{Y}} \inf_{\boldsymbol{\alpha} \in \mathbf{X}} d(\boldsymbol{\beta}, \boldsymbol{\alpha}) \right\}, \tag{2.21}$$

where $d(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is the distance function of (2.2).

In our result, we map the high velocity model into $2D$ spatial domain, then discretize the result into a set of points. The Hausdorff distance can be viewed as the maximum distance among all points in a set to the nearest point in the other set [60] and is widely used in computer vision to measure the difference between 3D curves or binary images [40]. It also has an interesting property in that when $d_H(\mathbf{X}, \mathbf{Y}) = 0$ then $\mathbf{X}$ and $\mathbf{Y}$ have the same closure. In our case, because we use convex object models, if the Hausdorff distance is zero it implies that the two objects are equal $\mathbf{X} = \mathbf{Y}$. From previous discussion, for a given model parameter $\boldsymbol{\Theta}$, $f(\boldsymbol{\Theta})$ is the function that represents the shape of the corresponding high velocity structure in spatial domain. Then, the error metric with respect to the true HVS $\boldsymbol{\Theta}_{\text{truth}}$ can be written as

$$\begin{aligned} \mathscr{E}(\boldsymbol{\Theta}) &= \int d_H(f(\boldsymbol{\Theta}), f(\boldsymbol{\Theta}_{\text{truth}})) \cdot \sigma(\boldsymbol{\Theta}) d\boldsymbol{\Theta} \qquad (2.22) \\ &\approx \sum_{i=1}^{N} d_H(f(\boldsymbol{\Theta}_i), f(\boldsymbol{\Theta}_{\text{truth}})) \cdot \sigma(\boldsymbol{\Theta}_i), \qquad (2.23) \end{aligned}$$

where $\boldsymbol{\Theta}_i$ are sampled models.

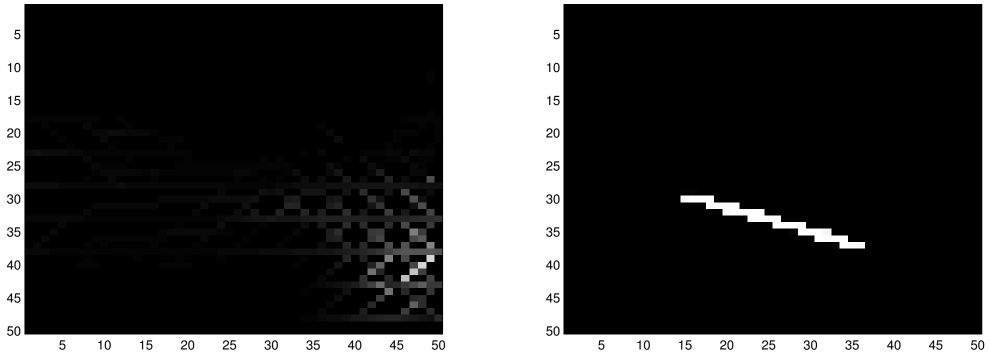


(a) (b)

Figure 2.16: (a) The ground truth and (b) the error metric for experiment 3. Note that the error metric decreases when the number of sensor increases.

We list the error metric for $5 * 5$, $10 * 10$, $25 * 25$, $50 * 50$ transmitter-receiver pairs (see Figure 2.16). Our results show that the error metric decreases as the number of sensors increases. We also list the result with grid-based model and linearized reconstruction algorithm for comparison. Due to the travel paths only covering very few cells, most of the cell's velocity remain unknown and the linearized reconstruction algorithm will assign the high velocity to cells which give the most significant changes for the error function, which usually are the cells closed to sensors. (see Figure 2.17)

In **Experiment 4**, we use the same sensor constellation as in Experiment 3 with a more complicated HVS. For the inversion, we use 3 line objects and the result shows some difficulties to resolve the vertical structure of HVS. In Figure 2.19 and 2.20, it shows that increasing the number of measured data points does not increase the vertical resolution. This is an inherent limitation of travel time tomography, which comes from the relative location of the sensors and the HVS. If the "line" HVS is orthogonal to the travel path, it will not affect the travel time at all. In this case most of the transmitter-receiver pairs are in the horizontal direction, so



(a)

(b)

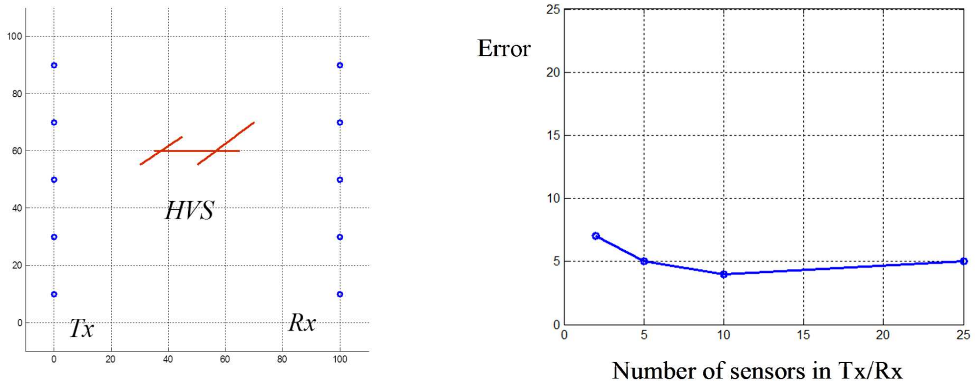
Figure 2.17: (a) The grid-based result with 25*25 transmitters/receivers and (b) the appearance probability map for Experiment 3. Note the grid-based model fails to recover the HVS location.

that the vertical resolution is very limited. This is the reason we have "phantoms" in the vertical direction and the error metric (see Figure 2.18) does not decrease when we increase the number of measured data points.

From the above experiments, we show that our algorithm can estimate the PDF and recover possible models for high contrast travel time tomography with sparse data. Our algorithm is robust to noise because we estimate the PDF from all measured data. For example, if one measured travel time is affected by shot noise, the corresponding error function E_i will be significantly changed. But the error function is defined as the sum of all individual error functions (see equation (2.12)), thus, the effect of shot noise will be "averaged" out by other good measure data points which provides the robustness for estimation.

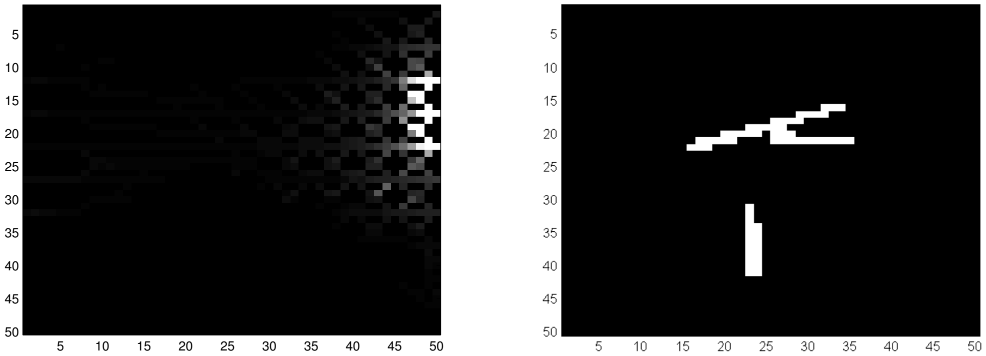
2.6 Conclusion

The main purpose of this paper is to propose a new approach for the reconstruction of high contrast discrete velocity models in travel time tomography. To image the



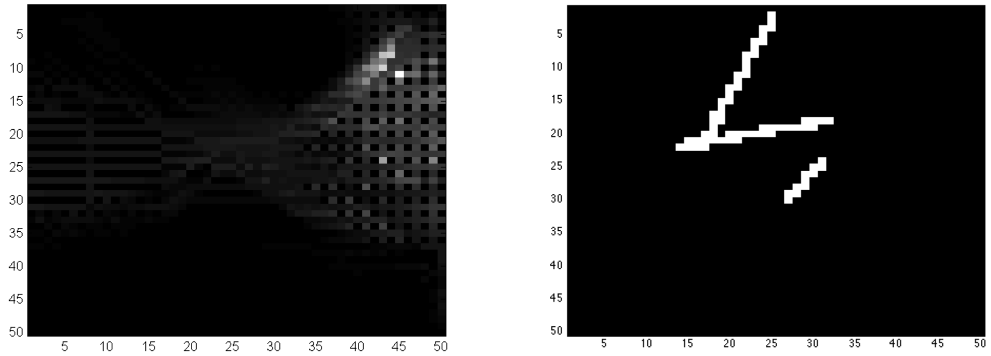
(a) (b)

Figure 2.18: (a) Ground truth and (b) error metric for experiment 4. Note the error metric does not decrease monotonically due to the vertical phantom.



(a) (b)

Figure 2.19: (a) Grid-based result and (b) appearance probability map for $10 * 10$ sensors in experiment 4



(a) (b)

Figure 2.20: (a) Grid-based result (b) Appearance probability map for 25 * 25 sensors in experiment 4

high contrast media, we take advantage of the high velocity structures and prove that the travel trajectory is piecewise linear when we consider convex object model. We show that our travel path finding method is able to compute the corresponding travel time much faster than the conventional ray-tracing method because we consider the number of objects as “nodes” instead of number of cells in grid model. Thus, the complexity scales with the number of objects, instead of the (much larger) number of cells in a grid.

A model based approach for high velocity structures (HVS) is studied and the error function is defined by the misfit between the predicted travel time based on current model and the measurements. We develop a reconstruction algorithm to efficiently sample the error function in the model parameter space. After we map the corresponding model parameters back to the object shape in spatial domain, we can obtain the HVS appearance probability in different areas. In our simulations, we show how our algorithm samples and approximates the error function, finding set of possible HVS models. The results also show our algorithm has better reconstructions compared to the typical grid-based model.

Future work will be to explore how to refine the idea of object based model to achieve an efficient representation of the HVS structure. For example, we are looking at how to define or adaptively change the shape of objects to have a sparse representation. And the optimum number of objects for the model is another interesting question. Increasing the number of objects will provide a better detail representation of structure, but also increase the dimension of parameter space. We plan to explore the trade-off between the number of objects and computational complexity of our randomized sampling algorithm.

Chapter 3

Waterflooding Tomography

3.1 Introduction

Recently, due to the high demand of crude oil, enhanced oil recovery (EOR) techniques have been widely used to boost oil production. The most commonly used EOR approach is fluid e.g., gas or water injection in order to maintain the pressure of the reservoir and increase the amount of oil that can be extracted. In this chapter, we consider the "water-flooding" process where water is injected to the reservoir to stimulate production. In a water-flooding project, we would like the injected water to uniformly increase the pressure through the reservoir and "push" the residual oil toward the production wells. The fluid flow through a porous medium is described by Darcy's law, which states that the flow rate is proportional to the permeability. Therefore, it is critical to control the injection amount in different locations because the heterogeneous structures affect the sweep efficiency dramatically. Heterogeneous structures could be fractures or faults, which act as a high permeability channels or barriers for the flow. For example, if a high permeability channel is very close to the injection-production well pair, most injected water will flow through it and fail to sweep the oil in other regions. This phenomenon is called “water cycling”, since much of the water injected is later produced. Water cycling decreases the sweep efficiency and increases the water cut (the ratio of water produced compared to the volume of total liquids produced), and high water cut requires special treatment in the oil-water separation. Moreover, when the oil production efficiency is low, the overall cost (per barrel of oil produced) increase significantly because it includes lifting, separation, filtering, pumping and injection of water. Another example is that if an injection well is near a fault, very little water can flow through the barrier, which will also decrease the sweep efficiency dramatically. To handle this situation, many authors have considered production optimization as an approach to compensate the impact of heterogeneity and improve sweep efficiency [5, 15, 67]. Thus, to optimize the water-flood efficiency it is critical to identify these heterogeneous structures in a reservoir.

Many reservoir characterization methods have been proposed in the past few decades. The most widely used one might be seismic cross-hole tomography, which provides very detailed geological structures [14,64]. However, it is difficult to use the seismic results to infer flow properties directly. In early work, pulse testing [41, 53] was used to measure the pressure build-up curve in nearby wells. This method provides a direct measurement of the pressure wave propagation time, but field operations need to be interrupted (because the field needs to be shutdown to enable accurate measurements) Moreover, downhole pressure gauges are hard to maintain. Another popular method is tracer testing [2, 3, 24], which injects a tracer chemical at one well and monitors the concentration at all the production wells to estimate the diffusion process between injector-producer pairs and provide a direct estimate of flow permeability distribution. Vasco [73] also proposed combining the dynamic data obtained from tracer tests and that obtained from productions rates in order to generate a high-resolution representation of the reservoir.

A common aspect of these methods is that they all require extra equipment and may interrupt day-to-day operations. Seismic cross-hole testing needs to induce seismic waves and deploy sensors to monitor the reflected signals, which requires additional equipment and manpower. In tracer tests, chemicals or radioactive materials are injected and the concentration changes are monitored to map the inter-well flow properties. But repeating the testing is difficult - either a different tracer is used or we need to wait a significant amount of time for the tracer concentration to return back to the background level.

In this chapter, we consider the case where water-flooding is applied to an oil reservoir where wells are equipped with automatic measurements and control valves, which gives us the freedom to control the rates of injection and monitor the resulting production changes in real time. Thus, we propose to vary the injection rate and use these variations as a “probe” signal then measure the resulting changes in production. The variation of injection rates will induce a pressure wave passing through the field, so that the injection/production wells can be treated as transmitting/receiving sensors as described in Chapter 2. Then we estimate the travel time between well pairs (the time it takes for a producer to respond to a change in an injector), and apply the travel time tomography techniques we developed in Chapter 2 to characterize the permeability of the reservoir. The main advantage over other methods (e.g., seismic and tracer test) is that this technique can be applied based on current infrastructure without additional cost.

We treat the reservoir as a multiple input multiple output (MIMO) system and use the capacitance model (CM) [80] to model the injection-production response. We design the variation of injection rates based on the previously proposed system identification techniques [49], which allow the injection rates to be chosen to maintain a constant average rate, so that we can keep the same overall average production level and estimate the CM parameters from the production rate changes at individual wells. The "time delay constant" in CM is used as an estimate of travel time between well pairs.

In this chapter, we mainly focus on identifying fractures in a low permeability reservoir, where the fractures act as high permeability channels in low permeability background. The permeability contrast can be assumed to be very high - open fractures are about 10^5 times more permeable than the host rock. Thus, the travel time for the pressure wave to propagate through high permeability channels can be considered to be almost negligible. Note that travel times can only be estimated for the existing injection-production well-pair locations, which are sparsely located in the field. In computed tomography, it is well known that the quality of the reconstructed results is related to the density of travel ray-path. Thus, the reconstructed reservoir image resolution is fundamentally limited by the spatial distribution of well locations.

Based on these conditions, we can formulate a high contrast travel time tomography problem with the goal to characterize these high permeability channels in a low permeability fractured reservoir. If we use a grid-based model and apply the conventional iterative least-squares methods [12], the reconstruction results will be very poor in general. To identify these high permeability channels, following the approach in Chapter 2, we use an object-based model, with "lines" as fundamental objects to represent fractures in $2D$. We use the algorithm described in Chapter 2 to estimate the probability distribution of the model parameters, directly pointing to the location of high permeability channels. We can also map the result into the spatial domain and view it as a probability map to represent the probability of appearance for high permeability channels in different regions. We validate our approach by running simulations on a commercial reservoir simulator [50], and also report the results of a field experiment. The results show very good model estimation in our simulations, and the field trial result was seen to be consistent with the known fault location as well as the expectations from field production engineers.

The rest of this chapter is structured as follows: in Section 3.2, we introduce the system model to treat the reservoir as a linear MIMO system. We use the capacitance model to characterize the injection-production response, and explain why we can use the time delay constant in CM as the travel time in a tight reservoir. With the system model, in Section 3.3 we describe how to estimate the CM parameters from historical data and formulate the tomography problem in Section 3.4. In Section 3.5 we describe the simulation results using CMG, as well as the field trial result. We conclude this chapter in Section 3.6.

3.2 Physical Model

3.2.1 Injection-Production model

The recently introduced capacitance model (CM) [79, 80] has generated significant interest in the reservoir modeling community. Different from traditional complex reservoir models, it treats injection and production as inputs and outputs of the system, respectively, and uses signal processing techniques to estimate the unknown system parameters, which relate to reservoir physical properties. CM uses a linear multi-input multi-output (MIMO) system to model the response between injection/production rate. Yousef et al. [81] show how to extract reservoir physical properties from historical data based on the parameters of the capacitance model and use this model to optimize the water-flooding recovery efficiency. CM provides a powerful tool to model the flow properties of a reservoir and evaluate the water-flooding performance.

The mathematical formulation for one injection-production well pair in CM can be written as:

$$\tau \frac{dP(t)}{dt} + P(t) = I(t) - \tau \cdot J \frac{dP_{wf}}{dt}, \tag{3.1}$$

where $I(t)$ is the water injection rate and $P(t)$ is the fluid production rate. P_{wf} represents the flowing bottom hole pressure (BHP) and J stands for the productivity index. If we consider the case where the reservoir is mature with fixed bottom-hole pressure, we can ignore the production induced by initial pressure and the change of BHP. Under this assumption, the production rate only depends on the injection rate and can be simplified as follows:

$$P(t) = \int_0^t \frac{e^{-\eta/\tau}}{\tau} \; I(t-\eta) \; d\eta. \tag{3.2}$$

The corresponding discrete form will be

$$P[n] = \sum_{k=0}^{n} \frac{1}{\tau} \; e^{-k/\tau} \; I[n-k], \tag{3.3}$$

where $k = 0, 1, \ldots, n$. The impulse response is controlled by the “time delay constant” τ, which is defined by the total compressibility c_t, the productivity index J, and the pore volume V_p associated with the control volume between injector-producer well pair:

$$\tau = \left(\frac{c_t V_p}{J} \right). \tag{3.4}$$

The formulation in (3.3) can be extended to cases with multiple injection-production well pairs. For a MIMO system, additional inter-well connectivity parameters λ_{ij}, are introduced to represent the flow distribution from injector i to producer j. Then the output flow rate in producer j is the sum of contribution from all different injectors:

$$P_j[n] = \sum_i \sum_k \lambda_{ij} \frac{1}{\tau_{ij}} e^{-(n-k)/\tau_{ij}} I_i[k]. \tag{3.5}$$

According to the conservation of mass, the output flow cannot be greater than the input flow, which implies the sum of flow distribution for an injector is less than or equal to one. Thus, the sum of inter-well connectivities for any injector should be less than or equal to one. Moreover, for a MIMO system, the time delay constant is controlled by the reservoir property of the region corresponding to well pair $i-j$:

$$\sum_j \lambda_{ij} \leq 1, \quad \tau_{ij} = \left(\frac{c_t V_p}{J}\right)_{ij}. \tag{3.6}$$

Thus, we model the reservoir as a MIMO system, where the impulse response is related to the reservoir properties. In the next section, we will explain the relation between the time delay constant τ_{ij} and the flow paths, and also justify why we can use τ_{ij} as a proxy for the travel time time between well pair $i-j$ in a low permeability fractured reservoir.

