Distributed Tomography with Adaptive Mesh Refinement in Sensor Networks

Goutham Kamath*

Department of Computer Science,
Georgia State University,
Atlanta, GA, 30303, USA
Email: gkamath1@student.gsu.edu
*Corresponding author

Lei Shi

Department of Computer Science,
Georgia State University,
Atlanta, GA, 30303, USA
Email:lshi1@student.gsu.edu

Edmond Chow

College of Computing,
Georgia Institute of Technology,
Atlanta, GA, 30332, USA
Email:echow@cc.gatech.edu

Wen-Zhan Song

Department of Computer Science,
Georgia State University,
Atlanta, GA, 30303, USA
Email:wsong@gsu.edu

Abstract: Existing seismic instrumentation systems do not yet have the capability to recover the physical dynamics with sufficient resolution in real time. Currently, seismologists use centralized tomography inversion algorithm for which the data is gathered either manually from each station or by using limited number of expensive broadband stations. This scheme can take months to generate tomography and also lack the resolution due to limited number of sensors. It also introduces a bottleneck in computation and increases the risk of data loss in case of node failures, especially the base station. To address these issues a distributed approach is required which can avoid costly data collection from large number of sensors and perform in-network imaging to obtain high resolution real-time tomography. In this paper, we present a distributed adaptive mesh refinement solution to invert seismic tomography over large dense network, which avoids centralized computation and expensive data collection. Our approach first discretizes the high fidility data and later filters them using adaptive mesh to make it well-conditioned. We show that this filtered well conditioned system has lower dimension and improved convergence rate than the original system, thereby decreasing the communication overhead over the network. The system is implemented and evaluated using a CORE emulator and the results show that our method is able to obtain high-resolution images in real-time by distributing the computation load over the network.

Keywords: Distributed Sensing, Adaptive Mesh, Seismic Tomography, Sensor Network, In-network Computing

Reference to this paper should be made as follows: Kamath, G.,Shi, L., Chow, E. and Song W. (2015) 'Distributed Tomography with Adaptive Mesh Refinement in Sensor Networks', *Int. J. Sensor Networks*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Goutham Kamath received his B.E. degree from India in 2009 and M.S in Electrical Engineering from University of Wyoming in 2012. He is currently pursuing his PhD in the Department of Computer Science, Georgia State University. His research interests include wireless sensor networks, distributed systems and mobile ad-hoc networks.

Lei Shi received his BSc degree in Software Engineering from Tongji University in 2007, and the MSc degree in Computer Science from Shanghai Jiao Tong University in 2010. He is currently pursuing the PhD degree in the Department of Computer Science of Georgia State University. His research interests include wireless sensor networks, in-network processing and distributed systems.

Edmond Chow did his B.A.Sc. from University of Waterloo in 1993 and PhD from University of Minnesota, 1997. After graduating he worked in LLNL until 2005 and later worked at D.E. Shaw research until 2010. He is currently an associate professor at Georgia Tech. His research interest includes developing and applying numerical methods and high-performance computing to solve large-scale scientific problems.

Wen-Zhan Song is a professor in Georgia State University. His research mainly focuses on sensor web, smart grid and smart environment where sensing, computing, communication and control play a critical role and need a transformative study. His research has received 6 million+ research funding from NSF, NASA, USGS, Boeing and etc since 2005, and resulted in 80+ journal articles, conference articles and book chapters in this area.

1 Introduction

Current volcano data collection and monitoring systems lack the capability of obtaining real time information and also recovering the physical dynamics of seismic activity with sufficient resolution. At present, the seismic tomography process involves aggregating raw data from seismic sensors into a centralized server for post-processing and analysis. The raw seismic samples are typically in the range of $16-24$ bit at $50-200$Hz. These high precision sampling from each node makes it extremely difficult to collect raw, real-time data from a large-scale dense sensor network due to severe limitations on energy and bandwidth. Due to these restrictions many of the threatening active volcanoes worldwide use fewer than 20 nodes (Song et al., 2009), limiting our ability to understand dynamics and physical processes of volcanoes in real-time. The centralized solution introduces a bottleneck in computation and also increases the risk of data loss in case of node failures. The advancement in current wireless sensor technology makes it possible to deploy and maintain a large-scale network for environmental monitoring and surveillance. However, the currently used tomography algorithms cannot be easily implemented under this distributed scenario as it relies on centralized processing. Thus, real-time volcano tomography requires a practical approach which is distributed, scalable, and efficient with respect to tomography computation.

