Object-based high contrast travel time tomography

Yenting Lin* and Antonio Ortega*

*Department of Electrical Engineering,
University of Southern California

Los Angeles, CA, 90089

(February 6, 2014)

GEO-Example

Running head:

ABSTRACT

We consider travel time tomography problems involving detection of high contrast, discrete high velocity structures. This results in a discrete nonlinear inverse problem, for which traditional grid-based models and iterative linearized least-squares reconstruction algorithms are not suitable. This is because travel paths change significantly near the high contrast velocity structure, making it more difficult to inversely calculate the travel path and infer the velocity along the path. We propose a model-based approach to describe the high velocity structure using pre-defined elementary objects. Compared to a grid-based model, our approach has complexity that increases as a function of the number of objects, rather than increasing with the number of cells (usually very large). A new reconstruction algorithm is developed that provides estimates of the probability that a high velocity structure appears at any point in the region of interest. Simulation results show that our method can efficiently sample the model parameter space, and we map the model parameters into the high velocity structures in spatial domain to generate a “probability map”, which represent

arXiv:1303.3052v1 [physics.data-an] 12 Mar 2013
the appearance of the high velocity structure in different regions. We show the probability
map not only gives the highest probability to the optimal solution, but also includes other
possible models as well.
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 (Pratt 1999) and acoustic tomography (for example, in the atmosphere’s surface (Wilson et al. 2001) or in the ocean layers (Munk and Wunsch 1979)). However, different from many other transmission tomographic reconstruction problems (X-Ray CT, Positron emission tomography CT) where the straight-line trajectory assumption is commonly used in the problems we consider here the travel path may bend severely if the travel velocity variation is very high. 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 trajectory assumption, travel paths can be characterized by Fermat’s principle: the travel path observed is the path traversed in the shortest time (Berryman 1989), 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 permeable fracture zones in ground-water flow characterization (National Research Council (US). Committee on Fracture Characterization and Fluid Flow 1996), monitoring the water/oil saturation in vegetable oil bio-remediation
different from the case where the velocity distribution only has small variations [Wilson et al., 2001], the travel path in high velocity contrast media not only bends severely but is almost dominated by these high velocity structures 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 [Berryman, 1991a].

Our initial motivation for this work comes from the flow permeability characterization in a fractured reservoir [Vasco and Datta-Gupta, 1997]. 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 [Welge, 1952]. 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 bore-hole loc-
cations, 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. Bore-hole 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 (Tsang et al., 2003), where the main goal is to identify the nodes where con-
gestion 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 ve-
locity distribution. Grid-based methods (Berryman, 1991b), 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 (Lane Jr et al., 2004) 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 distribution of velocity 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 (Bai and Greenhalgh, 2005) proposed to add irregular nodes on the boundary of cells to improve the stability when determining nonlinear ray travel paths. Berryman (Berryman, 1990) 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 which may not even be covered by any travel path. For these areas, 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 paper, 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 (Engl et al. 1996). 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 (Ng 1997).

An alternative approach is to estimate the probability distribution for the model space according to the data-fitting (Tarantola 1987). This gives a full description of the relative probabilities of the different models, so that we can explicitly consider all likely solutions.
However, generating samples and estimating the probability density is very challenging and time consuming for a high dimensional model space (MacKay, 2003).

In this paper, 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 the next section, and introduce the forward step and define the mathematical formulation for the travel path finding problem. Then we give an overview of our proposed algorithm for probability density estimation. Simulation results and conclusions are presented in the end of this paper.
OBJECT BASED MODEL

In this paper, 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 Fig. 1, where the forward step predict the travel time for a given velocity model. In the inverse step, a velocity model is estimated based on the measured travel time data. In this section, we will introduce the object based approach to represent the high contrast velocity model.