3.2.2 Travel time in a tight fractured reservoir

From the above discussion, we know that the time delay constant is determined by the control volume and productivity index. If the reservoir is homogeneous, the fluid flow should be uniform and the travel path will be a straight line between injector and producer. In particular, we can assume that the control volume is proportional to the distance between the corresponding well pair if the region is uniform.

If the the reservoir is not homogeneous, we need to take into account the permeability variations between well pairs. One possible approach is to apply a grid-based model to parameterize the permeability distribution, which assigns the same permeability value to the area within one cell in the grid. Then we can obtain the CM parameters and cascade all the responses in each cell to model the total response between injection i and producer j [62]. However, solving and cascading CM for every cell is very computationally expensive. When we consider a tight fractured reservoir, which has tight rocks as background embedded with open fractures, we can assume that these fractures have much higher permeability than the background. Thus, the open fractures in a tight reservoir mean that the permeability model is “high contrast”, where we assume the tight rock to be homogeneous and low permeability, and fractures to be very high permeability objects.

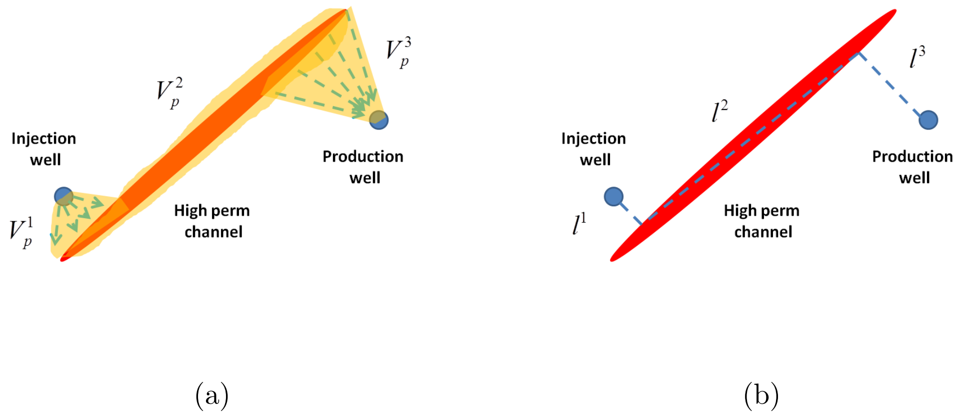
In such case, the flow path is dominated by these fractures because they have much higher permeability, and provide a fast pathway for the fluid (see Figure 3.1). Moreover, the total control volume can be modeled as a cascade of a few control volumes along the flow path connecting injector i and producer j. Each control volume is either a region in the high permeability channel or in the background. Note that the control volume in the background is proportional to the length of flow path through it because the background is uniform. (see Figure 3.1).

Thus, the time delay constant can also be viewed as the “sum” of different parts, each of which is proportional to the corresponding control volume divided by the local productivity index. Because the productivity index in the high permeability channel is much higher than the value in the background, when we calculate the time delay constant, we can ignore the part of time delay constant corresponding to these high permeability channels.

We use the travel time definition in Chapter 2, where the travel time is the length of travel path divided by travel velocity. In the high contrast object based velocity model, the travel path can be approximated by straight line segments between objects. We denote l^i as the travel path and v_i as the travel velocity in the ith object. With the assumption that the velocity in high permeability channels is much higher than in the background, we can ignore the cost time to travel through the channels and the travel time is dominated by the path and velocity in background v_{base}.

Following these assumptions, the time delay constant will be proportional to the travel time in a tight fractured reservoir. As an example, when we have an injection well and a production well with a fracture as high velocity object, we use l^1, l^2, l^3 to represent the path from injector to object, the path inside object, and the path from object to producer (see Figure 3.1b). The time delay constant can be represented as:

$$
\begin{aligned}
\tau &\approx \tau^1 + \tau^2 + \tau^3 && (3.7)\\
&= (\frac{c_t^1 V_p^1}{J^1}) + (\frac{c_t^2 V_p^2}{J^2}) + (\frac{c_t^3 V_p^3}{J^3}) && (3.8)\\
&\approx \frac{c_t^b}{J^b}\left(V_p^1 + V_p^3\right) && (3.9)\\
&= \frac{c_t^b}{J^b}\left(\mathfrak{c}\cdot l^1 + \mathfrak{c}\cdot l^3\right) && (3.10)\\
&= \frac{c_t^b}{J^b}\cdot\mathfrak{c}\cdot v_{base}\left(\frac{l^1}{v_{base}} + \frac{l^3}{v_{base}}\right) && (3.11)\\
&= \mathfrak{h}\cdot t && (3.12)
\end{aligned}
$$



(a) (b)

Figure 3.1: With high permeability channel (a) the total control volume can be viewed as cascade of small ones, and (b) the travel path is a combination of several line segments. Note that because the productivity index in high permeability channel is very high, the time delay constant inside can be ignored.

where V_p^i is the ith control volume, and c_t^i, J^i are corresponding compressibility and productivity index (see Figure 3.1a). If the control volume belongs to the background region, we use c_t^b and j^b to represent the compressibility and productivity in the homogeneous background. The constant $\mathfrak{h}$ can be estimated from lab experiments or field data. In particular, (3.7) allows us to use the time delay constant as the corresponding travel time and solve the fractured reservoir characterization problem as a high contrast travel time tomography.

3.3 Estimation of CM parameters

From the previous discussion, we use CM to model the response between injection and production rate, then take the “time delay constant” as the travel time between wells. In this section, we explain how to estimate the time delay constant from historical injection-production data. To formulate the CM parameter estimation problem, we define the fitting error for a certain model with parameters $\boldsymbol{\vartheta}$ by:

$$\varepsilon(t, \boldsymbol{\vartheta}) = P(t) - \hat{P}(t|\boldsymbol{\vartheta}), \tag{3.13}$$

where $\hat{P}(t|\boldsymbol{\vartheta})$ is the predicted production rate based on model $\boldsymbol{\vartheta}$. When the production rate $P(t)$ is measured for $t = 1, \ldots, N$, the fitting error can be computed.

To estimate the CM parameters, one approach is to run an optimization directly on the CM parameters to minimize the fitting error. In this approach, the number of unknown variables is equal to the number of CM parameters, but the estimated result may not be stable and may depend on how the minimization is initialized [62]. This is because the CM parameters are not linearly related to the production data, thus, estimating the CM parameters becomes a nonlinear optimization. Another approach is a "multi-stage" strategy: Instead of estimating CM directly, in the first stage we use a finite impulse response (FIR) model as an intermediate model and estimate FIR coefficients (see Figure 3.2). This has the advantage that the FIR parameters are linearly related to the data and the estimated coefficients are always stable. Then, in the second stage, we estimate the time delay constant by finding the CM parameters that best fit the FIR model obtained in the first stage.

The trade-off between these two approaches is that when we choose the multi-stage approach, we need to estimate the FIR model, which has many more parameters than CM. Thus, to achieve the same confidence level, it requires more data (longer experiment duration). For example, when the amount of injection-production data is limited, the multi-stage approach may not be a good choice because the FIR model estimation requires the amount of data to be at least the same as the length of the impulse response. In this situation, we may need to estimate the FIR coefficients based on insufficient data, which makes this problem ill-conditioned. Thus, when insufficient data is available, we choose to perform the CM parameter estimation directly by solving a nonlinear optimization.

In conclusion, the choice between direct estimation or multi-stage approach depends on the length of experiment duration. When the experiment duration is long enough to provide sufficient data, we can choose multi-stage approach to have a stable estimation result. Otherwise, we prefer the direct estimation approach to avoid using FIR as an intermediate model.

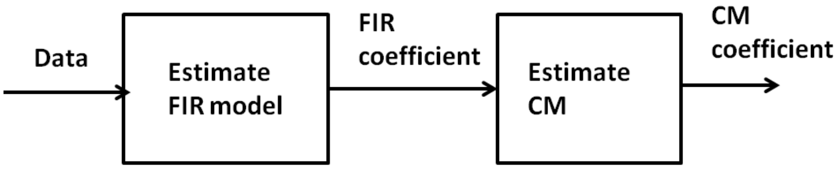


Figure 3.2: The multistage approach

3.3.1 Injection Sequence Design

To get a reliable estimate of the model parameters, the input sequences must have enough "richness" of information to distinguish the response of all possible models. For example, if we apply a constant input to a linear system, the output will also be constant and there is no way that we can estimate the model parameters from this dataset. This leads to the problem of "input design", which has been well-studied in system identification literature [49]. Based on system identification concepts, we choose the pseudo-noise (PN) sequences as the inputs. It is well known that PN sequences have the lowest average correlations, which implies that the estimated FIR parameters will have the lowest error variance [33]. This procedure has been proposed to determine the inter-well connectivity in reservoir modeling by Lee et al. [48].

However, in practice the amplitude and length of input sequences are two major constraints to design PN sequences. For example, higher input energy will provide a better signal-to-noise ratio (SNR) and the estimated results can achieve lower variance. But in a reservoir, the physical flow property is characterized by Darcy's law [76], where the input-output response is highly nonlinear. When we use a linear model to approximate the reservoir response, we assume the reservoir is operating in a region of almost linear response. If the input variation is too large, it may drive the physical system far away from the previous condition and the linear system response model may not be longer valid. On the other hand, if we choose a small variation for the input amplitude, the system response should be almost linear. However, the small change amplitude in input will result in low SNR and degrade the estimation results.

Another issue is the experiment time duration, which is related to the length of input sequences. It is well known that the length of PN sequence will determine the lowest input frequency and the cross correlation function, where longer sequences will provide better estimation results. For example, when we work with a very low permeability reservoir, the impulse response is very slow and it requires low frequency input to correctly estimate the model. In such cases, short input sequences cannot provide enough low frequency contents and the estimate reservoir response is very unreliable. Rezapour et al. [1] have studied the problem of injection design from a system identification perspective, and have analyzed the physical constraints and developed empirical design rules.

For our application, we use field knowledge and an ad-hoc approach to design the input amplitude and testing period. In the trial run, we use the PN sequences as input with maximum variation to be 50% of the average amplitude, and try different variation levels for PN sequences, then estimate the response. The goal is to find the maximum allowable input amplitude to provide enough SNR while the system still has the same response. Moreover, the testing period can be predicted according to prior knowledge of the reservoir. For example, we may use the core sample drilled from reservoir to estimate the background permeability, then use the distance between wells to predict the maximum possible time delay. In particular, the testing period should be chosen to be longer than the maximum possible time delay.

3.3.2 Model Estimation

With the designed injection schedule as the input signal, we use the production rates as the system outputs and formulate the model estimation problem. We assume that there are $\mathcal{A} = (a_1, \ldots, a_{tx})$ injectors and $\mathcal{B} = (b_1, \ldots, b_{rx})$ producers in the field, where $I_i[t]$ represents the injection rate for ith well and $P_j[t]$ is the production rate for jth well. Thus, we can model the inter-well response as a MIMO system with $I_i[t]$ as input and $P_j[t]$ as output. As we mentioned before, when we consider a mature reservoir, the system follows the material balance property, which implies that the total fluid output is less than or equal to the fluid input. Thus, the impulse response coefficient should be non-negative and the sum of impulse response should be less than one. This property provides us additional constraints to help us estimate the model parameters.

3.3.2.1 Multi-stage approach

In the multi-stage approach, we first estimate the parameter of an FIR model that is used as an intermediate step. The predicted output can be defined by the convolution of the FIR and the input signal:

$$\hat{P}_j[t] = \sum_{i=1}^{tx} \sum_{k=0}^{M-1} h_{ij}[k] \cdot I_i[t-k]. \tag{3.14}$$

where we assume the FIR filters have length up to M and $h_{ij}[k]$ represents the FIR coefficients corresponding to the $i-j$ well pair. When we choose the L_α norm as data fitting metric, then the FIR parameter estimation can be formulated as an optimization problem:

$$h_{ij}[k] = \underset{h_{ij}[k]}{\arg\min} \left(\sum_t \left| P_j[t] - \hat{P}_j[t] \right|^\alpha \right)^{1/\alpha} \tag{3.15}$$

$$\text{subject to} \quad h_{ij}[k] \geq 0, \ \sum_{j=1}^{tx} \sum_{k=0}^{M-1} h_{ij}[k] \leq 1 \tag{3.16}$$

where the constraints correspond to the non-negative response and material balance properties. Note that the FIR filter length should be longer than the maximum possible time delay between wells, which can be estimated from the prior knowledge of reservoir. For example, the maximum possible time delay can be estimated from reservoir core sample, tracer testing, even previous pilot water-flooding. In particular, if the FIR length is too short, the FIR will not model the reservoir response correctly. On the other hand, longer FIR length always provides a more accurate model of the reservoir response, but it requires more data to estimate the FIR parameters reliably.

In the second stage, we want to estimate the CM parameters and this is done by selecting the CM parameters that provides a best fit of the FIR model we estimated in the previous stage. The impulse response curve in CM, described in (3.3), is defined by the inter-well connectivity λ_{ij} and time delay constant τ_{ij}. To estimate λ_{ij} and τ_{ij}, we formulate a constrained optimization for the curve fitting:

$$\begin{aligned} \{\lambda_{ij}, \tau_{ij}\} = & \quad \underset{\{\lambda_{ij}, \tau_{ij}\}}{\arg\min} \sum_k \left| h_{ij}[k] - \lambda_{ij} \tfrac{1}{\tau_{ij}} e^{-(k/\tau_{ij})} \right|^2 && (3.17) \\ \text{subject to} & \qquad 0 \leq \lambda_{ij} \leq 1, \quad \tau_{ij} \geq 0. && (3.18) \end{aligned}$$

Note that we estimate $\{\lambda_{i,j}, \tau_{ij}\}$ based on the FIR coefficients $h_{ij}[k]$ corresponding to the $i - j$ well pair. Thus, each time we only need to solve a curve fitting problem for one input-output pair with two unknown parameters. Compared to the direct estimation described in the next section, we break down a complex nonlinear optimization into many small ones, which makes the multi-stage approach very effective.

3.3.2.2 Direct Estimation Approach

If we choose to estimate the CM parameters directly from the data, the predicted output from the CM will be:

$$\hat{P}_j[t] = \sum_{i=1}^{tx} \sum_{k=0}^{\infty} \lambda_{ij} \frac{1}{\tau_{ij}} e^{-k/\tau_{ij}} \cdot I_i[t-k]. \qquad (3.19)$$

Thus λ_{ij}, τ_{ij} will be chosen to minimize the error between predicted and measured output:

$$\begin{aligned}
\{\lambda_{ij}, \tau_{ij}\} = &\quad \underset{\lambda_{ij},\tau_{ij}}{\arg\min} \left(\sum_t \left| P_j[t] - \hat{P}_j[t] \right|^{\alpha} \right)^{1/\alpha} && (3.20) \\
\text{subject to} &\quad 0 \le \lambda_{ij},\ \sum_j \lambda_{ij} \le 1,\ \tau_{ij} \ge 0. && (3.21)
\end{aligned}$$

Note that in this case, the constraint $\sum_j \lambda_{ij} \le 1$ is not separable, thus, we need to solve a nonlinear optimization with $2 \times tx \times rx$ unknown parameters, where tx, rx are the number of injectors and producers, respectively. This implies that we may face a high dimensional nonlinear optimization problem which is very difficult to handle. The benefit is that does not require using FIR as an intermediate model, which would have a number of unknown parameters growing linearly with the length of the impulse response.

3.4 Tomographic Reconstruction

In the previous section, we explain how to estimate the "time delay constant" between wells from injection/production data. Thus, we can use injection and production wells as sensors and the estimated time delay constant as the equivalent travel time between them, then model the fractures as objects, i.e., "lines" in 2D or "planes" in 3D. Because the travel velocity of fluid through fractures is assumed to be much higher than through the tight rocks in the background, detecting the fractures can be formulated as a high contrast travel time tomography problem. Thus, we apply the object based reconstruction algorithm described in Chapter 2 to estimate the location of fractures.

In our approach, fractures are modeled by line objects which are parameterized by length l, center c and rotation angle ψ. We define a parameter vector $\boldsymbol{\theta}_i = \{l_i, c_i, \psi_i\}$ to represent ith line object. Assuming that there are at most K objects in the reservoir model, we can cascade all object parameters and define a vector of model parameters, $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K\}$.

We use the travel time function $\mathbf{T}(\boldsymbol{\Theta}, \mathcal{A}, \mathcal{B})$ defined in Chapter 2, which is a vector function representing the predicted travel time between each injector-producer pair based on the velocity model with parameter $\boldsymbol{\Theta}$:

$$\mathbf{T}(\boldsymbol{\Theta}, \mathcal{A}, \mathcal{B}) = (T_1(\boldsymbol{\Theta}, \boldsymbol{\alpha}_1, \boldsymbol{\beta}_1), \ldots, T_n(\boldsymbol{\Theta}, \boldsymbol{\alpha}_{tx}, \boldsymbol{\beta}_{rx})). \tag{3.22}$$

We can cascade all estimated travel times as a vector, $\mathbf{t} = \{t_{1-1}, \ldots, t_{tx-rx}\}$ and define the error function as a quadratic data-fitting error between the travel time predicted from the forward model and the measured travel time:

$$E(\boldsymbol{\Theta}) = \|\mathbf{t} - \mathbf{T}(\boldsymbol{\Theta}, \mathcal{A}, \mathcal{B})\|^2. \tag{3.23}$$

Thus, the probability density function $\rho(\boldsymbol{\Theta})$, which is modeled as a Gaussian uncertainty of the error, measures the confidence on different models:

$$\sigma(\boldsymbol{\Theta}) = c \cdot e^{-E(\boldsymbol{\Theta})}, \tag{3.24}$$

where c is a normalization constant. To visualize the result, we define the mapping function $f : \boldsymbol{\Theta} \rightarrow \mathbf{X} \times \mathbf{Y}$, which maps the object parameters into corresponding high velocity areas in 2D spatial domain. The weighted average of different models in 2D is defined as:

$$\mathcal{M}_f(x, y) = \int_{\mathbf{u}} \sigma(\boldsymbol{\Theta}) f(\boldsymbol{\Theta}) d\boldsymbol{\Theta}. \tag{3.25}$$

$\mathcal{M}_f(x, y)$ can be viewed as the appearance probability of the high permeability channels in different regions. We use the randomized sampling method in Section 2.4.2 to estimate the probability density in the model parameter space and calculate $\mathcal{M}_f(x, y)$.