Tomography can be defined as the science of computing reconstructions in 2D and 3D from projections, i.e., solving the system obtained by integrations along the rays that penetrate a domain Ω, typically a rectangle in 2D, and a box in 3D. Let us assume that there are P number of sensors deployed for volcano monitoring. The discretized version of this problem takes the form

$$\mathbb{A}x = \mathbb{B} \tag{1}$$

where,

$$\mathbb{A} = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_p \end{pmatrix}; \mathbb{B} = \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_p \end{pmatrix}; A_i \in \mathbb{R}^{m_i \times n}; B_i \in \mathbb{R}^{m_i}$$

where, B_i is a data or measurement vector, $x \in \mathbb{R}^n$ is a n-dimensional model parameter vector and $A_i \in \mathbb{R}^{m_i \times n}$ is a data kernel which acts as a linear operator that establishes the relationship between data and model parameters in i-th sensor. Further details regarding the formulation are provided in section 3.

To recover the magma/slowness model x, geophysicists are interested in solving (1) and at present these inversion methods rely on centralized data gathering scheme (sending $\{A_i, B_i\}$ to a base station) and has been implemented on volcanoes such as Mount St. Helens (Lees, 1992), Mount Rainier (Moran et al., 1999) as well as many others. Currently, seismic community faces two major hurdles in-order to obtain both real-time and high-resolution tomography. First, the data logged at each stations are of high fidelity and due to bandwidth limitations, information from only a small number of sensors can be obtained for real time imaging. Now, with limited sensors they cannot achieve high resolution tomography. On the other hand, large arrays of sensors can be deployed to obtain high-resolution tomography. However, limited bandwidth restricts the transfer of large volume of data from these sensors in which case they resort for manual data gathering which takes months for post processing and imaging. So to obtain high resolution and real time imaging we need large number of sensor stations which has the capability of performing in-network computing and also avoid costly data collection. The authors in (Liu et al., 2013) have discussed the use of low-cost sensor for P-phase detection of earthquake. The earthquake hypo-center detection forms the basic step for seismic tomography and we extend this further to obtain in-network imaging. Here, we assume that the sensors used are low-cost and has low computational power eg: Raspberry Pi/Beagle-Bone/Android.

In this paper we present a distributed approach to solve tomography problem using quadtree/octree based adaptive mesh refinement (AMR). Quadtree/octree based parametrization reduces the data overhead in the model space while maintaining enough information that are needed for image reconstruction. In this method, we simultaneously discretize the high fidelity data on each node and later filter them using adaptive mesh to make it well-conditioned. These filtered well conditioned systems have lower dimension than the original problem and we prove that it has improved

convergence rate. This method is implemented and evaluated using a CORE emulator and the results show that we are able to obtain high-resolution images in real-time by distributing the computation load over the network. We also compare this method with (Kamath et al., 2013) and show that this is superior in-terms of communication cost and also message overhead.

The rest of the paper is organized as follows. Section 2 presents related work on adaptive mesh and iterative methods for reconstruction. In section 3 we provide background on seismic tomography inversion and present the problem formulation. In section 4 we first discuss the advantages of AMR and explain how AMR is achieved using quadtree/octree decomposition. In section 5 we introduce the algorithm distributed tomography using adaptive mesh refinement (DT-AMR) that solves seismic tomography problem in sensor networks. Simulation results are shown in section 7. Finally we conclude the paper in section 8.

2 Related Work

The tomography inversion process mainly involve solving large sparse system of linear equations (1). Due to the property of sparsity and its sheer volume, iterative methods became almost mandatory as they are more efficient in-terms of memory and computational requirements compared to direct methods (Saad, 2003). Several parallel and distributed iterative methods have been developed and are currently used to solve a large variety of problems (Heath et al., 1991; Bertsekas and Tsitsiklis, 1991). All these methods are developed mainly for GPU computing, where communication bandwidth does not cause significant bottleneck. However in our case, the sensors which are deployed in harsh environment communicate through limited bandwidth and has resource constraints such as power.

A popular iterative method for solving overdetermined systems was proposed by Kaczmarz (KACZ) (Kaczmarz, 1937) which is an alternating projection method. This method is also known under the name Algebraic Reconstruction Technique (ART) in computer tomography (Herman, 1980). This algorithm does not require the full matrix to be in memory at one time and can incorporate new information (ray paths), on the fly. The vectors of unknowns are updated after processing each equation of the system and this cycle repeats until convergence. These iterative algorithms are distributed by averaging the boundary information, e.g., Component Averaging (CAV) (Censor et al., 2001) and Component Averaged Row Projections (CARP) (Gordon and Gordon, 2005). A survey paper comparing various block parallel methods based on their performance on GPU's is (Elble et al., 2010).