![Figure 1: The forward-inverse step of the velocity model and travel time](image.png)

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 [Agarwal and Suri 1998], for example, triangle or ellipse. In this chapter, we use convex objects as our fundamental objects, which can be selected based on user’s choice. For example, if we only use one kind of convex object (ellipse) as the fun-
damental object, the \( i \)th 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 \( \theta_i = \{s_i, c_i, \psi_i\} \), where \( s_i, c_i, \psi_i \) denote the scale, center location and the orientation respectively (see Fig. 2(b)). We denote \( |\theta_i| \) the area inside the \( i \)th object. This formulation can be easily extended to multiple convex shapes, for example, the object can be either a triangle or an ellipse with one more parameter to identify the type of shape.

![Figure 2](image.png)

**(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 and the background. Let \( \{\theta_1, \theta_2, \ldots, \theta_N\} \) 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 \( \theta_i \). Therefore, the velocity distribution can be represented by the objects as:

\[
V(x, y) = \begin{cases} 
v_i, & \text{if } (x, y) \in |\theta_i|, 
v_h, & \text{otherwise}. \end{cases}
\] (1)
Obviously the spatial velocity distribution is determined by the objects. We use the notation \( V(\theta_1, \ldots, \theta_N) \) to indicate the velocity distribution when we have \( N \) objects with parameters \( \theta_1, \ldots, \theta_N \) in the model.

We use \( d(\theta_i, \theta_j) \) to represent the distance between two objects, which is defined as:

\[
d(\theta_i, \theta_j) = \min_{\mu, \nu} \| \mu - \nu \|_2, \quad \mu \in |\theta_i|, \quad \nu \in |\theta_j|
\]

and the corresponding path is denoted by \( P(\theta_i, \theta_j) \). The same notation can be used for the distance between point and object, or point to point, i.e., \( d(\alpha, \beta) \) and \( P(\alpha, \beta) \) are the distance and path between points \( \alpha \) and \( \beta \), etc.

**FORWARD STEP**

In this section, we introduce the forward step: Given the velocity model as input, the forward model calculates the corresponding travel time between arbitrary transmitters and receivers. We use the mathematical formulation proposed in [Berryman, 1991a](#) 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 \( \{\alpha, \beta\} \) based on velocity model \( V \) is defined by the path integral:

\[
\tau_P(V, \alpha, \beta) = \int_P \frac{1}{V(x, y)} \, dl^P, \quad \text{with} \quad P_{\text{start}} = \alpha, \quad P_{\text{end}} = \beta.
\]

We define the travel path, \( P^* \), as the path with minimum time cost \( \tau^* \). Therefore, we can define the travel time \( \tau^* \) between two points \( \{\alpha, \beta\} \) as:

\[
\tau^*(V, \alpha, \beta) = \min_P \tau_P(V, \alpha, \beta) = \min_P \int_P \frac{1}{V(x, y)} \, dl^P.
\]
the path $P^*$ will be:

$$P^*(V, \alpha, \beta) = \arg \min_{P} \tau^{P}(V, \alpha, \beta). \quad (5)$$

Finding the analytical solution for the travel time $\tau^*$ and travel path $P^*$ is a classical problem in calculus of variations (Courant and Hilbert 1962). Conventional methods, including the shotgun ray-tracing method (Berryman 1991a) or the level set method (Sethian 1996), 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 velocity background media containing high velocity convex objects.

**Fast travel time/path finding**

For the object based velocity model proposed in the previous discussion, we now prove that a travel path finding algorithm can be built using induction on the number of objects included in the model. If there are $N$ objects in the model and their velocities are different from each other, the computational complexity will be $O(N!)$. 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, $V(\theta_1)$, then the travel path between a transmitter $\alpha$ and a receiver $\beta$ will be either the direct path connecting transmitter-receiver points $\alpha$ and $\beta$ or the one passing through the object, whichever is faster (see Fig. 3(a)). Thus, the travel
time function for any two points $\alpha$ and $\beta$ with a single object $\theta_1$ in the model is the fastest of two possible travel paths:

$$\tau^*(V(\theta_1), \alpha, \beta) = \min \begin{cases} 
1/v_h \cdot d(\alpha, \beta) \\
1/v_h \cdot (d(\alpha, \theta_1) + d(\theta_1, \beta)) + 1/v_1 \cdot d(\gamma, \zeta)
\end{cases} \quad (6)$$

where $\gamma, \zeta$ are the closest points to $\alpha, \beta$ in $|\theta_1|$, which are defined as:

$$\gamma = \arg \min_{\gamma} \|\alpha - \gamma\|, \quad \zeta = \arg \min_{\zeta} \|\beta - \zeta\|, \quad \gamma, \zeta \in |\theta_1|. \quad (7)$$