3.5 Experiment Results

3.5.1 Numerical Simulations

To validate our proposed method, we run numerical simulations on a commercial reservoir simulator [50]. In our simulated reservoir model, we place a high permeability channel in very low permeability background to simulate a tight fractured reservoir. Moreover, to simulate a mature reservoir, we first apply constant injection rates until the production rates become stable. Then we apply our designed injection schedule, where the purpose of our method is to detect the direction of fracture without stopping regular operation. In particular, we assume the fluid production is observed on the daily basis. Because we have the freedom to choose the length of testing period, we can collect enough data and use the multi-stage approach which estimates the FIR model first. Then we estimate the time delay constant and inter-well connectivity in CM by running the curve fitting between FIR and CM.

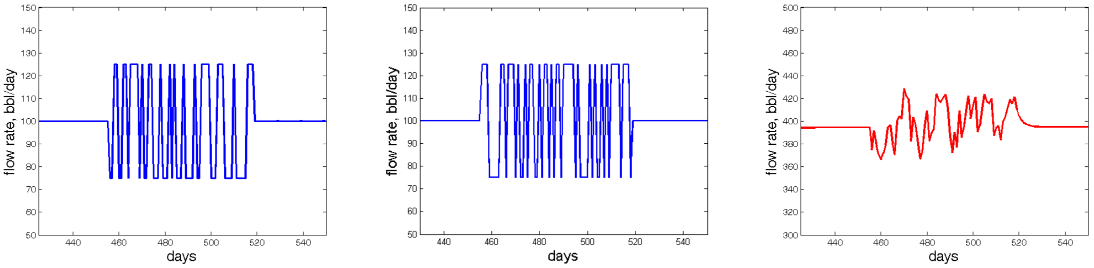
With the estimated time delay constant, we use the object-based reconstruction algorithm to estimate the locations of fracture. In this case, we model the fracture as a "line" object, and run the PDF estimation. Moreover, we calculate the estimated probability map and compare it with the ground truth in simulator.

In **Simulation 1**, the well installation pattern is a 5-spot with 4 injectors located on the corners and 1 producer in the center. For the reservoir model, we place a high permeability channel on the producer to simulate the effect of hydraulic fracturing. In this case, the fracture must pass through the production well, thus, we choose the number of "line" objects to be equal to 1 and add the constraint that the line object must pass through the location of producer, $\mathcal{P} \in \mathfrak{l}_1$, where $\mathcal{P}$ is the location of producer and $\mathfrak{l}_1$ is the line with length, center and angle parameters $\{l_1, c_1, \psi_1\}$.

We use the PN sequences for the injection rates at each injector, which roughly keeps the same average injection rate (see Figure 3.3). We set the frequency of injection rate changes to be to 1 day, and the total length of the testing period to be 63 days. Thus, according to the material balancing rule, the average production is almost at the same level as before the test, which implies our method does not stop or decrease the daily production as many other methods do.

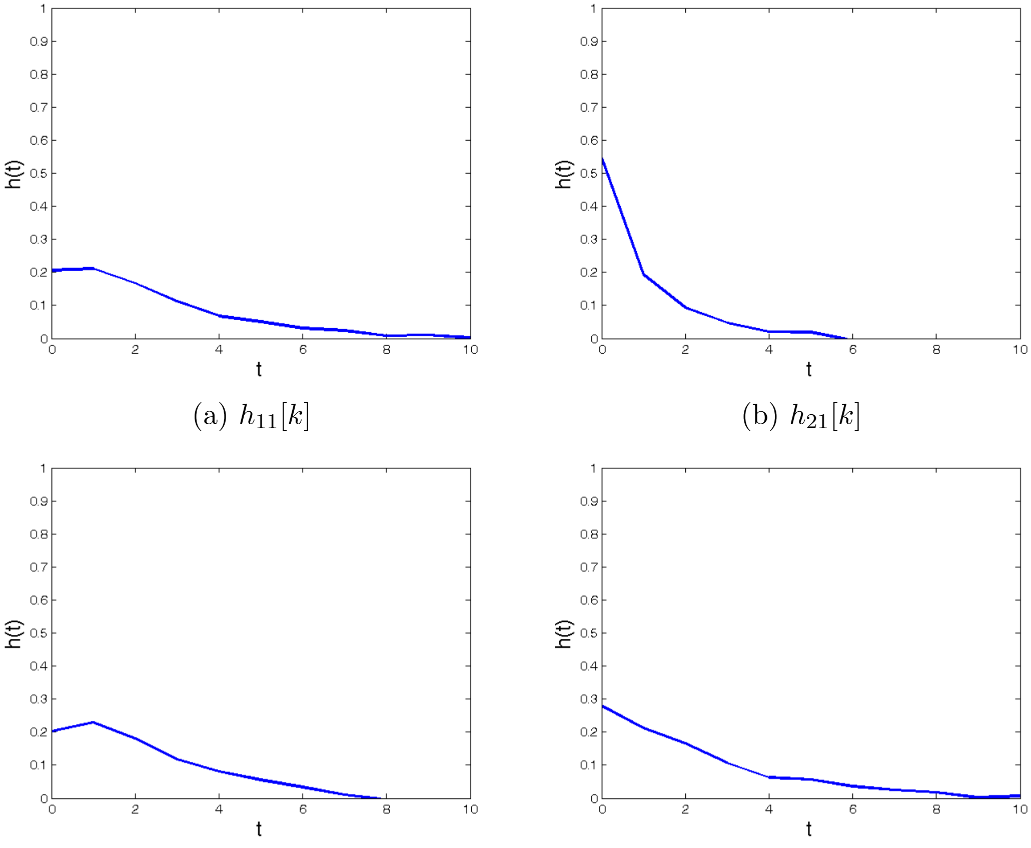
After running the multi-stage estimation, the result shows the time delay constant between injector 2 and producer 1 is significantly less than others, which suggests possible existence of high permeability channels along that direction. Moreover, we provide the estimated probability map in Figure 3.5, which shows high appearance probability for the high permeability channel is near the ground truth. The model with highest probability is very close to the ground truth - the angle difference between the estimated most likely line and the ground truth is within 5 degrees.

In **Simulation** 2, we test a case where the well installation pattern is a line drive with 5 injectors and 5 producers. For the reservoir model configuration, there is a high permeability channel between injector 5 and producer 3. To compare with different input sequences, we apply step functions as the change of injection rate in this simulation, which involves only changing one injection rate at each time.



(a) (b) (c)

Figure 3.3: Simulation 1 (a) injection rate in injector 1 (b) injection rate in injector 2 (c) measured production rate in simulation 1. Note that we vary the injection rate in injector 1 and 2 at the same time.



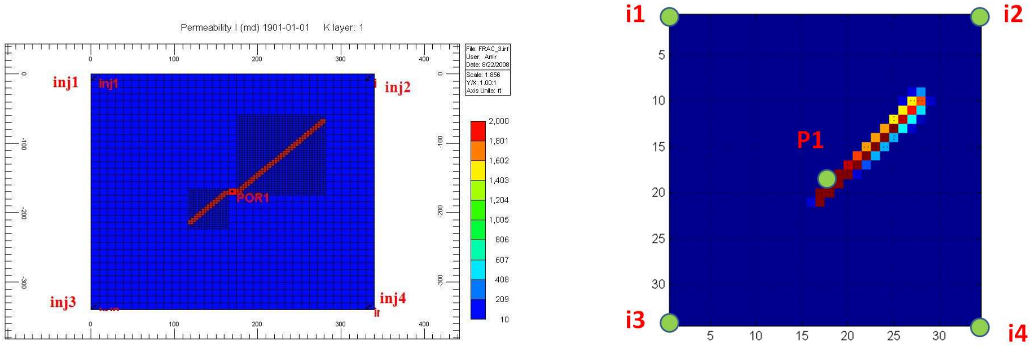
(a) $h_{11}[k]$

(b) $h_{21}[k]$

(c) $h_{31}[k]$

(d) $h_{41}[k]$

Figure 3.4: Simulation 1, the estimated FIR model for four injection-production pairs. After fitting with CM, the time delay constants are $\tau_{11} = 4.17$, $\tau_{21} = 1.87$, $\tau_{31} = 3.47$, $\tau_{41} = 3.96$.



(a)

(b)

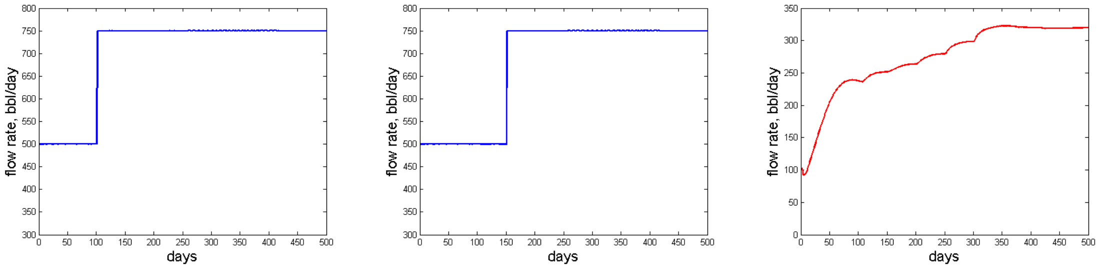
Figure 3.5: Simulation 1, the reservoir model. (a) Ground truth and (b) estimated probability map.

Compared to the PN sequences in Simulation 1, the response can be easily estimated from the change of production rate, because we only change one injection rate at each time that enables us to isolate the response of each well pair. However, the drawback is that it requires higher average injection rate, which might be limited by the pumping facility and the pressure limit. Moreover, the experiment duration is much longer because we have to change the injection and estimate the response one by one.

Different from the Simulation 1, now we have multiple producers, thus, the injected water can flow to any one them. In our reconstruction algorithm, we only use the travel time data such that corresponding inter-well connectivity is greater than 10%, i.e., the flow from injector i to producer j corresponds to at least 10% of total injection in injector i. Thus, the measured travel times $t_{1-3}, t_{1-5}, t_{2-3}$ are not considered because their corresponding inter-well connectivities are too low. In the object model, we choose the number of objects is equal to 3, which implies there are at most 3 fractures in the filed.

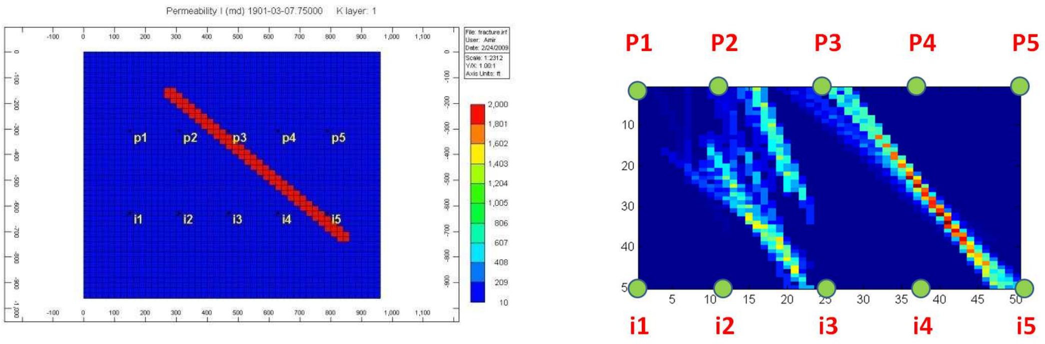
We show the estimated probability map in Figure 3.7, which gives high probability in the region between injector 5 and producer 3. Note that the probability map also shows the possible existence of high permeability channels in the region between injector 3 and producer 2, which is different from the ground truth in simulator. This comes from the non-uniqueness of the solution to our inverse problem - many different velocity models may all fit the measured travel time data well. Our algorithm can identify several possible models: in this case, one possible model is a single high permeability channel from injector 5 passing through the region between producer 2 and 3. Another possible model has a combination of one high permeability channels from injector 5 passing through the region between producer 3 and 4, and a shorter one in the region between injector 3 and producer 2. We can apply additional regularization criteria to choose between possible solutions. For example, we can select the model that fits the measurements with minimum fracture length, or use prior probability to favor some models. In this case, if we choose minimum length as additional regularization, the result will be a single fracture between injector 5 and producer 3.

In **Simulation 1** and **2**, we use PN sequences and step functions as injection rates, respectively. We are able to estimate the travel time in both cases, moreover, the estimated probability maps are consistent with the reservoir model in simulators. The major difference is that when we use the step function as input, we only change one injection rate at a time. Thus, we can use the change of production to map the response between well pairs directly. However, in such case we estimate the response for each injector one by one, thus, it takes much longer for experiment duration. On the other hand, by using PN sequences as the input, we can do a MIMO system identification and estimate the response between all well pairs at the same time, which dramatically reduces the experiment duration.



(a) (b) (c)

Figure 3.6: Simulation 2 (a) injection rate in injector 1 (b) injection rate in injector 2 (c) measured production rate in producer. Note that only one injection rate is changed at a time, thus we can estimate the response with respect to the change of injection directly from the increase of production



(a)

(b)

Figure 3.7: Simulation 2 (a) The ground truth and (b) estimated probability map. We successfully detect the high permeability channel between injector 5 and producer 3.

3.5.2 Field Experiment

We also conducted a field experiment in a a mature, low permeability fractured oil reservoir, which has been under water-flooding for many years. The pilot area had 12 injection, 3 production wells and the well installation pattern was a line-drive, which is roughly parallel to the direction of natural fractures estimated from seismic survey.

Two types of systems were used in the field to measure the fluid production: One is “well-testing”, which diverts the production from a well to a three phase separator. After a period of time, the accumulated fluid level can be measured in the separator and the average production rate is estimated. However, due to the cost, many wells share the same separation facility and these wells are tested sequentially. This fact limits the sampling frequency of the production rates: the system must measure each well one by one, and the system sampling period will be the sum of each testing period for different production wells. In this field, we usually have one measurement point every two weeks when well-testing is used, which makes the sampling rate for production data very low.

Another data source is the pump-off control (POC) data, which measures the load of each pumping unit. If we continuously monitor the load change in every stroke, we can calculate the daily production flow in real time [32]. However, the POC data needs to be calibrated with other references, because the pumping efficiency may change over time. In this application, we estimate the pumping efficiency by calculating the ratio of POC and well-testing data every two weeks, and use it to calibrate the POC data. The calibrated POC data is used as the daily production in the field.

For the field trial, we choose to estimate the CM parameters with the direct approach. The reason is that the amount of injection-production data is very limited, thus, the multi-stage approach is not appropriate because the number of FIR coefficients will be much larger than the measured data. In some days, due to the power outage or loss of data transmission, we do observe unusually low production rates and should treat them as outliers. Thus, to reduce the effect of these outliers, we choose L_1 instead using L_2 norm as the data fitting error metric. It is well known that L_2 norm is severely influenced by these outliers, while the L_1 norm provides a better balance between fitting good data points and outliers. From the above discussion, we know that the direct estimation has fewer unknown variables but requires to solve a nonlinear optimization. Thus, the estimated CM might lead to a local minimum and not model the system response well. To verify the estimated result, we use cross-validation [27] which divides the measured data into two parts. The first part is the training set, which is used to estimate the CM parameters. The second part is the testing set, where we use the estimated CM to predict the production rate and compare with the measurements (see Figure 3.8). If the prediction error is too high, we claim that this estimated CM is not accurately modeling the system response and re-do the nonlinear optimization with a different initialization until the error is below a threshold.

To simplify the problem, we partition the pilot area into upper and lower areas and estimate the travel time between the well pairs separately. The reason we did not perform the estimation for well pairs in the whole field is that this would require solving a nonlinear optimization with many more unknown variables. In a line drive water-flooding scenario, it can be assumed that the produced fluid is only affected by the injections from the nearest two rows of injection wells. Thus, we divide the whole field into two sub-fields and solve two smaller problems instead.

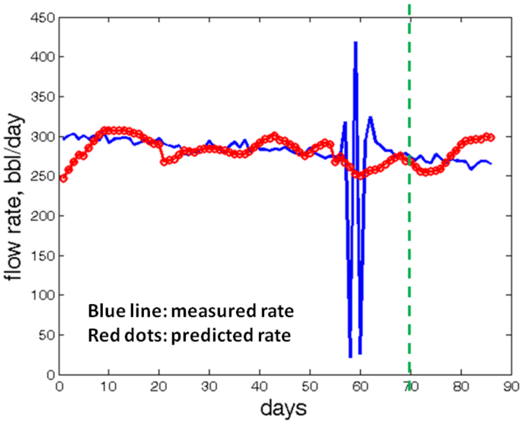
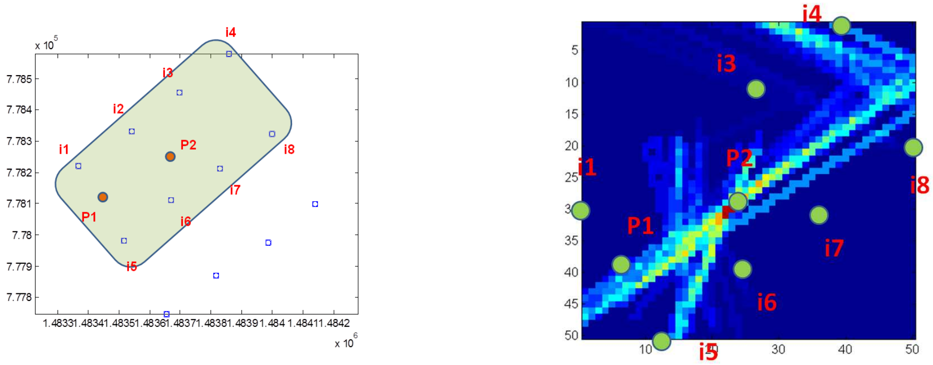


Figure 3.8: Field trial. We use the calibrated POC data and partition $1-70$ as training set, $71-85$ as testing set. Because L_1 norm is chosen as the error metric, the predicted output is not affected severely by outliers.

After estimating the CM parameters, we follow the same approach in Simulation 2 and use only these travel times corresponding to at least 10% of interwell connectivity. In the upper part of pilot area, the reliable estimated travel times are t_{1-1}, t_{3-2}, t_{4-2}, t_{5-1}, t_{5-2}, t_{6-1}, t_{7-1}, t_{7-2}, t_{8-1}, t_{8-2}. We show the probability map in Figure 3.9, which indicates that the high permeability channels are very closed to the production well 2 and roughly parallel to the installation of wells. This prediction is consistent with the fact that production well 2 is a 24 hours run, high production well, which implies there could be high permeability channels in nearby areas. In the lower part, the estimated travel times are $t_{7-3}, t_{9-3}, t_{10-3}, t_{11-3}, t_{12-3}$ and we show the results in Figure 3.10.