(Kamath et al., 2013) proposed an algorithm called component average distributed multi-resolution evolving tomography (CA-DMET) which involved modification of component average type algorithms such as CAV and CARP for seismic tomography. This was the first algorithm which was designed to run distributedly to solve tomography problem. Although they were able to obtain the image it failed to deliver images with sufficient resolution and the convergence stalled after certain iteration. Moreover, their algorithm was developed using regular grids i.e. the partial differential equation is discretized over a regular cell of same dimensions. In this paper, our main goal is to show that regular grid partition is not suitable for distributed tomography and we develop irregular grid method which outperforms CA-DMET in terms of convergence and also communication overhead.

Adaptive mesh refinement (AMR) has been studied widely and has been used as discretization tool for partial differential equation as early as 1980 (Berger and Oliger, 1984). However, only until early 90's it was used by seismology community to solve inverse problem on small set of data (Michelini, 1995). (Vesnaver, 1996) used SVD to interactively change the boundaries while, (Curtis and Snieder, 1997) used genetic algorithm to optimize the parametrization. These algorithms were suitable for small size data sets and required high computational power to run efficiently. (Spakman and Bijwaard, 2001) came up with a less computation intensive solution to parametrize the coefficient and this algorithm could run efficiently even for large matrices. However, this algorithm is only suitable for centralized architecture and is not feasible to be implemented in a distributed scenario. Adaptive mesh has lots of advantages over regular mesh in terms of tomography inversion and we will discuss this in section 4. To perform adaptive mesh in a distributed case, we had to come up with some novel method which was computationally light and also satisfy the requirements such as faster convergence. To the best of our knowledge, our work is the first attempt to compute seismic tomography using adaptive mesh over a distributed sensor networks.

3 Problem Formulation

In this section we will first provide an insight on seismic tomography and later explain the formulation of discretized inverse problem which we intend to solve. For the sake of simplicity we assume 2D model in order to derive the equations and can be easily extended to 3D case. Let us assume there are P seismic sensors deployed on a square domain $\Omega = [0,1] \times [0,1]$ (in arbitrary units), and also we are given a unknown function (slowness or velocity parameter) $f(t) = f(t_1, t_2)$ that we wish to reconstruct. This function creates damping of the signal (p-wave) caused due to earthquake which penetrates an infinitesimally small part $d\tau$ of a ray at position $\mathbf{t}$ such that it is proportional to the $f(t)(d)\tau$. The data in tomography problem consists of measurements of the damping of signals following ray tracing through the domain Ω. See (Iyer and Dawson, 1993) for more details on mathematical formulation.

Let us first view this problem in terms of ith seismic node. The eth measurement b_e, for $e = \{1, \cdots, m_i\}$ represents the total damping of signal e that penetrates Ω as shown in Figure 1. Here, m_i denotes number of earthquake events detected by i-th sensor node. Now, all the points t_e on ray_e can be obtained by,

$$t_e(\tau) = t_{e,0} + \tau d_e, \quad \tau \in \mathbb{R} \tag{2}$$

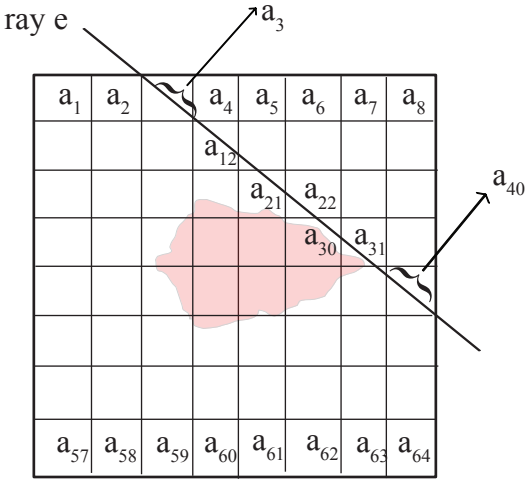


Figure 1 Example of discretized tomography with $N = 8$ in sensor i. The e-th ray intersect a total of 7 pixels, and thus the e-th row of matrix A_i has 7 non-zero elements (in columns 3, 12, 21, 22, 30, 31, and 40)

where $t_{e,0}$ is an arbitrary point on the ray, and d_e is a (unit) vector along the direction of the ray. Due to the above assumption, the damping associated with the e-th ray is given by,

$$b_e = \int_{-\infty}^{\infty} f(t_e(\tau))\mathrm{d}\tau \quad e = \{1, \cdots, m_i\} \tag{3}$$

where $\mathrm{d}\tau$ denotes the integration along the ray.