Because of the homogeneous background and convex object assumption, the travel path passing through the object is the combination of the shortest distance from $\alpha$ to $|\theta_1|$, a path inside $|\theta_1|$ and the shortest distance from $|\theta_1|$ to $\beta$, which are all straight lines and denoted $P(\alpha, \gamma)$, $P(\gamma, \zeta)$ and $P(\gamma, \beta)$ respectively. This travel path can be viewed as $\alpha - \theta_1 - \beta$ which describes the order of objects passing through. From the high velocity object assumption, the travel time inside $|\theta_1|$ is negligible, thus, the travel time function in the one-object case becomes:

$$\tau^*(V(\theta_1), \alpha, \beta) = \min \begin{cases} 
1/v_h \cdot d(\alpha, \beta) \\
1/v_h \cdot (d(\alpha, \theta_1) + d(\theta_1, \beta)) + 1/v_1 \cdot d(\gamma, \zeta)
\end{cases} \quad (8)$$

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 (8) we know how to compute the travel path/time for the one-object cases, which are $\alpha - \theta_1 - \beta$ and $\alpha - \theta_2 - \beta$. Thus, all we need to do now is to compute two new paths which pass through both $|\theta_1|$ and $|\theta_2|$, and compare them to the previous results.

We notice that the path $\alpha - \theta_1 - \theta_2 - \beta$ includes the shortest paths from $\alpha$ to object $\theta_1$, from object $\theta_1$ to object $\theta_2$, and from object $\theta_2$ to $\beta$. Similar to (8), the corresponding travel time will be $1/v_h \cdot (d(\alpha, \theta_1) + d(\theta_1, \theta_2) + d(\theta_2, \beta))$. But $d(\alpha, \theta_1)$ has been calculated
Figure 3: The travel path will be the faster one between the direct path or the one through high velocity object

in the previous step when finding the path $\alpha - \theta_1 - \beta$, and likewise $d(\theta_2, \beta)$ is also available. Thus $d(\theta_1, \theta_2)$ is the only new quantity to be calculated. Now it is clear that the travel path $\alpha - \theta_1 - \theta_2 - \beta$ can be found efficiently based on previous results. Similarly for $\alpha - \theta_2 - \theta_1 - \beta$, only $d(\theta_2, \theta_1)$ would be needed, but $d(\theta_2, \theta_1) = d(\theta_1, \theta_2)$ and thus no new distance needs to be computed.

Now the forward model for calculating the travel time given two objects in velocity model $\tau^*(V(\theta_1, \theta_2), \alpha, \beta)$ can be simplified as

$$\tau^*(V(\theta_1, \theta_2), \alpha, \beta) = 1/v_h \cdot \min \left\{ \begin{array}{l}
d(\alpha, \beta) \\
d(\alpha, \theta_1) + d(\theta_1, \beta) \\
d(\alpha, \theta_2) + d(\theta_2, \beta) \\
d(\alpha, \theta_1) + d(\theta_1, \theta_2) + d(\theta_2, \beta) \\
d(\alpha, \theta_2) + d(\theta_1, \theta_2) + d(\theta_1, \beta)
\end{array} \right\}.$$ (9)
Note that compared to the single object case in (8), in (9) we only need to compute three new terms \( d(\alpha, \theta_2), d(\theta_1, \theta_2) \) and \( d(\theta_2, \beta) \). By induction, it follows that when we add the \( N \)th object \( \theta_N \) the new terms that need to be computed will be \( d(\alpha, \theta_N) \), \( d(\theta_1, \theta_N), \ldots, d(\theta_{N-1}, \theta_N) \) and \( d(\theta_N, \beta) \). 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 \( O(N^2) \).