The combination of the two results is shown in Figure 3.11, with the comparison of the known fault location provided by previous seismic survey. Note that in the upper/lower area, we only estimate the probability map for the region between the wells (transmission tomography can only estimate the interior structure between



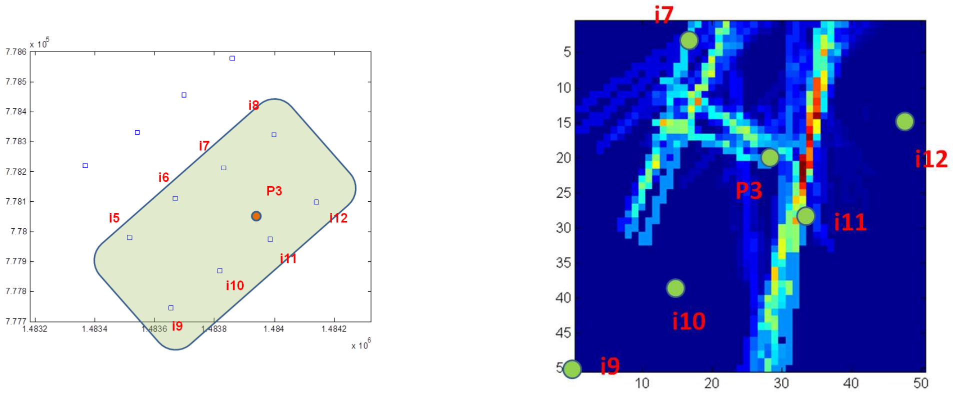
(a) (b)

Figure 3.9: Field trial, upper area. (a) The location of wells and (b) estimate probability map. Note that the high probability areas is roughly parallel to the installation of the wells, which is consistent to the seismic survey for the nature fracture direction

sensors). For instance, we see a sudden “cut” for possible fracture models in the right side area of injection 8. The reason is that area is already “out of bound”, thus the probability is set to zero. When we compared with the seismic survey, our result shows that the high probability areas are roughly parallel to the well installation, which agrees with prior knowledge. Also, we note that there is a known fault near injector 12 and our result also shows low appearance probability in that area.

3.6 Conclusion

In this chapter, we propose a novel method to detect high permeability channels in a water-flooding reservoir without additional instrument or altering the average daily production. This method uses the change of injection rate as the active probing signal and detects the field production changes in real-time, then estimates the equivalent travel time from the injection-production data. With the estimated travel time, we can formulate the detection of the high permeability channels as a



(a)

(b)

Figure 3.10: Field trial, lower area. The (a) location of wells and (b) estimated probability map.

high contrast travel time tomography and apply the algorithm described in Chapter 2 to solve it.

We validate our approach using both a simulator and also a field experiment. In order to apply our algorithm to real field data, we note some practical issues. First, the data sampling frequency and the data quality need to be considered. Limited by the field measuring system, we usually have very reliable daily injection rates, but production rates are often obtained from bi-weekly well-testing. The low sampling frequency of production rate reduces the time resolution of estimated travel times. Therefore, we try to increase the sampling rate by using the POC data (calibrated with well-testing data) to get the daily production rate.

The second issue we encounter in practical applications is the distribution of well locations, which is related to spatial resolution. It is well known that the spatial resolution for the reconstructed image is proportional to the density of travel paths. For a small variation velocity model, the travel path is almost a straight line between transmitter-receivers, thus, the density of travel paths can be calculated by the location of wells. However, in the high velocity contrast case the travel path

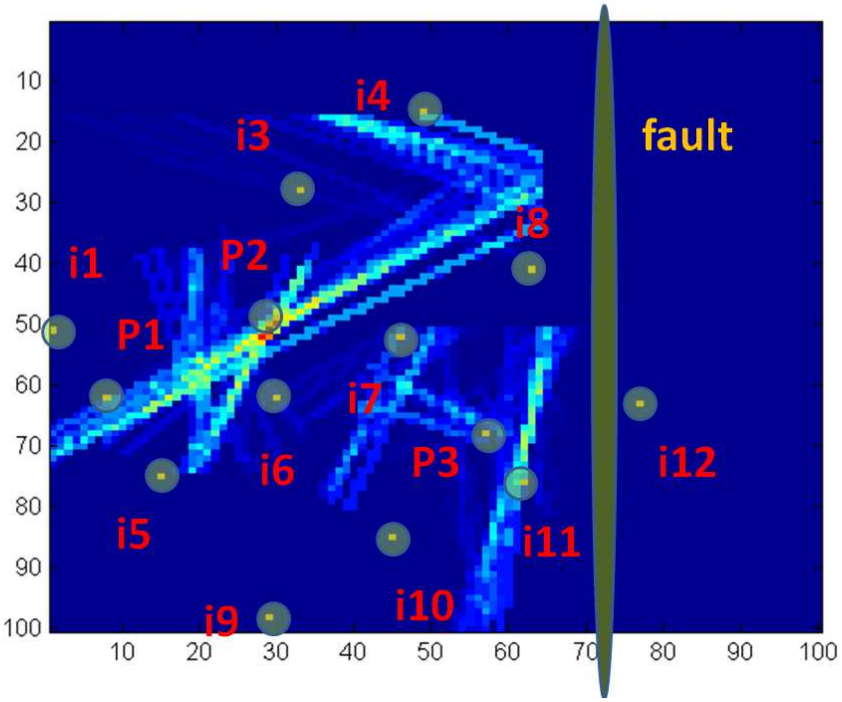


Figure 3.11: Field trial, the estimated probability map with a known fault. Our result shows the appearance probability is very low in the fault zone, which agrees with the prior survey.

bends severely near the high velocity objects, thus the density of travel paths cannot be estimated as in the previous case. In particular, the travel path may show a "jump" effect depending on the location of high velocity objects, which makes the estimation spatial resolution difficult. For example, assuming there is a small high velocity object near the transmitter-receiver pair, in such case the travel path is a straight line. When the object size increases, the travel path will "jump" from a straight line to a bent path through the object, thus, this object is "invisible" until it reaches certain size. In order to detect the high permeability channel in arbitrary direction, we prefer for the wells to be uniformly located in the field and to cover all angles, which may not be the case in a real field. We are currently studying this problem, and plan to address it in the future work.

	Simulation 1	Simulation 2	Field trial
Injection schedule	PN sequences	step functions	PN sequences
Experiment duration	64 days	400 days	90 days
Production data type	Exact	Exact	POC
Estimation approach	Multi-stage	Multi-stage	Direct

Table 3.1: Table for the method comparison in simulations and field trial

The estimated result is presented as a probability map to visualize the probability of the appearance for high permeability channels in different area. Future work needs to focus on how to combine the geological information in the result, and consider not only the natural fractures, but also the faults in the field.

Chapter 4

X-Ray Discrete tomography

4.1 Introduction

In this chapter, we investigate high contrast transmission X-ray tomography, which is widely used for noninvasive testing in medical imaging and industrial material inspection. In these applications, a series of X-rays target the object from different directions, and the X-ray energy is measured along the object boundary. The travel path of an X-ray can be assumed to be a straight line, and the decrease of energy is modeled as a line integral of the attenuation along the travel path. With the measured energy data, an image of the attenuation distribution within the object can be computed through the reconstruction algorithm.

It is well known that if we increase the amount of measured data, we can always achieve better reconstructed image quality. But for X-ray medical imaging, we want to keep the number of X-ray projections as small as possible. The major concern is the radiation doses for the patient: in angiography [39], a typical cardiac CT scan needs to be coupled with ECG-gating to synchronize the recording of data, which means that the patient is exposed to high energy X-ray for the entire cardiac cycle. To lower the radiation dose, we would need to reduce the number of X-ray projections, which limits the amount of data. Another scenario is when we can only measure the object from certain angles, e.g, in dental CT [59] the scanning angle is limited. Thus, we want to study the problem of how to improve the quality of reconstructed image based on a limited amount of data.

In this chapter, we consider the case where the object is assumed to be made of a few different materials (corresponding to very different X-ray attenuation coefficient). Thus, the target image can be segmented into several high contrast regions, and we can approximate the target image by assigning the same value inside a given region. We state our problem as follows: with limited amount of data, we want to reconstruct a high contrast image where there are only few distinct pixel values. This problem is called discrete tomography (DT) [36]. Note that in many problems of interest, we apply tomography techniques to reconstruct a digital image: the main difference in our case is that we expect that only a few of those levels may appear in the reconstruction. For example, we may have 8 bit resolution in intensity levels, but only a small subset of the 256 possible ones can be found in the reconstructed image. There are many practical examples of discrete tomography: For example, in industrial CT scanning to detect material flaws, we know the material compound of the object (aluminum, plastic, metallic composite, etc) and want to detect the existence of possible fractures in a reduced time (less amount of sampled data). In such case, the attenuation coefficients of the materials are known as a prior, and the fracture can be viewed as an empty space (almost zero attenuation for air). The reconstructed image can be approximated by a image with a few different intensity values, where the high pixel values correspond to the composite materials in the object, while near zero values correspond to the background or fracture parts. As another example, in angiography, we want to reduce the radiation doses and a contrast agent (Iodine) is injected to the blood vessel.

Thus, the area where blood flows through will have much higher X-ray attenuation compared to other body tissues and the result is also close to a two level intensity image.

Different from typical X-ray tomography, which can be formulated as a linear inverse problem, in DT the possible intensities are limited to a few discrete values. When the possible intensity levels are known as a prior, DT becomes a linear integer inverse problem, for which many algorithms have been proposed. One approach is to start from continuous reconstruction methods then force the solution to converge into discrete values. Batenburg [7] proposed a discrete algebraic reconstruction algorithm (DART) to iteratively update the object boundary. Fishburn [30] and Weber [74] used linear programming (LP) relaxation for this integer programming problem, and use a convex-concave regularization to enhance the discrete value solution. Another approach is to work on the discrete values directly: In combinatorics, researchers [46] are interested in reconstructing a special class of images, hv-convex objects, with only horizontal, vertical, and diagonal projection angles. For a comprehensive review, we refer to [36].

However, *none of the above methods has exploited the sparse nature of images having only a few distinct intensity values, with each region having a constant intensity and smooth boundary.* We propose to solve DT problem in the transform domain by exploiting sparseness properties of reconstructed images of interest in the transform domain. Specifically, we observe that piecewise constant images with a few gray levels have a sparse representation as linear combinations of unit step functions. For example, we can decompose these images in terms of step functions in vertical or horizontal axis. The step functions form a dictionary, which can be used to represent these images with sparse set of coefficients because step functions are needed only at the boundaries between regions.

In recent years a significant amount of work has been devoted to sparse signal recovery, e.g., in the context of compressed sensing [18]. It has been proved that with high probability, a sparse image can be perfectly reconstructed by a small amount of random projection data. But only limited effort has addressed scenarios where the data to be reconstructed is discrete in terms of intensity levels [66]. In particular, no work has considered how to take advantage of the potential sparseness of 2D images with only a few distinct intensity levels under certain transformations.

In this chapter, we start by considering a binary, known-level DT problem. We design a dictionary with step functions on "stripes" along different angles, thus, these atoms can form rectangles for different angles and positions by using two atoms to represent the boundary. If a binary image has smooth boundaries, this dictionary should provide a sparse representation, namely, only a few non-zero coefficients (few rectangles) will be required when representing images using the dictionary. Moreover, we prefer a specific type of representation: we want to represent a image with "non-overlapping" rectangles. The main reason is that we know the intensity levels as a prior. Thus, if each pixel is represented only by one rectangle, the coefficient for the dictionary representation also belongs to one of the known few discrete levels. For example, if we use step function on horizontal stripes as dictionary to represent a binary image with intensity levels $\{0, 1\}$, the intensity of the horizontal rectangle should be equal to 1. Thus, the coefficients in dictionary will belong to $\{0, -1, 1\}$ to form a rectangle with intensity 1. In particular, if the pixel intensity belongs to $\{0, k_i\}$, the coefficients in the dictionary should belong to $\{0, -k_i, k_k\}$ which preserve the "few discrete level" property.

From the above discussion, we can formulate the binary DT as a sparse recovery problem, in which the goal is to recover a sparse coefficient vector within known possible values. To reconstruct a binary image that fits the measured data, we have two constraints: (i) data fitting, and (ii) binary pixel intensity. Based on the linear programming (LP) relaxation, we relax the binary pixel intensity constraint, $\{0, 1\}$ into a convex region $[0, 1]$. With the designed dictionary, we use L_1 norm of the coefficients as a heuristic regularization for searching a sparse representation. The cost function will be convex, which implies that we can solve the problem efficiently using convex optimization.

Because we use linear relaxation to convert an integer programming problem into a linear programming problem, the reconstructed image may not have exactly binary intensity levels. To encourage the reconstructed image to reach binary intensity and have a sparser representation, we modify the reweighed L_1 minimization algorithm [19]. The reweighed L_1 algorithm uses majority-minimization to search a sparse representation in the transformed domain. From above discussion, we prefer a sparse representation with coefficients having intensity levels in $\{0, -1, 1\}$. Thus, if the representation has coefficients with non-integer value, we randomly modify the weight of the reweighed L_1 norm for those coefficients corresponding to non-integer values. This will help the solution jump out of a current local minimum and search for our preferred integer solution. By running this randomized reweighing, the solution has higher chance to converge to an integer solution. Simulation results show that our method does reconstruct an image with sparser, integer representation compared to the results without randomized reweighing.

Next, we extend our approach from the binary DT problem to a multi-level DT problem. This can be done by defining a multi-level image as a superposition of several binary-level images, which can be represented by rectangles. In particular, we prefer a representation that all the rectangles will not overlap with each other. Following the same approach as we proposed before, each binary level image will have a sparse representation under the dictionary. Thus, to extend a binary DT problem to an $M+1$ level DT problem, we increase number of coefficients to M times from the case in binary DT to recovery a $M+1$ level image with a superposition of M binary level images. For example, in a $M+1$ level DT problem with intensity levels $\{0, k_1, \ldots, k_M\}$, we can represent the image as $\mathcal{I} = k_1 \cdot \mathcal{I}_1 + \cdots + k_M \cdot \mathcal{I}_M$, and we recover the sparse representation for binary images $\mathcal{I}_1, \ldots, \mathcal{I}_M$ at the same time by running a sparse recovery with M times of unknown variables.

A common assumption for DT algorithms is that the set of possible intensity levels is known in advance. However, in practice determining the intensity levels is very challenging and coupled with other aspects of the problem, such as measurement calibration. For example, in angiography the target image will be very high-contrast, but the intensity level of blood vessels is very hard to determine *a priori* since it is related to the blood flow and the chosen contrast agent [16].

To the best of our knowledge, only a few authors explicitly address the *unknown intensity discrete tomography problem.* Different from known intensity DT problems, the unknown intensity increases the difficulty of reconstruction significantly. Not only we need to identify the discrete intensity level corresponding to each pixel, but we also need to estimate the set of possible levels. One possible approach is to build the intensity estimation based on the existing known level DT algorithms: Batenburg *et al.* [8] proposed a semi-automatic algorithm for intensity level estimation, which requires the user to select manually regions that are expected to belong to the same gray level. Another approach is to modify the cost function in DT to encourage it to converge to few intensity levels: Lukić [51] combined the multi-well potential function into the object function in order to encourage the solution staying on gray level values, but it is not easy to design the potential function without the solution being trapped in a local minimum.

We propose a *completely unsupervised* estimation algorithm for the unknown intensity level case that jointly reconstructs the image and estimates the intensity levels. We extend our binary DT reconstruction algorithm by (i) adding a level-estimation step within each iteration and (ii) allowing the superposition of multiple levels. For the ideal reconstructed discrete image, the histogram of pixel intensity levels should only have a few peaks. Thus, our proposed level-estimation step is essentially a clustering algorithm on the histogram of the reconstructed image.

In our DT reconstruction algorithm, we need to solve a convex optimization for sparse recovery which could be very computational expensive for high resolution images (corresponding to many unknown variables). The state of art convex solvers usually use second-order optimization methods (e.g, interior point method [56]) which require calculating the Hessian matrix. In high dimensional problems, this leads to very large time and memory requirements to store and compute the Hessian matrix (which has a size of N^2, where N is the number of variables) and makes the second order solver impractical. Thus, we use a first order gradient method, projection onto convex sets [29] (POCS), to solve this problem with reduced memory requirements. The first order methods only use the gradient of the cost function, and thus, the memory required grows linearly with respect to the number of unknown variables. The drawback is that POCS requires many more iterations to converge to the global minimum.

The rest of this chapter is organized as follows: In Sections 4.2 and 4.3, we present the formulation of the discrete tomography problem and the proposed dictionary representation. In Section 4.4 we explain our sparse reconstruction algorithm with the intensity level estimation. In Section 4.5, we present our reconstruction results from noiseless and noisy measurements and provide performance comparisons. In Section 4.6 we conclude this work and discuss some further directions.

4.2 Problem Definition

In this section, we formulate the DT problem in the 2D case. Extension to the 3D case will be straightforward. Let $f(x_1, x_2)$ represent the attenuation distribution of a 2D object we want to estimate, and assume we measure the energy decrease along parallel X-ray projections with different angles. Because the travel path of X-rays can be assumed to be a straight line, the measurements can be viewed as the integral of f along a straight line (see Figure 4.1). The relationship between the measurements and object f is modeled by Radon transform:

$$P_{\theta,f}(t) = \int\!\!\int_{-\infty}^{\infty} (x_1, x_2)\ \delta(x_1 \sin(\theta) + x_2 \cos(\theta) - t)\ dx_1\ dx_2. \tag{4.1}$$

Note that the Radon transform is a linear transform. If we discretize the object using a grid-based model, namely, assigning the same parameter within a given cell, we can represent the object by an image, where each pixel value corresponds to the physical property inside one cell. We assume the measured X-ray energy decreases as a function of the sum of all pixels in the given ray passes through, and a trigonometric interpolation is used for the non-grid points. If the image has $p \times q$ pixels, we can reshape the $2D$ image f into a $1D$ vector $\mathbf{x}$ with dimension $n = p \cdot q$. For each projection angle, the sampling distance between parallel X-rays (δt in Figure 4.1) is set equal to the length between neighboring pixel centers. If

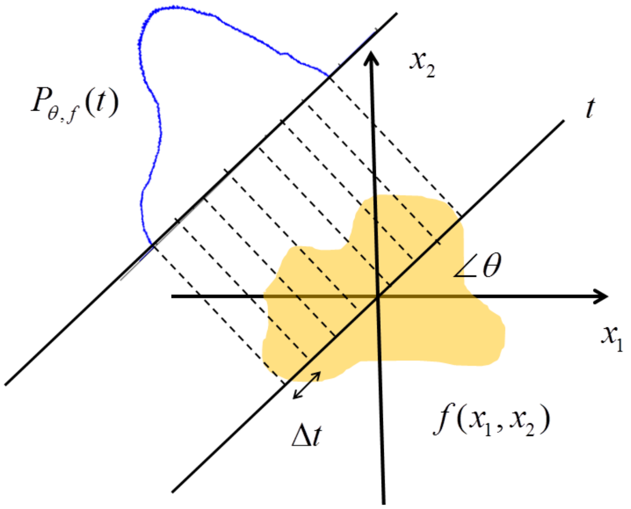


Figure 4.1: Example for the Radon transform. The measurement $P_{\theta,f}(t)$ is a line projection of $f(x_1, x_2)$ along angle θ.

we take γ samples, the measured data for one specific angle will be a γ-dimensional vector. In particular, we can write this operation into a matrix form as:

$$\mathbf{W}_\theta \cdot \mathbf{x} = \mathbf{p}_\theta, \tag{4.2}$$

where $\mathbf{W}_\theta$ is a $\gamma \times n$ line projection matrix with angle θ. If we have d different viewing angles, we can cascade the projection matrices and measurements as:

$$\mathbf{A}\mathbf{x} = \mathbf{y}, \tag{4.3}$$

with

$$\mathbf{A} = \begin{bmatrix} \mathbf{W}_{\theta_1} \\ \vdots \\ \mathbf{W}_{\theta_d} \end{bmatrix}, \mathbf{y} = \begin{bmatrix} \mathbf{p}_{\theta_1} \\ \vdots \\ \mathbf{p}_{\theta_d} \end{bmatrix}. \tag{4.4}$$

$\mathbf{A}$ will be a line projection matrix with size $m \times n$ that maps $\mathbf{x}$ into measured data $\mathbf{y}$, where $m = \gamma \cdot d$. A given entry A_{ij} represents the effect of pixel j on measurement i, and corresponds to the path length of the projection line inside the square region of pixel i.