Next we will show how the continuous function f(t) can be discretized by dividing Ω into grids of pixels. These grids can be regular or irregular, each one having its own advantages and disadvantages. This papers core idea is to show how discretization based on irregular grid can improve the convergence rate thereby decreasing the communication cost. We also show that irregular mesh reduces the packet overhead while achieving better result compared to regular grids on a distributed network. But before that we shall explain how the problem is formulated on a regular $N \times N$ grid and later in the next section we will show how it can be modified based on irregular grid using adaptive mesh refinement technique.

When (3) is discretized on $N \times N$ array of pixels assuming $f(t)$ takes a constant value f_{kl} over the pixels (k,l)

$$f(t) = f_{kl} \quad \text{for } t_1 \in I_k \text{ and } t_1 \in I_l$$

where,

$$I_k = [(k-1)/N, k/N], \quad k = \{1, \cdots, N\}$$
$$I_l = [(l-1)/N, l/N], \quad l = \{1, \cdots, N\}$$

Substituting (3) into (3)

$$b_e = \textstyle\sum_{(k,l)\in ray_e} f_{kl}\Delta L^e_{kl}$$

where,

$$\Delta L^e_{kl} = \text{length of } ray_e \text{ in pixel (k,l)}$$

We can re-write the above equation in a more simplified form by numbering pixel (k,l) with j i.e. $x_j = f_{kl}$, where $j = (l-1)N + k$, we get

$$b_e = \sum_{j=1}^{n} a_{ej}x_j, \quad e = \{1, \cdots, m_i\}; \quad n = N^2$$

In the matrix form the equation formed at the ith sensor node is given by,

$$A_i x_i = B_i \qquad A_i \in \mathbb{R}^{m_i \times n}, x \in \mathbb{R}^n \tag{4}$$

where,

$$B_i = [b_1, \cdots, b_{m_i}]^T$$
$$a_{ej} = \begin{cases} \Delta L^e_{kl} & \text{(k,l)} \in ray_e \\ 0 & \text{otherwise} \end{cases}$$

Now for P number of sensors i.e. $i \in \{1, \cdots, P\}$ we have

$$\mathbb{A}x = \mathbb{B} \tag{5}$$

where,

$$\mathbb{A} = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_P \end{pmatrix}; \mathbb{B} = \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_P \end{pmatrix}; A_i \in \mathbb{R}^{m_i \times n}; B_i \in \mathbb{R}^{m_i}$$

Current algorithms that solves (5) assumes a centralized setup that involve collection of data $\{A_i, B_i\}$ from all the sensor nodes (mostly manually or sometimes using expensive broadband station) to a centralized server. Although the centralized techniques are well established and prove to be more efficient in terms of final solution the process involved in data gathering turns out to be extremely slow (several months). With the increasing demand for high resolution and real-time prediction there is a need for new distributed approach which is scalable and efficient in terms of computation. In this paper, we present a technique that solves (4) locally using AMR instead of (5) to obtain slowness/velocity model x with lower dimension. In our approach since we solve (4) locally, we avoid collecting A_i, B_i to a centralized server. In this method, computation is completely distributed however it involves exchanging of partial solution x which is of lower volume and sparse. Equation (4) and (5) falls under the category of inverse problems and the matrices arising from these equations are generally ill-posed. These ill-posed problem exhibit slower convergence rate which in our case increased communication over the networks. In the next section we will exploit the geometric features of the seismic tomography using AMR through which we can decrease the communication cost by making the problem well-posed.

4 Adaptive Mesh Refinement

Many geophysical inverse problems are ill-conditioned i.e. model space contains more details than it can be resolved using available data space (Bertero et al., 1985). Model space matrix A typically has large null space and because of this, few

portions of the solution cannot be resolved leading to its non-uniqueness. Geophysicists commonly use two approaches to overcome this problem: firstly, by making the problem well conditioned using some a-priori information such as smoothness constraints, regularization etc (Herman, 1980). However, obtaining reliable prior information is hard and also sometimes the smoothness constraints can be unrealistic. The second approach is by identifying the eigenvalue and eigenvector corresponding to the null space of model and later removing them explicitly. Although this method reduces the effective information content of the data set in the model space in a nontrivial manner, it can be used to obtain maximum amount of information that can be resolved from the current data set which will then reduce the amount of additional a-priori information to be included in the solution.

Removal of null space from the model data is equivalent to parameter reduction and here we will explain this with a simple example. Consider the path geometry of the ray produced by source (circle) and receiver (square) as shown in Figure 2. Let us suppose that we have an error-free measurement of average velocity and each of the four cells in Figure 2(a) have exactly equal ray path coverage. From measurements along the two left-hand paths, the quantity $v1 + v2$ can be determined exactly. The quantity $v1 - v2$, however, remains entirely unresolved by the data, and hence velocities v1 and v2 cannot be determined. Similarly, the combination $v3 + v4$ may be determined exactly, but not $v3 - v4$ and hence v3 and v4 cannot be determined.