**Dijkstra path finding**

Note that in (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) (sources, destinations or objects, see Figure 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 (Dijkstra, 1959) (see Algorithm 1).

For example, in Fig. 4(b) we can construct the corresponding fully connected graph for velocity model \( V(\theta_1, \theta_2) \) with 4 nodes as shown in Fig. 4(b). We show the update of the path tracking algorithm from the source \( \alpha \) to each node for a numerical example with the same topology as the example of Fig. 4(b) in Fig. 5.
Figure 4: Graph representation of path tracking. (a) The distance between objects (b) Graph and distance metric

**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, $\{\vec{\theta}_1, \vec{\theta}_1'\}$, where $|\vec{\theta}_1'|$ is a dilation of $|\vec{\theta}_1|$. In other words, $\vec{\theta}_1'$ is a convex object with larger size but with the same overall shape as $\vec{\theta}_1$, thus, $|\vec{\theta}_1|$ is a subset of $|\vec{\theta}_1'|$, $|\vec{\theta}_1| \subset |\vec{\theta}_1'|$ (as shown in Fig. 4).

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 5: The example for Dijkstra algorithm. Note in (b) the $\text{dist}[\theta_2]$ is $\text{dist}[\alpha] + e(\alpha, \theta_2) = 5$, and in (c) after we add $\theta_1$ the $\text{dist}[\theta_2]$ is updated to $\text{dist}[\theta_1] + e(\theta_1, \theta_2) = 4$.
Lemma 1. Consider two high velocity objects, \( \{\theta_1, \theta'_1\} \), where \(|\theta'_1|\) is a dilation of \(|\theta_1|\), \(|\theta_1| \subset |\theta'_1|\). With the other \( N-1 \) objects fixed, consider two alternative velocity models \( V(\theta_1, \theta_2, \ldots, \theta_N) \) and \( V(\theta'_1, \theta_2, \ldots, \theta_N) \) using \( \theta_1 \) and \( \theta'_1 \) respectively. For the travel time function, we have \( \tau^*(V(\theta_1, \ldots, \theta_N), \alpha, \beta) \geq \tau^*(V(\theta'_1, \ldots, \theta_N), \alpha, \beta) \forall \alpha, \beta \). Thus, the travel time is monotonically non-increasing with respect to the size of high velocity object. *\n
Proof. The proof is straightforward, given that weights in the graph are smaller or equal with the same topology, the shortest path will be shorter.

\[\square\]

**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

*This property still holds for any even \(|\theta_1| \in |\theta'_1|\), even \(|\theta_1| \) and \(|\theta'_1| \) have different shapes.*
becomes ill-posed and the solution may not be unique. In this case, searching one single most possible model provides little information for the solutions of 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 (Tarantola, 1987). We start this section by introducing some notations.

Notations

The input data is obtained by measuring the travel time between the set of transmitters $\mathcal{A} = \{\alpha_1, \ldots, \alpha_{tx}\}$ and receivers $\mathcal{B} = \{\beta_1, \ldots, \beta_{rx}\}$. We denote the measured travel time for all transmitter-receiver pairs $(\alpha_i, \beta_j)$ as a vector $t = \{t_1, \ldots, t_n\}$, 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, $\Theta = \{\theta_1, \ldots, \theta_N\}$, thus the velocity model $V(\theta_1, \ldots, \theta_N)$ can be represented simply by $V(\Theta)$. Then we define the travel time function from the forward model, $T(\Theta, \mathcal{A}, \mathcal{B})$, as a vector function representing the travel time between each pair of transmitters and receivers based on the velocity model with parameter $\Theta$:

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

where

$$\begin{cases}
T_1(\Theta, \mathcal{A}, \mathcal{B}) &= \tau^*(V(\theta_1, \ldots, \theta_N), \alpha_1, \beta_1) \\
& \vdots \\
T_n(\Theta, \mathcal{A}, \mathcal{B}) &= \tau^*(V(\theta_1, \ldots, \theta_N), \alpha_{tx}, \beta_{rx}).
\end{cases}$$