The reconstruction problem is: Given the projection matrix $\mathbf{A}$ and measurements $\mathbf{y}$, where the number of measurements is far less than the number of unknown variables, $m \ll n$, we would like to find a solution that has few distinct intensity levels, $x_i \in \{0, k_1, \ldots, k_M\}$, such that its predicted projection will match the measurements $\mathbf{Ax} = \mathbf{y}$, where k_i is unknown and needs to be estimated, but the number of distinct intensity levels is known. Note that the possible solution set is discrete so that this is an linear integer inverse problem. In the next section, we will describe how to define a sparse representation for an image with only few distinct intensity levels and use this as regularization to solve the inverse problem.

4.3 Dictionary Representation

In the previous section, we define the reconstruction in DT as an inverse problem. However, we mainly focus on the cases when only limited data is available, thus, this inverse problem is ill-posed. The key to solve an ill-posed inverse problem is to find ways to constrain the possible solutions in order to favor solutions with desirable properties, based on available prior knowledge. In particular, we will identify representations such that the objects of interest can be described with a small number of coefficients (i.e., the representation is sparse). Once a representation that favors sparsity has been defined, we can use it as an additional regularization constraint to obtain a better solution. In our problem, we focus on images that have only few distinct intensity levels, and are composed of regions with constant intensity levels and smooth boundaries. Thus, we want to design a dictionary that can represent this type of images with very few pre-defined atoms. To characterize the "few distinct intensity levels" and "smooth boundary" properties, we choose unit-step functions along different directions as atoms and decompose the image on the discontinuities, which provides a sparse representation.

We define the 2D continuous basis function $\psi_{\theta,\alpha,\beta}(x_1, x_2)$ as the unit-step functions along the parallel lines, $x_1 \cos(\theta) + x_2 \sin(\theta) = \alpha$, with the angle θ, spacing distance between parallel lines α, and the shift β on the discontinuity of unit step function. With the delta function and step function, the basis function can be written as:

$$\psi_{\theta,\alpha,\beta}(x_1, x_2) = \delta(x_1 \cos(\theta) + x_2 \sin(\theta) - \alpha) \cdot U(x_1 \sin(\theta) - x_2 \cos(\theta) - \beta) \tag{4.5}$$

$$\delta(x) = \begin{cases} +\infty, & x = 0 \\ 0, & x \neq 0, \end{cases}, \quad \int_{-\infty}^{\infty} \delta(x)\, dx = 1, \quad U(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0. \end{cases} \tag{4.6}$$

In other words, we represent a image $f(x, y)$ with parallel "stripes" along the angle θ, where the stripes is defined by the unit step functions. The mathematical formulation can be written as:

$$f(x_1, x_2) = \sum_{\alpha,\beta} g_{\theta,\alpha,\beta} \cdot \psi_{\theta,\alpha,\beta}(x_1, x_2).$$

Following the same concept, we can also define the atoms for a 2D function with discrete variables, namely, $f(x_1, x_2), x_i \in \mathbb{N}$. Then we can represent a image with unit step functions along arbitrary angle θ. For example, if we choose angle $\theta_1 = 0$, the decomposition of the $2D$ binary image f will be:

$$\begin{aligned} f(x_1, x_2) &= \sum_{\alpha,\beta} g_{\theta_1,\alpha,\beta} \cdot \psi_{\theta_1,\alpha,\beta}(x_1, x_2) && (4.7) \\ &= \sum_{\alpha,\beta} g_{\theta_1,\alpha,\beta} \cdot \delta(x_2 - \alpha) U(x_1 - \beta). && (4.8) \end{aligned}$$

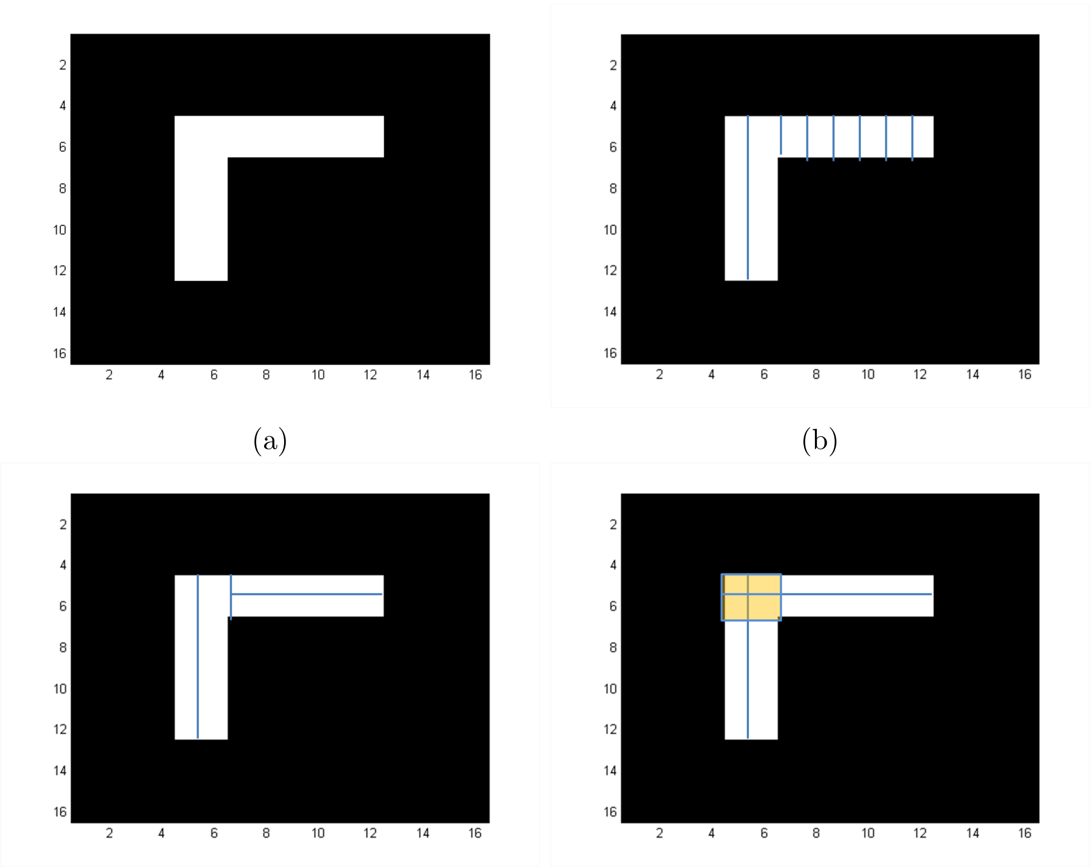
In particular, $g_{\theta_1,\alpha,\beta}$ decomposes $f(x_1, x_2)$ along the x_1 axis with unit step function, thus, it "marks" the discontinuities along the horizontal direction.

Since vector $\mathbf{x}$ is a reshape of image f, we can also reshape the coefficients $g_{\theta,\alpha,\beta}$ into a $1D$ vector $\mathbf{u}_\theta$ in the same way. Following the same approach, we can define two basis with different angles θ_1, θ_2. These two transforms are linear and invertible, that is $\mathbf{x} = \mathbf{T}_{\theta_1}\mathbf{u}_{\theta_1} = \mathbf{T}_{\theta_2}\mathbf{u}_{\theta_2}$ which provide decomposition of $\mathbf{x}$ along two different angles θ_1 and θ_2. Moreover, we can cascade two basis to form our over-complete dictionary, $\mathbf{D} = [\mathbf{T}_{\theta_1}, \mathbf{T}_{\theta_2}]$, and the signal $\mathbf{x}$ can be represented by an even sparse vector $\mathbf{u} = [\mathbf{u}_{\theta_1}, \mathbf{u}_{\theta_2}]^t$:

$$\mathbf{x} = [\mathbf{T}_{\theta_1}, \mathbf{T}_{\theta_2}] \begin{bmatrix} \mathbf{u}_{\theta_1} \\ \mathbf{u}_{\theta_2} \end{bmatrix} = \mathbf{D}\mathbf{u}. \qquad (4.9)$$

This dictionary can be extended to include more basis along different angles to have a sparser representation.

When we have an over-complete dictionary, we will have multiple possible representations for a binary image. In particular, we prefer the representation that the binary image is represented by disjoint stripes. The reason is since the possible intensity level is $\{0, 1\}$ and every pixel is represented only by one stripe, under this representation the intensity for stripes must be 1 and the possible coefficients



(a)

(b)

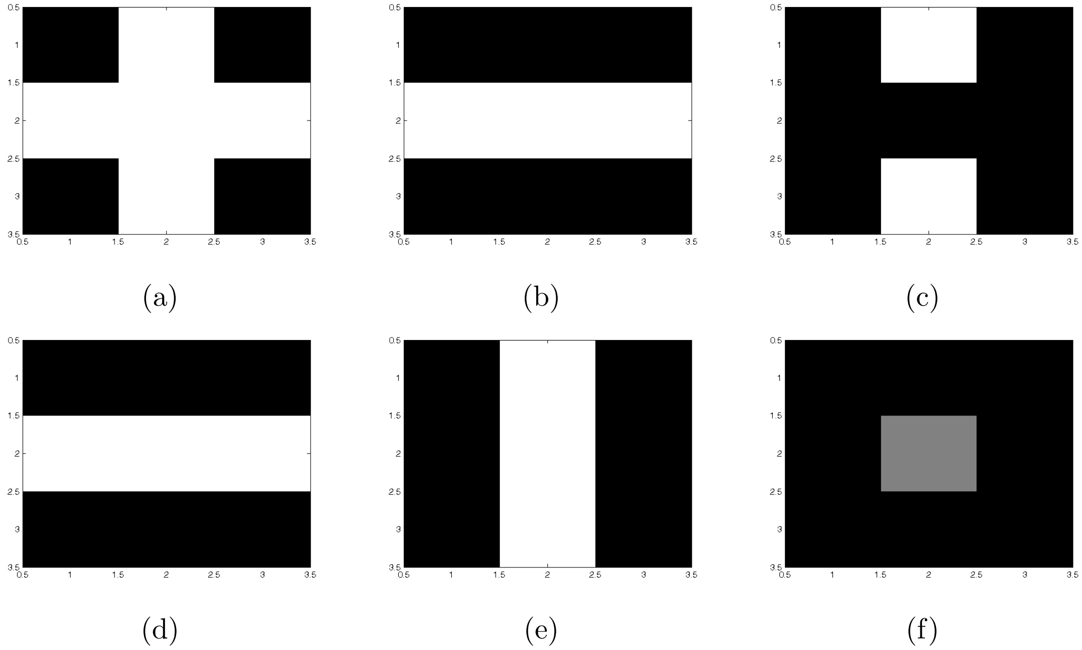
(c)

(d)

Figure 4.2: (a) Binary Phantom (b) Decomposition along the y axis (c) Decomposition using the dictionary along both x and y axis (d) Another possible representation when stripes are overlapping. Note that we prefer a representation where stripes tend not to overlap with each other.

must belong to $\{0, -1, 1\}$. Thus, this presentation may not provide the most sparse representation, but it preserves the possible intensity levels (see Figure 4.3). This property imposes additional constraints on the selection of an image model.

From the above discussion, we show how to represent a binary image efficiently with our designed dictionary. Next, we extend our approach to represent images with a few distinct intensity levels. Note that a multi-level image can be represented as a combination of many non-overlapping two level images. For example, if the



(a) (b) (c)

(d) (e) (f)

Figure 4.3: Sparse representation example: (a) The image we want to represent (b,c) one possible representation of this image with non-overlapping strips (d,e,f) Another possible representation, two over-lapping line function and subtract the overlapping part. We prefer (b,c) as our representation.

image pixel intensity belongs to $\{0, k_1, \ldots, k_M\}$, the signal $\mathbf{x}$ can be represented as a combination of at most M two-level signals:

$$\begin{aligned}
\mathbf{x} &= k_1 \cdot \mathbf{x}^1 + \ldots k_M \cdot \mathbf{x}^M && (4.10)\\
&= \sum_{j=1}^{M} k_j \cdot \mathbf{x}^j && (4.11)\\
&= \sum_{j=1}^{M} k_j \cdot \mathbf{D}\mathbf{u}^j && (4.12)
\end{aligned}$$

where $\mathbf{x}^j$ is the segmentation of different intensity levels and each entry of $\mathbf{x}^j$ belongs to $\{0, 1\}$. From (4.9), we know that a binary signal $\mathbf{x}^j$ can be represented by a dictionary $\mathbf{D}$ with a sparse coefficient vector $\mathbf{u}^j$, where the stripes that represent $\mathbf{x}^j$ do not overlap with each other. In particular, $\mathbf{x}$ can be represented by non-overlapping two level regions, where each region can also be represented by non-overlapping stripes (see Figure 4.4). The advantage of choosing this representation is that it preserves the "few distinct level" property for the coefficients. However, the drawback is that a coefficient of dimension M is needed to represent each pixel.

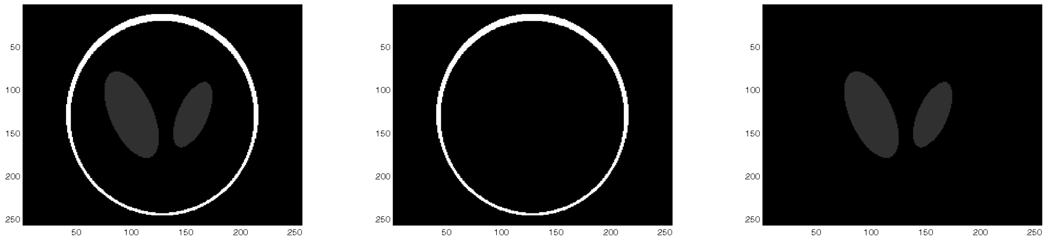


Figure 4.4: A multi-level phantom. We represent it by non-overlapping regions with different intensity.

4.4 Proposed Algorithm

In the previous section, we show how to efficiently represent a smooth boundary image with a few discrete intensity levels, using a pre-defined dictionary. Thus, if we are interested in this type of images, we can formulate the DT as a sparse recovery problem and use the dictionary representation as regularization in solving inverse problem. Our proposed algorithm iteratively recovers the sparse representation of the signal and estimates the possible intensity levels. It can be separated into two parts: In the first part, for given possible intensity levels, we recover the coefficients in dictionary representation by solving the reweighted L_1 minimization. In the second part, we run a clustering algorithm on the histogram of the reconstructed image to update the set of possible intensity levels, then return to the first step. Figure 4.5 shows the flowchart of our algorithm.

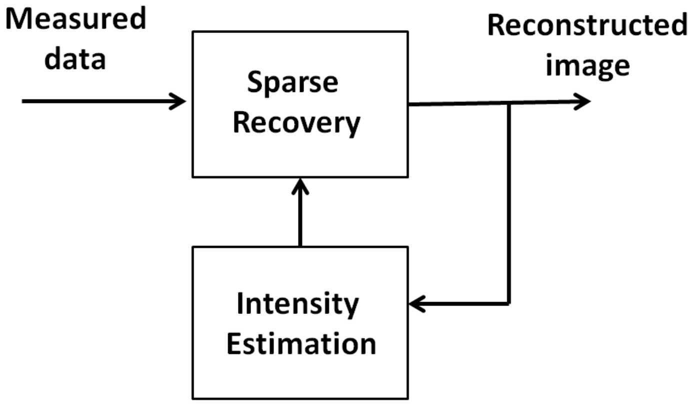
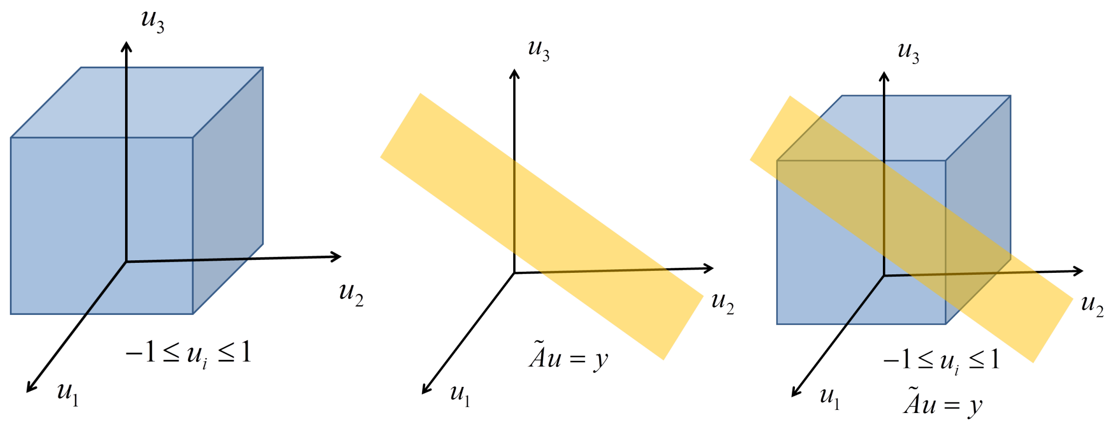


Figure 4.5: Flowchart of our algorithm. We iteratively perform sparse recovery and intensity estimation.