Let $v = [v_1, \cdots, v_2]^T$, then well-determined combinations are $e_1.v$ and $e_2.v$. Whereas, the undetermined combinations are $e_3.v$ and $e_4.v$, where

$$e_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} ; e_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} ; e_3 = \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} ; e_4 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}$$

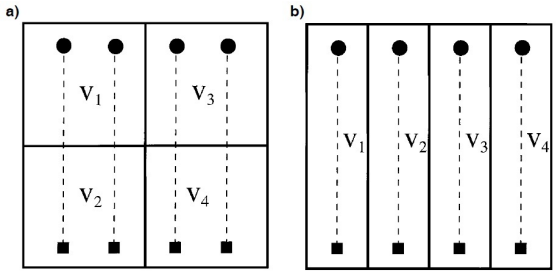


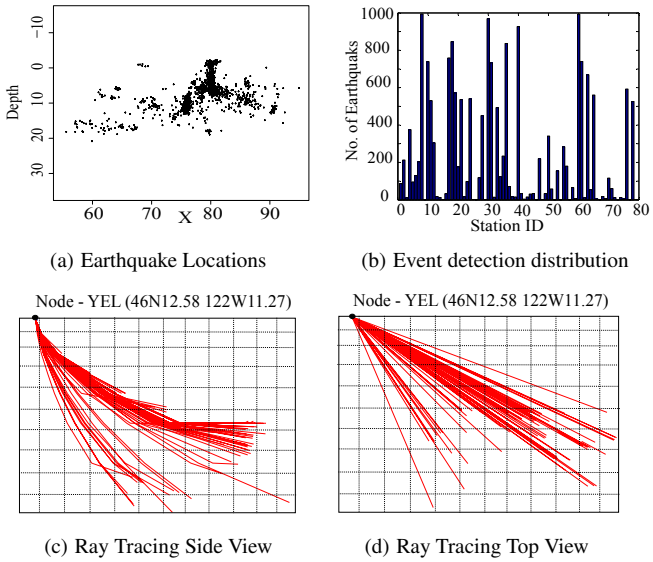
Figure 2 Cell geometries for the event (circle) station (square) paths shown. In (a) the cells bisect each path whereas in (b) each cell contains exactly one path length. Both cell geometries have exactly the same homogeneous path coverage within each cell

Vectors e_1 to e_4 are the eigenvectors of the inverse problem and v_1 to v_4 the velocities we are interested. Parameter combinations parallel to eigenvectors e_1 and e_2 are completely determined if and only if (iff) the corresponding eigenvalues are large, however combinations parallel to eigenvectors e_3 and e_4 are undetermined iff they correspond to small or zero eigenvalues.

Remark 1: *Removal of parameters that creates null space involve modification of the grid structure based on ray coverage which also changes the resolution and information content.*

4.1 Adaptive mesh in seismic tomography

Now we will provide the motivation for using adaptive mesh refinement in seismic tomography with the help of real data. The distribution of earthquake events and the ray coverage over a region of interest are typically non-uniform in nature. In Figure 3, we plot some data collected from the seismic sensors placed over Mt. St. Helens,WA,USA. This data set is obtained from 78 active stations spread across hundreds of km and the data life time is over a span of 10 years. Figure 3(a) shows around 1140 earthquake events distributed over a region. It is observed that these events are non-uniformly distributed having dense population near crater region compared to the rest. Figure 3(b) shows the histogram plot of number of events detected by each station. From this we infer that, few stations receive upto 1000 good events where as others fail to receive even a single event. This in turn corresponds to an uneven distribution of rays among each sensor nodes. Figure 3(c) and (d) plots the ray path traced by a station named YEL placed over top of volcano with coordinates $46N12.58122W11.27$. The entire region is viewed as a large cube discretized into regular sized voxels. From Figure 3(c) and (d), it is discerned that a large portion of the area is devoid of any penetration by these rays. These regular grid spacing creates ill-conditioned system as model space contains more detail than it can be resolved using available data space (Bertero et al., 1985) and this situation increases in case of distributed tomography. From analyzing data from two different sensor placed far apart, we saw that the ray coverage pattern in these nodes are different and it is often advantages to use different mesh refinement based on the local data to discretize rather than uniform mesh on all the nodes.