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:

$$E(\Theta) = \|t - T(\Theta, \mathcal{A}, \mathcal{B})\|^2.$$
We use the Bayesian approach (Tarantola, 1987), which leads us to update the belief for different models after accounting for the observations. The likelihood function \( L(\Theta) \), which is modeled as a Gaussian uncertainty of the error, measures the confidence on different models:

\[
L(\Theta) = \tilde{k} \cdot e^{-E(\Theta)},
\]

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

\[
\sigma(\Theta) = k \cdot \rho(\Theta)L(\Theta),
\]

where \( k \) is again a normalization constant. For the rest of this paper, we assume a uniform distribution for the prior probability \( \rho(\Theta) \), 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 \( E(\Theta) \). In the next section, we will introduce our algorithm to estimate the error function.

**Proposed algorithm**

Because the travel time function is nonlinear, the error function \( E(\Theta) \) will have a complex and multi-modal shape. We can “sample” the error function at any point \( \Theta \) by calculating the value of travel time function from forward model for the corresponding parameters. 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 (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 [MacKay, 2003]. 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.

**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 [Duane et al., 1987], 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 [Kirkpatrick, 1984], to draw more samples near the local minima.

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

$$\dot{\Theta} = p, \quad \dot{p} = -\frac{\partial E(\Theta)}{\partial \Theta} - \epsilon p. \quad (15)$$

To sample the error function, we simulate and record the states of particle movement. With a randomized momentum $p$, we solve the Hamiltonian dynamics in (15) during a simulated time of duration $t$. In the dynamical movement, the momentum variable $p$
determines where the parameter \( \Theta \) goes, and the gradient of potential function determines the change of momentum \( p \). The friction term \( \epsilon p \) decides the loss of total system energy.

Starting with the initial value \( \Theta_0 \), we define the simulation time \( t \) and steps \( \Delta t \), then use (15) to identify the successive steps in the walk through parameter space. During the simulated time \( t \), we can record the state variables \( [\Theta(\Delta t), p(\Delta t)], [\Theta(2\Delta t), p(2\Delta t)], \ldots, [\Theta(t), p(t)] \) which describes the variable movement on the error function. Then we take all the \( \Theta(\Delta t), \ldots, \Theta(t) \) as new sampling points for the error function.

In Fig 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 \( \Theta(t) \) as a new starting point for the next round of simulation by the Metropolis rule (Metropolis et al., 1953). This acceptance rule is based on the change of the error function, \( \Delta E(\Theta) = E(\Theta(t)) - E(\Theta(0)) \), where \( \Theta(0) \) is the current starting point. If \( \Delta E(\Theta) \leq 0 \), 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 \( \exp(-\Delta E(\Theta)) \geq r \). We iteratively simulate this dynamic system \( L \) times, and the detail algorithm is shown in Algorithm 2.

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

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 $N$th root of the number of samples, where $N$ is the number of parameters. 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:

\[
E(\Theta) = \|t - T(\Theta, A, B)\|^2
\]  
(17)

\[
= \|t_1 - T_1(\Theta, A, B)\|^2 + \|t_2 - T_2(\Theta, A, B)\|^2 + \ldots
\]  
(18)

\[
= E_1(\Theta) + E_2(\Theta) + \ldots,
\]  
(19)

where each separate error function has a quadratic form \( E_i(\Theta) = \|t_i - T_i(\Theta, A, B)\|^2 \).

Consider the change in the travel time function \( T_i(\Theta, A, B) \) with respect to the size \( s_j \) of object \( j \), while all other parameters are left unchanged. By Lemma I, 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, \( E_i(\Theta) \), with respect to a single object size will be a weakly unimodal function (see Fig 8).