4.4.1 Sparse recovery with known intensity levels

Given intensity levels, we start to build our sparse recovery algorithm in a binary case, then we extend this algorithm to multi-level scenarios. When the intensity levels are known, namely, a binary image has intensity levels in $\{0, 1\}$. With the dictionary we described in Section 4.3, we can transform the image of interest and represent it with very few coefficients. Therefore, DT can be formulated as a sparse recovery problem, where the reconstructed image is binary and has a sparse representation using the dictionary. However, the "binary pixel" constraint means that the solution space becomes discrete, and the recovery problem will be an integer programming problem. Thus, we use linear relaxation to relax the integer constraint into a convex region constraint, and transform the NP-hard optimization (integer programming) into a related problem which can be solved in polynomial time (linear programming). Instead of searching for an integer solution, we search for a solution in a continuous feasible region. With our designed dictionary, we can transform the possible intensity region into $[-1, 1]$ in the transformed domain. In particular, the possible intensity region will be a hypercube (see Figure 4.6a). For the data fitting constraint, because the X-ray projection is linear and modeled by the measurement matrix, the feasible region for data fitting will be a hyperplane (see Figure 4.6b). Then the feasible region can be defined as the intersection of data fitting and intensity boundary constraints (see Figure 4.6). Now, this problem can be formulated as a sparse signal recovery with a new measurement matrix $\widetilde{\mathbf{A}} = \mathbf{AD}$.

$$\begin{aligned} \min \quad & \|\mathbf{u}\|_1 \\ \text{subject to} \quad & \widetilde{\mathbf{A}}\mathbf{u} = \mathbf{y}, \quad \widetilde{\mathbf{A}} = \mathbf{AD} \\ & \mathbf{0} \leq \mathbf{x} \leq \mathbf{1}, \quad -\mathbf{1} \leq \mathbf{u} \leq \mathbf{1}, \quad \mathbf{Du} = \mathbf{x} \end{aligned} \tag{4.13}$$



(a) (b) (c)

Figure 4.6: The feasible set for (a) intensity boundary constraint (b) data fitting constraint (c) intersection of data fitting and boundary. Note that the solution in DT must belong to the intersection to satisfy both constraints.

Candès and Tao [18] have shown that if the measurement matrix $\widetilde{\mathbf{A}}$ has the restricted isometry property (RIP), it is guaranteed that a perfect reconstruction of sparse signal $\mathbf{u}$ can be obtained. However, in DT the RIP property of the measurement matrix $\widetilde{\mathbf{A}}$ depends on the number of viewing angles and the chosen transform.

When we design the dictionary, we would like to choose $\mathbf{D}$ such that the signal $\mathbf{x}$ has sparse representation and the combined sensing matrix $\widetilde{\mathbf{A}} = \mathbf{AD}$ also has low coherence. In [26] joint learning of the dictionary and sensing matrix is considered. However, in the X-ray tomography problem, the sensing matrix is a line projection matrix and the coherence of the sensing matrix is fixed by the number of projections. In the joint learning approach, the learned dictionary will be a balance between the vectors that diagonalize the sensing matrix and the principal components of the training images. Thus, the atoms in the learned dictionary usually have non-zero entries all over the space and fail to provide a representation that preserve the "discrete intensity level" property. In our approach we rather choose a deterministic dictionary that can lead to reconstructed images with discrete intensity levels and use reweighed L_1 minimization to recover a sparse solution even for the high correlation of the combined sensing matrix.

Next, we extend this approach to images with multi-level discrete intensity. When the possible intensity levels k_i are given, in (4.10) we can represent a multi-level image $\mathbf{x}$ as the linear combination of several two-level signals: Thus, if k_j are given, the sparse reconstruction problem for multi-level DT can be written as

$$\begin{aligned}
\min \quad & \sum_{j=1}^{M} \|\mathbf{u}^j\|_1 && (4.14)\\
\text{subject to} \quad & \mathbf{y} = \mathbf{A}\mathbf{x}, \quad \mathbf{x} = \sum_{j=1}^{M} k_j \cdot \mathbf{D}\mathbf{u}^j && (4.15)\\
& -\mathbf{1} \leq \mathbf{u}^j \leq \mathbf{1}, \quad \mathbf{0} \leq \mathbf{x} \leq \mathfrak{U} && (4.16)
\end{aligned}$$

where $k_j \cdot \mathbf{D}\mathbf{u}^j$ represents the region with intensity k_j and $\mathfrak{U}$ is the boundary of feasible intensity values. The feasible region for the intensity values $\mathfrak{U}_\mathfrak{i}$ is initialized with the maximum possible intensity value $\{\max_j k_j\}$, and will be updated iteratively as the estimated intensity levels change.

After the sparse recovery, we will have the coefficients $\mathbf{u}^j$ corresponding to the intensity value k_j. Thus, we can define M segments as $\mathbf{z}^j = k_j \cdot \mathbf{D}\mathbf{u}^j, j = 1, 2, \ldots, M$, which maps the coefficients in the dictionary back to the 2D spatial domain. We say that pixel i belongs to segment z^j if the ith entry in z^i is greater than a small constant ξ. Note that under this definition, it is possible for one pixel to belong to several different segments. For perfect reconstruction, all the pixels in the jth segment will have the same intensity k_j and segments will not overlap with each other (which implies every pixel only belongs to one segment). Thus, if a pixel belongs to only one segment, $x_i \in \mathbf{z}^j$, the intensity boundary for a pixel can be assigned as $\mathfrak{U}_{\mathrm{i}} = k_i$.

However, when the sparse recovery is not perfect, it is possible that some segments are overlapping. Under this circumstance, a pixel x_i may belong to more than one segment. Thus, we use a "soft" decision rule and choose the intensity boundary as the maximum of intensity levels for all segments covering that pixel, namely, $\mathfrak{U}_{\mathrm{i}} = \max\limits_{j_1,\dots,j_p} k_j$ if $x_i \in \bigcap\limits_{j_1,\dots,j_p} \mathbf{z}^j$. By choosing a loose intensity boundary, we guarantee the existence of solution in the sparse recovery.

With the presence of noise, the measurements will be $\mathbf{y} = \widetilde{\mathbf{A}}\mathbf{u} + \mathbf{z}$, where $\mathbf{z}$ is the noise with power $\|\mathbf{z}\|_2 \leq \epsilon$. In order to operate with noisy measurements, we change the feasible region in (4.13) from $\mathbf{y} = \mathbf{A}\mathbf{x}$ to $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \delta$, where δ is the search allowance for noisy measurements. This approach is closely related to LASSO [70] and sparse Bayesian learning [78], where the goal is to achieve a good trade-off between data fitting and model complexity. Without knowing the noise power ϵ, selecting an appropriate δ is non-trivial. If we choose δ to be too large, then the hyper ball $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2$ will contain the origin $\mathbf{x} = \mathbf{0}$. In that case, the solution will be a zero vector which is the global minimum for reweighed L_1 norm. On the other hand, if δ is too small, the intersection of data fitting $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2$ and boundary constraint $\mathbf{0} \leq \mathbf{x} \leq \mathbf{1}$ may be empty. Thus, no solution can be found. Boufounos et al [13] have proposed a cross validation method to estimate noise power. With a good choice of δ, we can assume that the hyper ball $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2$ contains the actual solution we want to recover. For the noisy case, because we use L_1 norm as regularization, the solution is the one with minimum L_1 norm within feasible region. Moreover, even with correct search allowance δ, the solution will still not be exactly all integer. It will always to be on the inner bound of the feasible region because it has lower L_1 norm compared to the ground truth (see

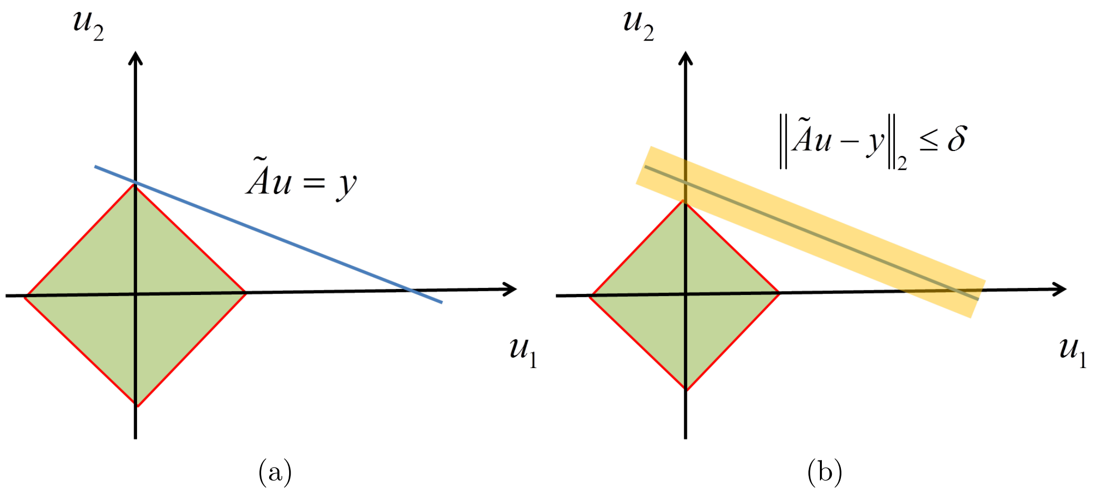


Figure 4.7: The minimum L_1 solution for (a) noiseless case and (b) noisy case. Note the solution will be on the inner boundary of feasible region.

Figure 4.8). Thus, for the noisy case our algorithm is likely to recover a solution with non-discrete levels. In such case, additional regularization constraint may be needed to prefer a solution with discrete levels.

4.4.1.1 Reweighed L_1 minimization

To recover a sparse solution, we modify the reweighed L_1 minimization algorithm proposed by Candès et al. [19]. We first introduce our algorithm for the binary case, with known intensity levels $\{0, 1\}$, then extend it into multi-level cases. Given the measured projection data $\mathbf{y}$, our algorithm can be described in Algorithm 3.

Our algorithm finds the minimum weighted L_1 norm (which can be viewed as a generalized L_1 ball) that satisfies the constraints. The solution will be the intersection of the minimum L_1 ball and the convex feasible sets for the constraints, where the shape of the L_1 ball is determined by the weights, and is updated iteratively. This is one type of majority minimization: After we get the solution from the initial condition, we will modify the corresponding weights. Then, in the next iteration,

Algorithm 3 Reweighed l_1 norm minimization for binary DT

```
1: Define the maximum iteration number $t_{max}$ and set the iteration number $t = 0$. Initialize a weight vector $\mathbf{w}(t) = \mathbf{1}$, where the dimension is equal to $\mathbf{u}$. Set a small positive constant $\kappa$.
2: Solve the reweighed $L_1$ minimization problem
       $\mathbf{u}(t) = \arg\min \|\mathbf{w}(t) \circ \mathbf{u}\|_1$
       subject to $\mathbf{y} = \tilde{\mathbf{A}}\mathbf{u}(t),\ \mathbf{x} = \mathbf{D}\mathbf{u}(t)$
       $\mathbf{0} \leq \mathbf{x} \leq \mathbf{1}$
   where the operator $\circ$ is defined as the element-by-element product of two vectors (Hadamard Product), $(\mathbf{w} \circ \mathbf{u})_i = w_i u_i$.
3: Update the weights according to the previous solution: for each entry, $i = 1, \ldots, n$, modify the weight by
       $w_i(t+1) = \frac{1}{|u_i(t)| + \kappa}$.    (4.17)
4: Reset the weights for non-integer solutions: Every 5 iterations, we reset $w_i(t)$ corresponding to non-integer $u_i(t)$ by a random number that is uniformly distributed between $[0, 2/\kappa]$
       $w_i(t) \leftarrow U(0, 2/\kappa)$    (4.18)
5: Terminate the iteration if the solution converges or reach the maximum number of iteration. Otherwise, increase $t \rightarrow t + 1$ and go to step 2.
```

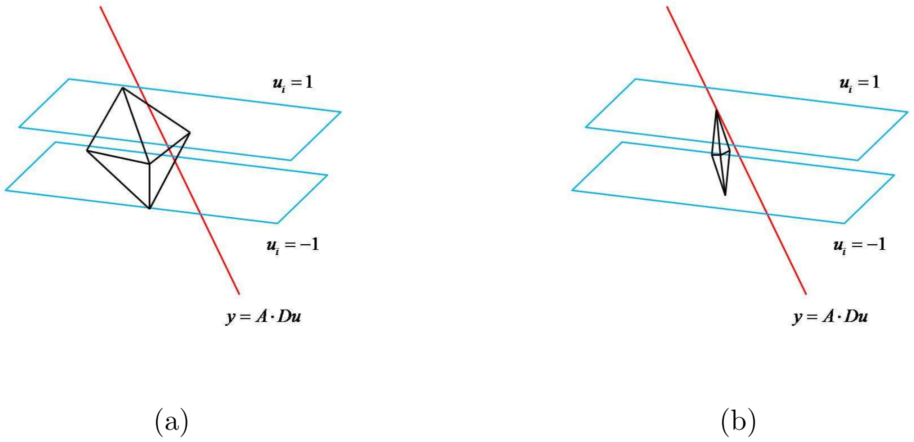


Figure 4.8: The reweighed L_1 minimization. Note that the feasible region is the intersection of data fitting $\mathbf{y} = \mathbf{A} \cdot \mathbf{Du}$ and $-\mathbf{1} \leq \mathbf{u} \leq \mathbf{1}$. (a) solution with initial weights (b) after reweighing, the solution stays on the boundary

the large entries of w_i will force those solution to concentrate on the dimensions where w_i is small [19]. The updated L_1 ball becomes a closer approximation to the L_0 norm, where the reweighing scheme is in charge of "choosing" the non-zero entries.

The key difference in our DT algorithm is that we know the possible values as a prior. With the designed dictionary, we preserve the "discrete level" property in transformed domain. In a typical sparse recovery, we need to identify the non-zero entries and estimate the values. However, in DT the value belongs to few possible levels, and we use it as an additional constraint to enhance the sparse recovery. We show the geometrical interpretation in Figure 4.8. The feasible region is the intersection of the data fitting $\mathbf{y} = \mathbf{A} \cdot \mathbf{Dx}$ and the intensity boundary $-\mathbf{1} \leq \mathbf{u} \leq \mathbf{1}$. In particular, the reweighed scheme will make the L_1 ball sharper, increasing the chances that the solution will be on the intensity boundary which corresponds to the possible discrete levels.

Moreover, we can encourage the solution to reach the known intensity levels by randomized modification of the weights. When the reconstruction is not perfect, it is possible that the solution will be trapped in the non-integer regions. In order to jump out from these non-integer regions, every 5 iterations we check the solution and identify the non-integer entries u_i and reset the corresponding weights w_i with a random value from the uniform distribution $U(0, 2/\kappa)$. This induces a "random search" for the non-integer entries, making it possible to converge to integer points. The reason we draw the weight from $U(0, 2/\kappa)$ is that we want to randomize the slope of the L_1 ball, and make sure this entry still has weight less than the all-zero entries (otherwise this entry will become zero from next iteration). Our randomized algorithm tends to find solutions with more integer values, while a non-randomized method will stop after a few iterations providing solutions with fewer integer values.

4.4.1.2 Projection onto convex sets

To solve the convex optimization in (4.14), the state of art second order methods (e.g., interior point method) use the Hessian matrix, which requires space of N^2 for storage. When we consider a typical 512×512 image, the Hessian matrix requires more than $6 \cdot 10^{10}$ variables which makes impractical for current computers. Thus, we switch to the first order method to solve the convex optimization with limited memory usage. In convex optimization, we need to search the solution in a convex feasible region, we use projection onto convex sets (POCS) [17] to project to solution from unconstrained optimization onto feasible region. The feasible region of the original convex problem can be separated as two constraints:

$$\begin{array}{rl} \text{Min} & \sum\limits_j \|\mathbf{u}^j\|_1 \\ \text{Subject to} & C_{1:\text{Data fitting}} : \mathbf{Ax} = \mathbf{y} \\ & C_{2:\text{Boundary}} : \mathfrak{L}_\mathsf{i} \leq x_i \leq \mathfrak{U}_\mathsf{i} \end{array}$$

where $\mathbf{x} = \sum_j k_j \cdot \mathbf{D}\mathbf{u}^j$. The convex set and projection operator for C_1 and C_2 are well-defined (C_1 is a hyper-plane of null space $\mathbf{A}$, and C_2 is a hyper-cube). But, directly projecting the solution onto the feasible region $C = C_1 \cap C_2$ is not an easy task, thus, we move one constraint into cost function and alternately solve these two smaller problems. The first one is:

$$\text{Min} \quad \sum_j \|\mathbf{u}^j\|_1 + \mu \cdot \phi_{Bd}(\mathbf{x}) \tag{4.19}$$

$$\text{Subject to} \quad \mathbf{u}^j \in C_{1:\text{Data fitting}} \tag{4.20}$$

And the second one is:

$$\text{Min} \quad \sum_j \|\mathbf{u}^j\|_1 + \nu \cdot \|\mathbf{Ax} - \mathbf{b}\|_2 \tag{4.21}$$

$$\text{Subject to} \quad \mathbf{u}^j \in C_{2:\text{Boundary}} \tag{4.22}$$

where we use $\|\mathbf{Ax} - \mathbf{b}\|_2$ as the cost function for data fitting, and $\phi_{Bd}(\mathbf{x})$ as the cost function for the boundary constraints. Each entry of $\phi_{Bd}(\mathbf{x})$ is defined as:

$$\phi_{Bd}(\mathbf{x})_i = \begin{cases} 0, & \mathfrak{L}_\mathrm{i} \leq x_i \leq \mathfrak{U}_\mathrm{i} \\ |\mathfrak{L}_\mathrm{i} - x_i|, & x_i \leq \mathfrak{L}_\mathrm{i} \\ |\mathfrak{U}_\mathrm{i} - x_i|, & x_i \geq \mathfrak{U}_\mathrm{i} \end{cases} \tag{4.23}$$

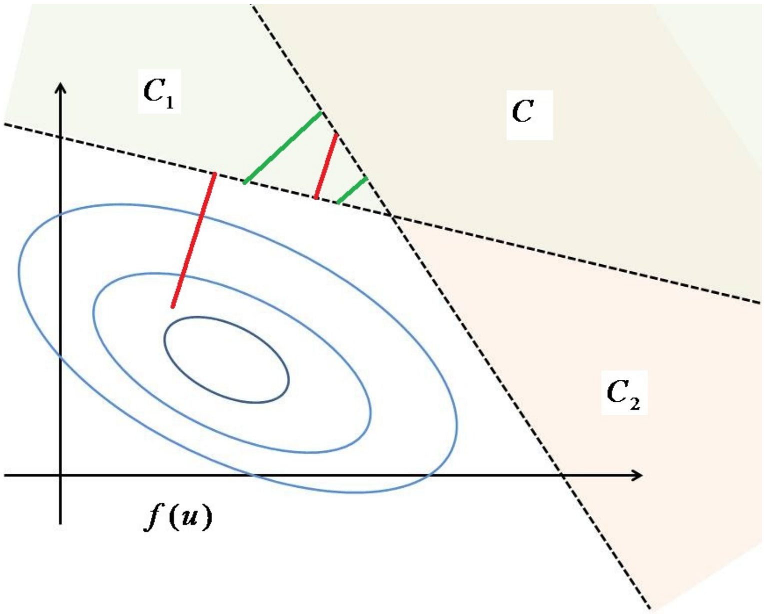


Figure 4.9: Alternating projection on two convex sets

We use POCS to alternate solving these two small problems. In the first step, we solve the first problem by running an unconstrained minimization for cost function in (4.19), then we project the solution onto the convex set in (4.20) to satisfy the data-fitting constraint. The unconstrained minimization is done by first order method (gradient based), which only requires space of size N to store the gradient information. In the second step, we use the result from the previous step as an initial point, then solve the second problem by the same unconstrained minimization - projection onto convex set procedure. We do this iteratively until the solution converges.