(a) Earthquake Locations (b) Event detection distribution

(c) Ray Tracing Side View (d) Ray Tracing Top View

Figure 3 Non-uniform distribution of rays and events at Mt St Helens.

Several methods like genetic algorithms, Singular Value Decomposition (SVD) and other direct methods have been used to reduce the null space. These methods are very effective in conditioning the problem, however these methods are not feasible to be implemented on sensors with low computational power and memory. For this reason our research direction focused on reducing the null space by studying the geometry of the problem which involves ray coverage. (Vesnaver, 1996) and (Spakman and Bijwaard, 2001) studied this problem of irregular grid extensively based on ray geometry and they pointed out the two situations that give rise to null space in tomography.

1. pixels/grids not crossed by any ray.
2. groups of two or more ray that are linearly dependent.

By using above results, we tried to remove null space by first performing the ray tracing on the regular grid of finest resolution and formulated the problem as shown in (4). The finest level of discretization of ray coverage will be of the order 128×128 for 2D and $128 \times 128 \times 128$ in case of 3D. Direct methods like SVD or genetic algorithm mentioned above will fail to compute null space considering the memory size and computational power of our sensors. So, we used a simpler technique of measuring hit count (rays passing in each grid) to identify the regions that are not crossed by the rays. Later, this criteria was used to merge the grids that did not have enough rays or had rays which where linearly dependent. Below we show the mathematical formulation.

Let us assume that we have a regular grid of finest resolution with non-overlapping cells which are defined with a constant functions $r_j(s), j = \{1, 2, \cdots, N_r\}$ such that:

$$r_j(s) = \begin{cases} R_j^{-\frac{1}{2}} & \text{s inside cell j with volume } R_j \\ 0 & \text{otherwise} \end{cases}$$

The r_j cells which are made of fine resolution become the building blocks in the construction of irregular cell. We wish to construct the irregular cells $c_k(s), k = \{1, \cdots, N_c\}$ with $N_c \ll N_r$. The only restriction we have is the irregular constructed cells are also non overlapping. The relation between the c_k and the r_j is given by

$$c_k(s) = \sum_{j=1}^{N_b} \Delta_{kj} r_j(s), \tag{6}$$

$\Delta_{kj} = 1$ when r_s is a building block of cell c_k and 0 otherwise.

Calculating Δ which forms the building blocks for merging the finest resolution grid to form irregular mesh is computationally expensive. In our method these are calculated using quadtree/octree based technique and performed simultaneously on all nodes as it requires only the local ray information. The choice of quadtree or octree based refinement to compute Δ was mainly for three reason 1) computationally less intensive 2) can be stored efficiently on devices with low-memory 3) forms structured cell partitioning. The structured cell partitioning plays a key role during distributed computing where the solution from different grids needs to be merged. In section 5 we show how this will simplify the problem and reduce the communication overhead. Next, we will explain the adaptive mesh refinement method using quadtree on 2D tomography problem. The method can be easily extended to 3D case using octree.

4.2 Adaptive mesh using quadtree/octree

To perform quadtree based AMR on seismic tomography first we need to generate a density matrix I which is checked for homogeneity criteria and later split. Matrix I should be in $\mathbb{R}^{n \times n}$ where $n \in 2^k, k > 0$ and this restricts us to have the finest resolution to be power of 2. We use hit count to generate I i.e. I_{ij} = number of rays passing through ij-th grid of finest resolution. There are other ways to choose I such as region of interest and other hybrid methods (Spakman and Bijwaard, 2001) and this requires domain knowledge and will not be discussed in this paper. Now with only hit count, I can be generated by performing columnSum(A) and then reshaping it to $n \times n$. To generate quadtree S we use QTDecomp function which first initializes S to size of I which becomes the root. Later this S is divided into four equal sized squares if the corresponding blocks in I satisfy the criterion of homogeneity. This process is further continued recursively and stops if either the block does not meet the criteria or depth of the tree is k i.e. until finest resolution. After obtaining the quadtree S we can easily generate Δ by using algorithm [1]. In this paper, we do not discuss the efficient way to implement quadtree data structures and further details on this can be found in (Gargantini, 1982).