![Figure 8](image)

(a) The travel time function \( T_i(\Theta, A, B) \) (b) The error function \( E_i(\Theta) = \|t_i - T_i(\Theta, A, B)\|^2 \). The travel time is monotonically non-increasing, therefore the error function is weakly unimodal.
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 to size parameters, $s_1, s_2, \ldots$. We choose one size parameter each time and use the above property. Because $E_i(\Theta)$ 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 (Kiefer, 1953) 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 $\Theta = \{\theta_1, \theta_2\}$ and two measured data points $t = \{t_1, t_2\}$, the error function is defined as $E(\Theta) = E_1(\Theta) + E_2(\Theta)$. Given an initial sample $\Theta(0)$, the parameters can be reordered as $\Theta(0) = \{s_1(0), s_2(0), \tilde{\Theta}(0)\}$. To find the re-sampling locations on the $s_1$ axis, we run the golden section search $k$ times along the $s_1$ dimension for each of $E_1(\Theta)$ and $E_2(\Theta)$. The re-sampling locations will be $\{s_1(1), s_2(0), \tilde{\Theta}(0)\}, \ldots, \{s_1(k), s_2(0), \tilde{\Theta}(0)\}$ and $\{s_1(k+1), s_2(0), \tilde{\Theta}(0)\}, \ldots, \{s_1(2k), s_2(0), \tilde{\Theta}(0)\}$ 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 we choose $2k$ re-sampling locations for the error function on the axis $s_1$ based on $E_1$ and $E_2$ separately. The reason is that the error function is the sum of each separate error function $E(\Theta) = E_1(\Theta) + E_2(\Theta)$, which may not be unimodal (sum of unimodal
functions is not necessary unimodal, see Fig. 9). Thus, running the golden section search directly on $E(\Theta)$ may not 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 Fig. 10 we show an example of how this approach works better than choosing sampling locations directly on the total function.

![Figure 9: Sum of two unimodal functions. Note the sum of two unimodal functions is not necessary a unimodal function.](image)

**SIMULATION RESULTS**

Following our previous assumptions, in our simulations we use a velocity ratio $v/v_h \to \infty$ and choose “line” as the pre-defined convex geometrical object to model the HVS. The model
Figure 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.
parameters $\theta_j = \{s_j, c_j, \psi_j\}$ will be the length, line center and 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 $\{\Theta_1, \ldots, \Theta_N\}$ N points in the parameter space, we define a mapping function $f(\Theta)$ which maps the object parameters into object shapes in spatial domain. Then we calculate the average:

$$M_f = \int \sigma(\Theta) f(\Theta) d\Theta \approx \sum_{i=1}^{N} \sigma(\Theta_i) f(\Theta_i). \quad (20)$$

Because $f(\Theta)$ represents the high velocity structure in spatial domain, $M_f$ can be viewed as the “appearance probability map” of the high velocity in different region.

In simulations, we need to choose the random momentum $p$ and friction $\epsilon \cdot p$. We draw the momentum from a normal distribution $N(0, \rho)$, where $\rho$ is equal to the 10% of the maximum possible value of $\Theta$. The friction coefficient $\epsilon$ 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 Fig. 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 Fig. 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.

The estimated PDF from the two different approaches are shown in Fig. 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.

![Figure 11: The (a) geometry of sensor settings and (b) (c) the PDF for experiment 1.](image)

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 PDF and our estimation are shown in Fig. 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 $E(\Theta) \leq \delta$, 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 Fig. 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
Figure 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 $\psi$ but cover the whole $s$ axis.

Figure 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.
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.

![Figure 14: The PDF and our estimation in experiment 2. Note comparing to experiment 1, the PDF has sharper changes which implies higher model resolution.](image)

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.

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:

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

(21)

where $d(\alpha, \beta)$ is the distance function of $[2]$. 

33
Figure 15: (a) The ground truth and (b) the appearance probability map. It shows several different models closed to ground truth all have high probability.