Because the original problem is convex, the convergence of the solution is guaranteed. In POCS, we separate the original convex problem into two smaller problems, and use basic operations to solve it iteratively. Each operation (unconstrained optimization, projection) can be done with reduced space requirement, however, the drawback is that compared to second order convex optimization, although POCS converges to global optima, it does require much more time to converge.

4.4.2 Estimating the unknown intensity level

In Section 4.4.1, we proposed how to solve a DT problem with known intensity level. Now, we want to move forward to the unknown intensity level case. Based on our previous algorithm, we add the intensity level estimation step, and alternate these two steps iteratively. In this step, we will estimate the unknown intensity levels $\{0, k_1, \ldots, k_M\}$ based on the solution from the sparse recovery step. We use the histogram of pixel values and run clustering algorithm on it. We only choose partial data, which have intensity values greater than a small constant ξ, and run the Gaussian Mixture Model (GMM) based clustering to cluster the histogram of data into M groups. We only use pixel values greater than ϵ for clustering because 0 is a prior known level, whose intensity does not need to be estimated. Furthermore, the Gaussian distribution used in GMM is symmetrical and pixel intensities are non-negative, thus if we performed GMM clustering on all pixel values we would never obtain a cluster with mean 0. In summary, we use simple thresholding to assign small pixel values to the cluster with intensity 0, while using GMM to estimate the remaining intensities from pixel data above the threshold.

After GMM clustering, the result will be G_j, with their mean c_j, variance σ_j and cardinality $|G_j|$. Then we update the possible intensity level k_j based on the mean c_j and variance σ_j of each cluster:

$$k_j \leftarrow k_j - \gamma(\frac{\sigma_j}{|G_j|}) \cdot (k_j - c_j). \tag{4.24}$$

The new intensity estimate k_j is chosen so that it moves closer to the mean c_j of cluster j: the coefficient function $\gamma(x)$ is defined as $\min\{\frac{1}{4}, x\}$ to control the update speed. We use the ratio of the variance and cardinality $\frac{\sigma_j}{|G_j|}$ to define the step size, which implies large average variance will speed up the convergence. After updating the intensity estimate k_j, we use it as the input for the sparse reconstruction step and run it iteratively.

4.5 Simulation Results

In this section, we show the simulation results for our algorithm. The testing image is a simplified Shepp-Logan phantom, and the measurements are taken with equally spaced projections. The projection angles are chosen to be uniform in the range $[0, 180]$ degree. In order to test the noisy measurement, we add white Gaussian noise with zero mean and different variance to the measured data, and the search allowance δ is chosen to be equal to the noise variance.

In **Simulation 1**, we assume the intensity levels are given, and the testing image is a binary level Shepp-Logan phantom with size 32×32. The measurements are taken with 5 different angles uniformly distributed in $[0, 180]$, and the added

Gaussian noise is zero mean with variance $\{0, 0.1, 0.25, 0.5, 1\}$. Because the intensity level is known, we just show the results from sparse recovery step in Figure 4.10. For comparison, we list the mean square error achieved by the method from Weber [74] and by a filtered back-projection reconstruction. The results show that in the noiseless case, our algorithm achieves a perfect reconstruction with only 5 different viewing angles. For high signal-to-noise ratio (SNR) cases, our algorithm shows superior performance to the other two methods. However, in low SNR, because the measurements are corrupted by noise and we choose the search allowance to be equal to the noise variance, the solution will always stays on the inner bound of feasible region which means the solution is closer to the origin than the high SNR cases. One extreme example is if the noise variance is very high and the feasible region contains the origin, the minimum weighed L_1 solution will be $\mathbf{0}$ because it gives the lowest possible L_1 norm and also inside the feasible region. Thus, for very low SNR cases, we may need other regularization technique to combine with reweighed L_1 norm to solve DT.

In **Simulation 2**, we use a 64×64 simplified Shepp-Logan phantom with three intensity levels as the testing image. The intensity levels are $\{0, 16, 80\}$, which correspond to the background, left lobe and outer circle, respectively. The maximum possible intensity value is set to 100. For the measurements, $\{12, 18\}$ different projection angles are taken uniformly between $[0, 180]$ and zero mean Gaussian noise is added with variance $\sigma = \{0, 0.1, 0.2, 0.5, 1, 2, 5\}$.

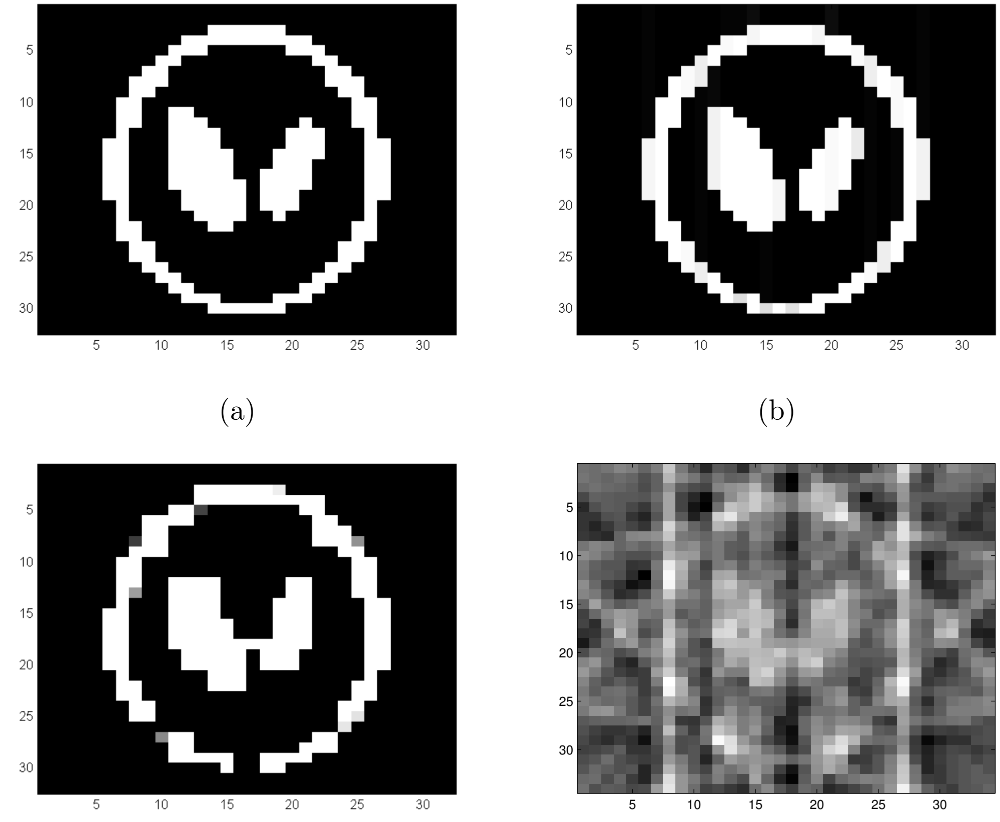
We show the histogram of the reconstructed image through iterations in Figure 4.12. After we update the intensity level, the distribution of histogram becomes more concentrated and finally converges to a few peaks. The result shows that most of the non-zero entries x_i stay on the boundary of feasible region, as we expected. The mean square error of the reconstructed image with different noise variance and number of projections is shown in Figure 4.13. We compare the results with the standard filtered back-projection [42], and total variation (TV) reconstruction [65]. Results show that our method has much better reconstructed image quality than the other two approaches, and even outperforms the TV method in the case with same noise variance, fewer measurements. Even though unknown the intensity level case is a more difficult problem, the overall results still show good performance.

In **Simulation 3**, we use a testing image with size 256×256. The number of variables is over $50,000$ and the second order method can not be used to solve the convex optimization due to the requirement for the large memory space to store the Hessian matrix. We switch to the first order algorithm, projection onto convex sets (POCS) [17] method to solve it with limited memory requirement.

When we alternate solving these two small problems in (4.19), after we project the solution onto the data fitting feasible set, some pixels may become negative (see Figure 4.14). In the next step, after we finish the minimization and project the solution onto the convex set for boundary constraint, all pixels go back to greater than zero in order to satisfy the boundary constraint but with slightly higher data fitting error. The change of MSE with iterations is listed in Figure 4.15, and the result shows it iteratively converges to the ground truth.

4.6 Conclusions

In this chapter, we have presented a new sparse reconstruction formulation for discrete tomography, which focuses on reconstruction of few level intensity images that have a sparse representation using a well designed dictionary. Our binary reconstruction algorithm uses LP to relax the integer solution condition and also search the the sparse representation. We introduce a randomized reweighed L_1



(a)

(b)

(c)

(d)

Figure 4.10: The Simulation 1 results. (a) The binary testing phantom (b) Reconstruction result with noise variance = 0.1 (c) S. Weber's method with same noise variance (d) Filtered back projection result with same noise

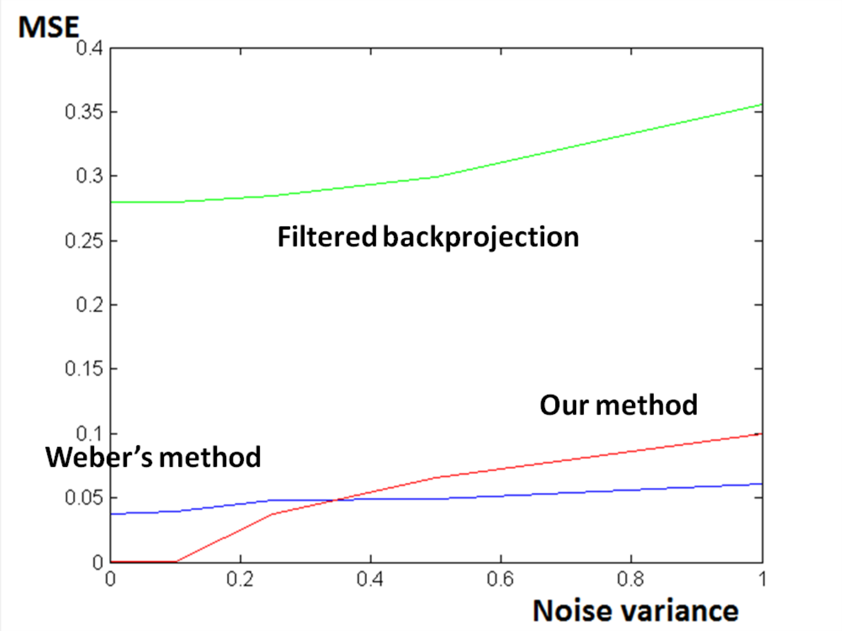
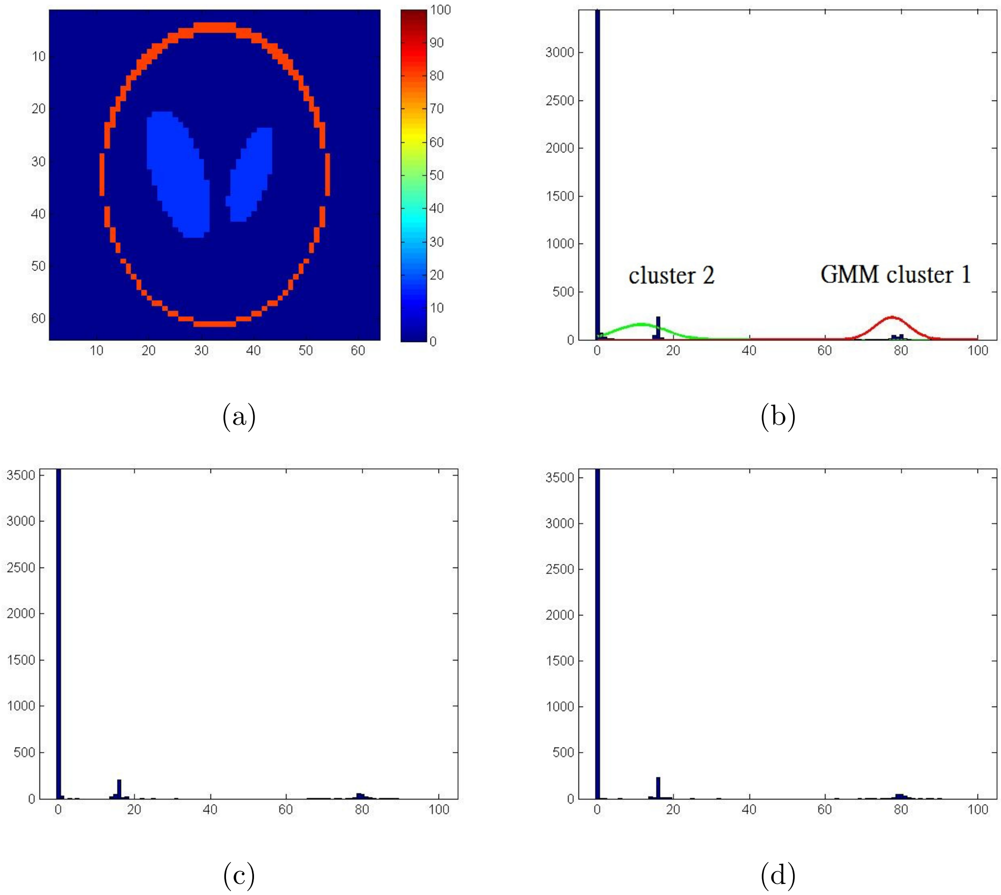


Figure 4.11: Comparison of the mean square error with respect to different noise level in Simulation 1. Note our method have perfect reconstruction in noiseless case and outperform other methods in high SNR case.

minimization to enhance convergence to an integer solution. We also present a new automatic intensity level estimation algorithm adding the binary reconstruction algorithm. Our algorithm iterates between reweighed L_1 minimization and GMM histogram clustering steps to estimate the unknown intensity levels, and it has very good performance in our preliminary experimental evaluations for multi-level DT problems. It outperforms the conventional TV method even with fewer measurements. For large image reconstruction, we use POCS to handle the problem. but this algorithm suffers from the slow convergence speed.

Future work involves understanding the performance of the proposed algorithm under different noise and image models, for example using Poisson models for the noise statistics of the X-ray detector. We also plan to use the discrete value property to enhance signal recovery from noisy measurements. The speed of convergence for the first-order method in our DT problem is also very interesting. Finally, we want



(a)

(b)

(c)

(d)

Figure 4.12: Simulation 2. (a) Test Shepp-Logan phantom (b) Histogram of the reconstructed image after 1 iteration (c) Histogram after 2 iterations (d) Histogram after 2 iterations. After few intensity updates, the histogram is more concentrated on few spots.

to develop a method on top of our current work to handle the unknown number of intensity levels case.

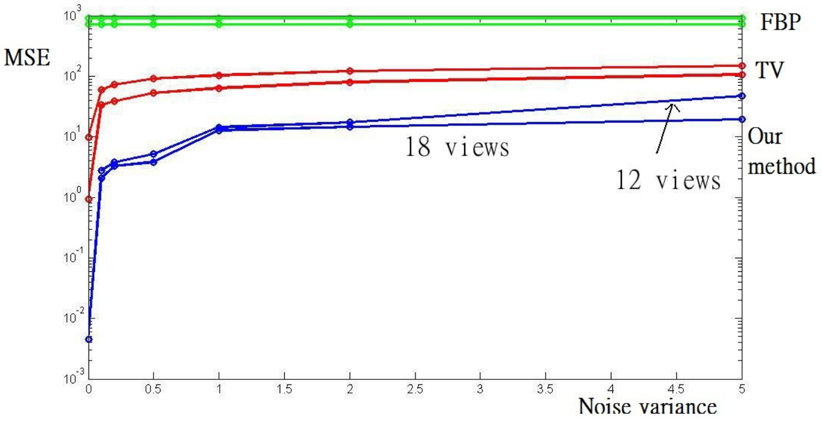
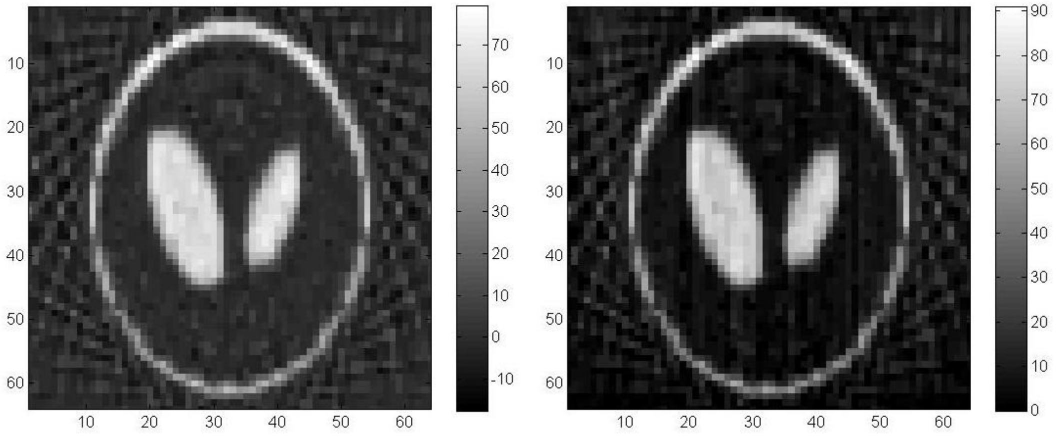


Figure 4.13: Reconstruction mean square error with respect to the noise level with $\{12, 18\}$ number of projection views in Simulation 2. This shows that our method outperform Total-Variation reconstruction in all cases.