Algorithm 1 $\Delta \leftarrow$ TransMat(S)

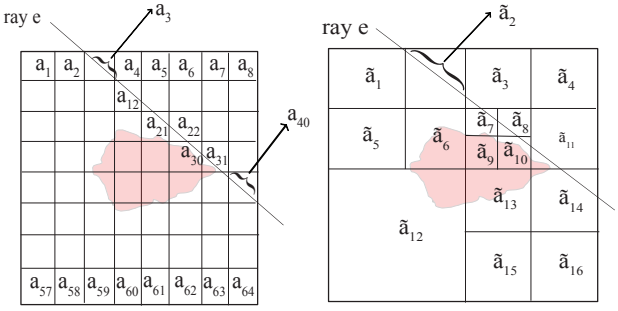
```
for i ← root until all nodes do
  if(i ==leafNode(S))
    idx = getIndex(i)
    Δ(i,idx) = 1
  endif
endfor
```

Next we will show how we can apply AMR to our problem and derive equations that satisfy the travel time setup explained in section 3. Let $\tilde{A}$ denote the data kernel formed by irregular mesh and the coefficients in $\tilde{A}_{ek}$ denotes the length of e-th ray in k cell. Computing $\tilde{A}_{ek}$ solely by using ray tracing can be computationally expensive especially because of the irregular geometry of the mesh. Therefore, to avoid that we use (6) to obtain the relationship between ray lengths on irregular and regular grids, and it is given by,

$$\tilde{A}_{ek} = \sum_{j=1}^{N} \Delta_{kj} A_{ej} \tag{7}$$

Figure 4 shows the relation between rays on two grids that is given by (7). In this, $\tilde{a}_2$ is obtained by summing $\{a_3, a_4, a_{11}, a_{12}\}$ whereas, $\tilde{a}_7$ to $\tilde{a}_{10}$ maintains the finest resolution as regular grid. This criteria of splitting is decided by Δ as discussed earlier and now the equation can be re-written in matrix form as,

$$\tilde{A} = A\Delta^T \tag{8}$$



(a) Regular Grid (b) Adaptive Grid

Figure 4 Relation between ray tracing in different grids

Substituting (7) and (8) in (4) equation at each node becomes,

$$B_i = A_i \Delta_i^T y$$
$$B_i = \tilde{A}_i y$$

where, B_i and Δ_i is the travel time vector and transformation matrix formed at node i respectively. y is the new model vector on irregular grid and can be transformed to original grid by $x = \Delta_i y$

We summarize the entire process of adaptive mesh refinement using quadtree in Figure 5.

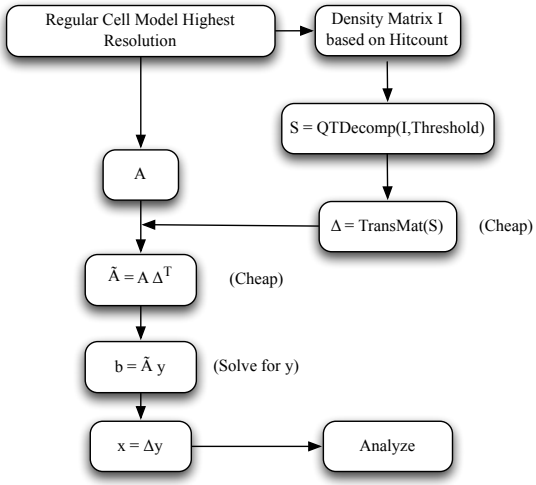
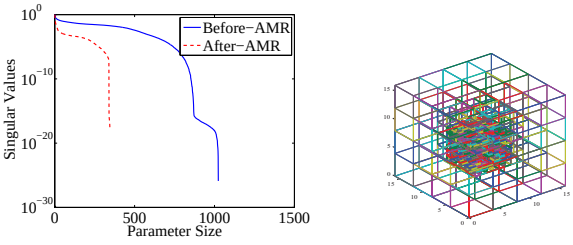


Figure 5 Flowchart of the mesh refinement process



(a) Singular values of A and $\tilde{A}$ (b) Octree based AMR

Figure 6 Effect of AMR on singular values and Octree based AMR

We applied adaptive mesh refinement on seismic tomography problem and we observed that it can improve the condition number of matrix A by removing the small singular values as shown in Figure 6(a). Figure 6(b) shows the adaptive mesh refinement on 3D problem using octree decomposition and have finer grids at the center of magma and becomes courser towards outside.

Remark 2: *Adaptive mesh refinement can be viewed as a non-trivial way of adaptive pre-conditioning as it decreases the condition number of the data kernel.*

The technique of selecting pre-conditioner using geometry of the problem and simple parametrization technique (quadtree/octree) is computationally less intensive. However, it should be noted that AMR relies on the threshold we choose and this requires domain knowledge. Also, if thresholds are not selected carefully it may result in removal of good singular values thereby leading us to different or bad solution. We will show the analysis of threshold sensitivity in the evaluation section. Until now we have seen the working of AMR for seismic tomography and we have shown how quadtree/octree based approach is suitable to run on sensor nodes. In the next section we will discuss how these transformed system of linear equation can be solved distributedly over a sensor network.