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\cite{Rote1991} and is widely used in computer vision to measure the difference between 3D curves or binary images\cite{Huttenlocher}. It also has an interesting property in that when \( d_H(X, Y) = 0 \) then \( X \) and \( Y \) 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 \( X = Y \). From previous discussion, for a given model parameter \( \Theta \), \( f(\Theta) \) 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 \( \Theta_{\text{truth}} \) can be written as

\[
\mathcal{E}(\Theta) = \int d_H(f(\Theta), f(\Theta_{\text{truth}})) \cdot \sigma(\Theta) \, d\Theta
\]

(22)

\[
\approx \sum_{i=1}^{N} d_H(f(\Theta_i), f(\Theta_{\text{truth}})) \cdot \sigma(\Theta_i),
\]

(23)
where $\Theta_i$ are sampled models.

We list the error metric for $5 \times 5, 10 \times 10, 25 \times 25, 50 \times 50$ transmitter-receiver pairs (see Fig. 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 Fig. 17)

Figure 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.

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 Fig. 19 and 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
Figure 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.

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 that the vertical resolution is very limited. This is the reason we have “phantoms” in the vertical direction and the error metric (see Fig. 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 (12)), thus, the effect of shot noise will be “averaged”
Figure 18: (a) Ground truth and (b) error metric for experiment 4. Note the error metric does not decrease monotonically due to the vertical phantom.

Figure 19: (a) Grid-based result and (b) appearance probability map for 10 * 10 sensors in experiment 4
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 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.
REFERENCES

Agarwal, P., and S. Suri, 1998, Surface approximation and geometric partitions: SIAM Journal on Computing, 27, 1016–1035.

Bai, C., and S. Greenhalgh, 2005, 3-d non-linear travel-time tomography: Imaging high contrast velocity anomalies: Pure and Applied Geophysics, 162, 2029–2049.

Berryman, J., 1989, Fermat’s principle and nonlinear traveltime tomography: Physical review letters, 62, 2953–2956.

——, 1991a, Lecture notes on nonlinear inversion and tomography: 1, borehole seismic tomography: Technical report, Lawrence Livermore National Lab., CA (United States).

Berryman, J. G., 1990, Stable iterative reconstruction algorithm for nonlinear traveltime tomography: Inverse Problems, 21–42.

——, 1991b, Lecture notes on nonlinear inversion and tomography: Technical report.

Courant, R., and D. Hilbert, 1962, Methods of mathematical physics: CUP Archive, 1.

Dijkstra, E., 1959, A note on two problems in connexion with graphs: Numerische mathematik, 1, 269–271.

Duane, S., A. Kennedy, B. Pendleton, and D. Roweth, 1987, Hybrid monte carlo: Physics letters B, 195, 216–222.

Engl, H., M. Hanke, and A. Neubauer, 1996, Regularization of inverse problems: Springer Netherlands.

Huttenlocher, D., G. Klanderman, and W. Rucklidge, 1993, Comparing images using the Hausdorff distance: Pattern Analysis and Machine Intelligence, IEEE Transactions on, 15, 850–863.

Kiefer, J., 1953, Sequential minimax search for a maximum: Proc. Amer. Math. Soc, 502–506.
Kirkpatrick, S., 1984, Optimization by simulated annealing: Quantitative studies: Journal of statistical physics, 34, 975–986.

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

MacKay, D., 2003, Information theory, inference, and learning algorithms: Cambridge Univ Pr.

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

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

National Research Council (US). Committee on Fracture Characterization and Fluid Flow, 1996, Rock fractures and fluid flow: contemporary understanding and applications: National Academy Press.

Ng, A., 1997, Preventing overfitting of cross-validation data: Proceedings of the Fourteenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., 245–253.

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

Rote, G., 1991, Computing the minimum Hausdorff distance between two point sets on a line under translation: Information Processing Letters, 38, 123–127.

Sethian, J., 1996, A fast marching level set method for monotonically advancing fronts: Proceedings of the National Academy of Sciences of the United States of America, 93, 1591.
Tarantola, A., 1987, Inverse problem theory: Methods for data fitting and model parameter estimation.

Tsang, Y., M. Coates, and R. Nowak, 2003, Network delay tomography: Signal Processing, IEEE Transactions on, 51, 2125–2136.