(a)

(b)

Figure 4.14: Projection onto convex set results in Simulation 3 (a) After projection onto the data fitting set. Note that some pixels may be negative. (b) After projection onto boundary constraint set. All pixels are within the range now.

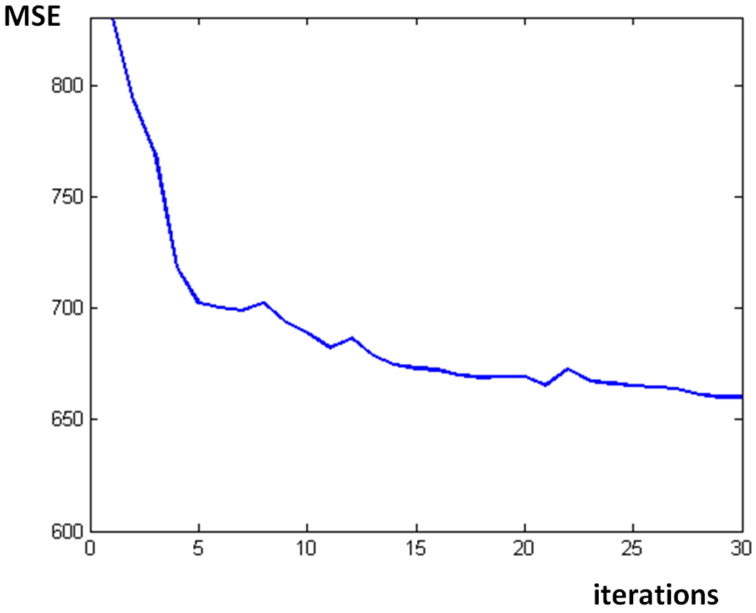


Figure 4.15: The reconstructed mean square error with respect to the number of iterations in Simulation 3. The result shows the MSE decreases with respect to number of iterations, and also reflects the slow convergence of POCS.

Chapter 5

Conclusions

A high contrast transmission tomography problem has been proposed and studied for sparse data scenario. In Chapter 2, an object based model is proposed to represent the high velocity structure in travel time tomography. This model provides a sparse representation and greatly reduces the number of parameters needed for the model. A fast path tracking algorithm is proposed in Section 2.3 that provides a efficient way to find the travel path in the high velocity object model. To do tomographic reconstruction, we suggest to use the probabilistic approach to estimate the probability distribution of model space. Moreover, we propose an accelerated random walk algorithm to explore multiple minima and generate the probability map. The result can be interpreted as a "appearance" probability map of high velocity structures in different areas. The probability map describe the most possible models and cover the non-uniqueness nature of the solution, where the user can add prior knowledge or personal expertise to choose between possible models.

In Chapter 3, fracture detection problem is proposed and the tomographic reconstruction algorithm is used for a low permeability reservoir under water-flooding. Based on the current infrastructure, we use the "injected water" as a probe signal and injection-production wells as transmitters-receivers. Compared to existing reservoir characterization methods, our approach has benefit that it won't interrupt the usual production and be able to detect the dynamical change of reservoir on-line. The conditions of designing injection schedule are provided in Section 3.3, and we show how to estimate reservoir response and travel time from production data. The result of field experiment is shown in Section 3.5.

For X-ray transmission tomography, piecewise-constant dictionary is proposed to represent the type of image for our interest, e.g., high contrast images which can be segmented to several smooth boundary regions with same intensity. This provides a very sparse representation since only boundaries and intensity values of each region needs to be coded. We formulate this problem as a discrete sparse recovery and propose an algorithm to alternating recovering the boundary and estimating the intensity level of each region in Section 4.4. Moreover, the reconstructed image quality outperform conventional reconstruction algorithms (total-variation minimization and filter-back projection) when the measured data is sparse, and it leads to better interpretation of image in applications that number of X-ray projections is limited.

5.1 Future Work

There are several interesting directions for the future work. One is to consider the very low velocity structure in the travel time tomography. The heterogeneous structure could be either high or low velocity, and the path finding algorithm should be extended to cover both cases where the object may speed up or slow down the travel time. Currently, the shape and number of objects in our algorithm predefined. In particular, if the object shape is closed to the natural shape of the high contrast structure, our method can have superior reconstructed image quality comparing to grid-based methods. However, in many situations the geometrical shape of structure is unknown, and there is no guarantee that our pre-defined number of objects will provide a good approximation for it. Thus, adaptively modifying the shape and number of the object could be a new way to achieve better reconstructed image quality.

In X-ray tomography, the dictionary with unit-step functions as atoms is capable of representing the smooth boundary, few intensity level image efficiently. However, this dictionary also has high coherence, which increase the uncertainty for sparse recovery. This scheme can be improved by considering different type of dictionary, e.g., graph based transform should be very efficient to smooth edge images. Moreover, the transform can be adaptive, which based on the sparse recovery result. The combination of dictionary design and sparse recovery could lead to a more efficient representation and better reconstruction quality.

Reference List

[1] Tayeb A. Tafti, Iraj Ershaghi, Amin Rezapour, and Antonio Ortega. Injection scheduling design for reduced order waterflood modeling. In *SPE Western Regional & AAPG Pacific Section Meeting, 2013 Joint Technical Conference*, 2013.

[2] M. Abbaszadeh-Dehghani and W.E. Brigham. Analysis of well-to-well tracer flow to determine reservoir heterogeneity. In *Annual California Regional Meeting*, San Francisco, California, USA, March 1982.

[3] M. Abbaszadeh-Dehghani and W.E. Brigham. Analysis of well-to-well tracer flow to determine reservoir layering. *Journal of Petroleum Technology*, pages 2257–2270, 1984.

[4] P.K. Agarwal and S. Suri. Surface approximation and geometric partitions. *SIAM Journal on Computing*, 27(4):1016–1035, 1998.

[5] Ahmed Alhuthali, Adedayo Oyerinde, and Akhil Datta-Gupta. Optimal waterflood management using rate control. *SPE Reservoir Evaluation & Engineering*, 10(5):539–551, 2007.

[6] C.Y. Bai and S. Greenhalgh. 3-d non-linear travel-time tomography: Imaging high contrast velocity anomalies. *Pure and Applied Geophysics*, 162(11):2029–2049, 2005.

[7] KJ Batenburg and J. Sijbers. DART: A fast heuristic algebraic reconstruction algorithm for discrete tomography. In *Proceedings of the IEEE International Conference on Image Processing, San Antonio, Texas*, 2007.

[8] KJ Batenburg, W. van Aarle, and J. Sijbers. A Semi-Automatic Algorithm for Grey Level Estimation in Tomography. *Pattern Recognition Letters*, 2010.

[9] James G. Berryman. Stable iterative reconstruction algorithm for nonlinear traveltime tomography. In *Inverse Problems*, pages 21–42, 1990.

[10] James G. Berryman. Lecture notes on nonlinear inversion and tomography. Technical report, October 1991.

[11] J.G. Berryman. Fermat's principle and nonlinear traveltime tomography. *Physical review letters*, 62(25):2953–2956, 1989.

[12] J.G. Berryman. Lecture notes on nonlinear inversion and tomography: 1, borehole seismic tomography. Technical report, Lawrence Livermore National Lab., CA (United States), 1991.

[13] P. Boufounos, M. Duarte, and R. Baraniuk. Sparse signal reconstruction from noisy compressive measurements using cross validation. In *IEEE Workshop on Statistical Signal Processing*, pages 299–303. Citeseer, 2007.

[14] ND Bregman, RC Bailey, and CH Chapman. Crosshole seismic tomography. *Geophysics*, 54(2):200–215, 1989.

[15] DR Brouwer and JD Jansen. Dynamic optimization of water flooding with smart wells using optimal control theory. In *European Petroleum Conference*, 2002.

[16] F. Cademartiri, N.R. Mollet, A. van der Lugt, E.P. McFadden, T. Stijnen, P.J. de Feyter, and G.P. Krestin. Intravenous Contrast Material Administration at Helical 16–Detector Row CT Coronary Angiography: Effect of Iodine Concentration on Vascular Attenuation1. *Radiology*, 236(2):661, 2005.

[17] P.H. Calamai and J.J. Moré. Projected gradient methods for linearly constrained problems. *Mathematical Programming*, 39(1):93–116, 1987.

[18] E.J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509, 2006.

[19] E.J. Candes, M.B. Wakin, and S.P. Boyd. Enhancing sparsity by reweighted l 1 minimization. *Journal of Fourier Analysis and Applications*, 14(5):877–905, 2008.

[20] R. Courant and D. Hilbert. *Methods of mathematical physics*, volume 1. CUP Archive, 1962.

[21] Stanley R Deans. *The Radon transform and some of its applications*. Courier Dover Publications, 2007.

[22] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[23] Kris A Dines and R Jeffrey Lytle. Computerized geophysical tomography. *Proceedings of the IEEE*, 67(7):1065–1073, 1979.

[24] Oliver D.S. The sensitivity of tracer concentration to nonuniform permeability and porosity. *Transport in Porous Media*, 30, 1998.

[25] S. Duane, A.D. Kennedy, B.J. Pendleton, and D. Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

[26] J.M. Duarte-Carvajalino and G. Sapiro. Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization. *Image Processing, IEEE Transactions on*, 18(7):1395–1408, 2009.

[27] B. Efron and G. Gong. A leisurely look at the bootstrap, the jackknife, and cross-validation. *The American Statistician*, 37(1):36–48, 1983.

[28] H.W. Engl, M. Hanke, and A. Neubauer. *Regularization of inverse problems*. Springer Netherlands, 1996.

[29] M.A.T. Figueiredo, R.D. Nowak, and S.J. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *Selected Topics in Signal Processing, IEEE Journal of*, 1(4):586–597, 2008.

[30] P. Fishburn, P. Schwander, L. Shepp, and RJ Vanderbei. The discrete Radon transform and its approximate inversion via linear programming. *Discrete Applied Mathematics*, 75(1):39–61, 1997.

[31] Joachim Frank. *Electron tomography: methods for three-dimensional visualization of structures in the cell*. Springer Science+ Business Media, 2006.

[32] Sam G Gibbs, Kenneth B Nolen, et al. Inferred production rates of a rod pumped well from surface and pump card information, May 1 2007. US Patent 7,212,923.

[33] S.W. Golomb, L.R. Welch, R.M. Goldstein, and A.W. Hales. *Shift register sequences*, volume 51. Holden-Day San Francisco, 1967.

[34] Jörg Hausleiter, Tanja Meyer, Franziska Hermann, Martin Hadamitzky, Markus Krebs, Thomas C Gerber, Cynthia McCollough, Stefan Martinoff, Adnan Kastrati, Albert Schömig, et al. Estimated radiation dose associated with cardiac ct angiography. *JAMA: the journal of the American Medical Association*, 301(5):500–507, 2009.

[35] Gabor T Herman. *Fundamentals of computerized tomography: image reconstruction from projections*. Springer, 2009.

[36] G.T. Herman and A. Kuba. *Discrete Tomography: Foundations, Algorithms, and Applications*. Birkhauser, 1999.

[37] JA Hole. Nonlinear high-resolution three-dimensional seismic travel time tomography. *Journal of Geophysical Research: Solid Earth (1978–2012)*, 97(B5):6553–6562, 1992.

[38] Godfrey N Hounsfield. Computerized transverse axial scanning (tomography): Part 1. description of system. *British Journal of Radiology*, 46(552):1016–1022, 1973.

[39] L. Husmann, I. Valenta, O. Gaemperli, O. Adda, V. Treyer, C.A. Wyss, P. Veit-Haibach, F. Tatsugami, G.K. Von Schulthess, and P.A. Kaufmann. Feasibility of low-dose coronary ct angiography: first experience with prospective ecg-gating. *European heart journal*, 29(2):191–197, 2008.

[40] D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge. Comparing images using the Hausdorff distance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(9):850–863, 1993.

[41] C.R. Johnson, RA Greenkorn, and EG Woods. Pulse-testing: a new method for describing reservoir flow properties between wells. *Journal of Petroleum Technology*, 18(12):1599–1604, 1966.

[42] A.C. Kak and M. Slaney. Principles of computerized tomographic imaging. 1988.

[43] Avinash C.. Kak and Malcolm Slaney. *Principles of computerized tomographic imaging*. Society for Industrial and Applied Mathematics, 2001.

[44] J. Kiefer. Sequential minimax search for a maximum. In *Proc. Amer. Math. Soc*, volume 4, pages 502–506, 1953.

[45] S. Kirkpatrick. Optimization by simulated annealing: Quantitative studies. *Journal of statistical physics*, 34(5):975–986, 1984.

[46] A. Kuba and E. Balogh. Reconstruction of convex 2D discrete sets in polynomial time. *Theoretical Computer Science*, 283(1):223–242, 2002.

[47] J.W. Lane Jr, F.D. Day-Lewis, R.J. Versteeg, and C.C. Casey. Object-based inversion of crosswell radar tomography data to monitor vegetable oil injection experiments. *Journal of Environmental and Engineering Geophysics*, 9:63, 2004.

[48] K.H. Lee, A. Ortega, A. Nejad, N. Jafroodi, and I. Ershaghi. A novel method for mapping fractures and high-permeability channels in waterfloods using injection and production rates. In *SPE Western Regional Meeting*, 2009.

[49] L. Ljung. *System identification: Theory for the User*. Upper Saddle River, NJ: Prentice Hall, 2nd edition, 1999.

[50] Computer Modeling Group Ltd. CMG numerical simulators. "http://www.cmgroup.com/software/completesuite.htm", October 2010.

[51] T. Lukić. Discrete tomography reconstruction based on the multi-well potential. *Combinatorial Image Analysis*, pages 335–345, 2011.

[52] D.J.C. MacKay. *Information theory, inference, and learning algorithms*. Cambridge Univ Pr, 2003.

[53] RM McKinley, S. Vela, and LA Carlton. A field application of pulse-testing for detailed reservoir description. *Journal of Petroleum Technology*, 20(3):313–321, 1968.

[54] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21:1087, 1953.

[55] W. Munk and C. Wunsch. Ocean acoustic tomography: A scheme for large scale monitoring. *Deep Sea Research Part A. Oceanographic Research Papers*, 26(2):123–161, 1979.

[56] Y. Nesterov and A. Nemirovsky. Interior point polynomial methods in convex programming, 1994.

[57] A.Y. Ng. Preventing overfitting of cross-validation data. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 245–253. Morgan Kaufmann Publishers Inc., 1997.

[58] R.G. Pratt. Seismic waveform inversion in the frequency domain. i. theory and verification in a physical scale model. *Geophysics*, 64:888–901, 1999.

[59] M. Rantala, S. Vanska, S. Jarvenpaa, M. Kalke, M. Lassas, J. Moberg, and S. Siltanen. Wavelet-based reconstruction for limited-angle x-ray tomography. *Medical Imaging, IEEE Transactions on*, 25(2):210–217, 2006.

[60] G. Rote. Computing the minimum Hausdorff distance between two point sets on a line under translation. *Information Processing Letters*, 38(3):123–127, 1991.

[61] Geoffrey D Rubin, Maria C Shiau, Andrew J Schmidt, Dominik Fleischmann, Laura Logan, Ann N Leung, R Brooke Jeffrey, and Sandy Napel. Computed tomographic angiography: historical perspective and new state-of-the-art using multi detector-row helical computed tomography. *Journal of computer assisted tomography*, 23:S83, 1999.

[62] Morteza Sayarpour. *Development and Application of Capacitance-Resistive Models to Water/CO2 Floods*. PhD thesis, University of Texas, 2008.

[63] J.A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences of the United States of America*, 93(4):1591, 1996.

[64] R.E. Sheriff and L.P. Geldart. Exploration seismology. volume 1: History, theory, and data acquisition. 1983.

[65] Emil Y Sidky and Xiaochuan Pan. Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization. *Physics in medicine and biology*, 53(17):4777, 2008.

[66] V. Stankovic, L. Stankovic, and S. Cheng. Compressive sampling of binary images. In *CISP08 Congress on Image and Signal Processing.*

[67] Bagus Sudaryanto and Yannis C Yortsos. Optimization of fluid front dynamics in porous media using rate control. i. equal mobility fluids. *Physics of Fluids*, 12:1656, 2000.

[68] A. Tarantola. Inverse problem theory: Methods for data fitting and model parameter estimation. 1987.

[69] C Thurber, S Roecker, W Ellsworth, Y Chen, W Lutter, and R Sessions. Two-dimensional seismic image of the san andreas fault in the northern gabilan range, central california: Evidence for fluids in the fault zone. *Geophysical Research Letters*, 24(13):1591–1594, 1997.

[70] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[71] Y. Tsang, M. Coates, and R.D. Nowak. Network delay tomography. *Signal Processing, IEEE Transactions on*, 51(8):2125–2136, 2003.

[72] Alessandro Vallebona and Pietro Amisano. *Trattato di stratigrafia*. Vallardi, 1952.

[73] DW Vasco and A. Datta-Gupta. Integrating field production history in stochastic reservoir characterization. *SPE Formation Evaluation*, 12(3):149–156, 1997.

[74] S. Weber, C. Schnorr, T. Schule, and J. Hornegger. Binary tomography by iterating linear programs. *COMPUTATIONAL IMAGING AND VISION*, 31:183, 2006.

[75] H.J. Welge. A simplified method for computing oil recovery by gas or water drive. *Petroleum Transactions AIME*, 195:91–98, 1952.

[76] S. Whitaker. Flow in porous media i: A theoretical derivation of darcy's law. *Transport in porous media*, 1(1):3–25, 1986.

[77] DK Wilson, A. Ziemann, VE Ostashev, and AG Voronovich. An overview of acoustic travel-time tomography in the atmosphere and its potential applications. *Acta Acustica united with Acustica*, 87(6):721–730, 2001.

[78] D.P. Wipf and B.D. Rao. Sparse Bayesian learning for basis selection. *IEEE Transactions on Signal Processing*, 52(8):2153–2164, 2004.

[79] A.A. Yousef. *Investigating Statistical Techniques to Infer Interwell Connectivity from Production and Injection Rate Fluctuations*. PhD thesis, The University of Texas at Austin, Austin, Texas, USA, 2005.

[80] A.A. Yousef, P.H. Gentil, J.L. Jensen, and L.W. Lake. A capacitance model to infer interwell connectivity from production and injection rate fluctuations. *SPE Reservoir Evaluation and Engineering Journal*, 9:630–646, 2006.

[81] A.A. Yousef and L.W. Lake. Analysis and interpretation of interwell connectivity from production and injection rate fluctuations using a capacitance model. In *SPE Symposium on Improved Oil Recovery*, Tulsa, Oklahoma, USA, April 2006.

[82] BG Ziedses des Plantes. Examen du troisième et du quatrième ventricule au moyen de petites quantités d'air. *Acta radiol., Stockholm*, 34:399–407, 1950.