Vasco, D., and A. Datta-Gupta, 1997, Integrating field production history in stochastic reservoir characterization: SPE Formation Evaluation, 12, 149–156.

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

Wilson, D., A. Ziemann, V. Ostashev, and A. Voronovich, 2001, An overview of acoustic travel-time tomography in the atmosphere and its potential applications: Acta Acustica united with Acustica, 87, 721–730.
Algorithm 1 Dijkstra algorithm for path tracking

for $v \in G$ do
  \begin{align*}
  & \text{dist}[v] = \infty \quad \triangleright \text{Unknown distance from source to } v \\
  & \text{previous}[v] = \emptyset \quad \triangleright \text{Previous node in optimal path from source} \\
  & \text{dist[source]} = 0 \quad \triangleright \text{Distance from source to source} \\
  & Q := \forall v \in G \quad \triangleright \text{Put all nodes in } Q \text{ to be scanned}
  \end{align*}
end for

while $Q \neq \emptyset$ do
  \begin{align*}
  & u := v \in Q \text{ with minimum dist}[v] \quad \triangleright \text{Start node in first case} \\
  & Q = Q \setminus u \quad \triangleright \text{remove } u \text{ from } Q \\
  & \text{if } \text{dist}[u] = \infty \text{ then} \\
  & \quad \text{break} ; \quad \triangleright \text{all remaining vertexes are inaccessible from source}
  \end{align*}
  \begin{align*}
  \text{else} \\
  & \quad \text{for } \forall \text{ neighbor } v \text{ of } u \text{ do} \quad \triangleright \text{where } v \text{ is still in } Q \\
  & \quad \quad \text{alt} = \text{dist}[u] + e(u, v) \\
  & \quad \quad \text{if } \text{alt} \leq \text{dist}[v] \text{ then} \quad \triangleright \text{update the distance for } v \\
  & \quad \quad \quad \text{dist}[v] = \text{alt} \\
  & \quad \quad \quad \text{previous}[v] = u \\
  & \quad \quad \text{end if} \\
  & \quad \text{end for}
  \end{align*}
end if
end while
Algorithm 2 Modified Hamiltonian Monte Carlo method

\[ \Theta(0) = \Theta_{init} \quad \triangleright \text{Initialize} \]

for \( l = 1 : L \) do \quad \triangleright \text{loop L times}

\[ g = \nabla E(\Theta(0)) \quad \triangleright \text{set gradient using initial parameter} \]

\[ E = E(\Theta(0)) \quad \triangleright \text{set error function value} \]

\[ p \leftarrow \text{randn( size}(\Theta(0)) \text{)} \quad \triangleright \text{initialize momentum} \]

\[ \Theta_{new} = \Theta(0), \quad g_{new} = g \]

for \( t_{sim} = 0 : \Delta t : t \) do \quad \triangleright \text{Use “leapfrog” steps to simulate the dynamics}

\[ p = p - \delta \cdot g_{new}/2 \]

\[ \Theta_{new} = \Theta_{new} + \delta \cdot p \]

Sample list \( \leftarrow \Theta_{new} \quad \triangleright \text{Record the state samples} \]

\[ g_{new} = \nabla E(\Theta_{new}) \quad \triangleright \text{Update the gradient and momentum} \]

\[ p = p - \delta \cdot g_{new}/2 - \epsilon \cdot p \]

end for

\[ E_{new} = E(\Theta_{new}) \quad \triangleright \text{Find the final value of error function} \]

\[ \Delta E = E_{new} - E \quad \triangleright \text{Use Metropolis rule to decide the new starting point} \]

if \( \Delta E < 0 \) then

\[ \Theta(0) = \Theta(t) \quad \triangleright \text{Accept the new starting point} \]

else

if \( \exp(-\Delta E) \geq \text{rand()} \) then

\[ \Theta(0) = \Theta(t) \quad \triangleright \text{Accept the new starting point if } \exp(-\Delta E) \leq \text{a random number} \]

end if

end if

end if

end for