5 Distributed Tomography Inversion using Adaptive Mesh

In the previous section we have seen how to transform the system of linear equation formed over regular grid to a more well-condition system using irregular partitioning. Now applying (8) and (4.2) to (5) we get,

$$\mathbb{A}\Delta^T y = \mathbb{B}$$
$$\tilde{\mathbb{A}} y = \mathbb{B}$$

where,

$$\tilde{\mathbb{A}} = \begin{pmatrix} A_1\Delta_1^T \\ A_2\Delta_2^T \\ \vdots \\ A_p\Delta_p^T \end{pmatrix} = \begin{pmatrix} \tilde{A}_1 \\ \tilde{A}_2 \\ \vdots \\ \tilde{A}_p \end{pmatrix}; \mathbb{B} = \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_p \end{pmatrix}; \tilde{A}_i \in \mathbb{R}^{m_i \times n_i}$$

From the above equation we see that at each sensor $i \in \{1, \cdots, P\}$ $A_i \in \mathrm{R}^{m_i \times n}$ gets transformed to $\tilde{A}_i \in \mathrm{R}^{m_i \times n_i}$. Note that during this transformation the number of rays i.e. m_i and right hand side B_i are unchanged while the number of grids are changed from n to n_i where $n_i \leq n$. At each sensor station the new linear subsystem formed after AMR is given by $\tilde{A}_i y_i = B_i$. The rows of this local subsystem contains rays and can be solved locally using row projection method like ART algorithm (2) due to the advantage it offers as mentioned in section 2.

The asymptotic convergence of this method requires $0 < \lambda < 2$, where λ is called relaxation parameter. Recently the

Algorithm 2 $x^k \leftarrow$ART(λ, A, b, x^{k-1})

1: $x^{k,0} = x^{k-1}$,
2: $x^{k,i} = x^{k,i-1} + \lambda \frac{(b_i - \langle a_i^T x^{k,i-1} \rangle)}{\|a_i\|_2^2} a_i, \quad i = \{1, \cdots, m\}$
3: $x^k = x^{k,m}$

rate of convergence of ART was shown to be governed by cond(A) (Strohmer and Vershynin, 2009) where cond(A) = $\kappa(A) = \frac{\sigma_{max}(A)}{\sigma_{min}(A)}$ denotes the condition number of A where σ_{max} and $\sigma_{min}$are its maximum and the minimum singular values. The partial solution obtained from all the nodes using algorithm (2) has to be combined to form next iterate, but before we go into that detail we will show how the transformed system at each node can obtain partial solution with fewer steps than the original system.

Theorem 1: *Let $\tilde{x} = \Delta y$ be the solution to equation $\tilde{A}y = b$, where $\tilde{A} = A\Delta^T$ Then algorithm (2) has $E\|\tilde{x}_k - x_\star\|_2^2 \leq E\|x_k - x_\star\|_2^2$ where, $x_\star$ is the true solution of original system $Ax = b$*

Proof 1: *Let the system of linear equation on regular grid arising due to tomography be $Ax = b$. The transformed system on irregular grid be $\tilde{A}y = b$ where $\tilde{A} = A\Delta^T; \tilde{x} = \Delta y$ From (Strohmer and Vershynin, 2009) the convergence rate of ART on regular grid is given by,*

$$E\|x_k - x_\star\|_2^2 \leq \left[1 - \frac{1}{\kappa(A)}\right]^k \|x_0 - x_\star\|_2^2 \tag{9}$$

similarly for irregular grid we have,

$$E\|\tilde{x}_k - x_\star\|_2^2 \leq \left[1 - \frac{1}{\kappa(\tilde{A})}\right]^k \|x_0 - x_\star\|_2^2 \tag{10}$$

We have shown that AMR reduces the null space of the system i.e $0 \leq \sigma_{min}(A) \ll \sigma_{min}(\tilde{A})$
Since $\kappa(A) = \frac{\sigma_{max}(A)}{\sigma_{min}(A)}$ and $\sigma_{max}(A) \approx \sigma_{max}(\tilde{A})$
We have, $\kappa(\tilde{A}) \ll \kappa(A)$, i.e the system becomes well-conditioned after AMR

Now we have $\left[1 - \frac{1}{\kappa(\tilde{A})}\right]^k \ll \left[1 - \frac{1}{\kappa(A)}\right]^k$
Multiplying both sides by, $\|x_0 - x_\star\|_2^2$ we get,

$$\left[1 - \frac{1}{\kappa(\tilde{A})}\right]^k \|x_0 - x_\star\|_2^2 \ll \left[1 - \frac{1}{\kappa(A)}\right]^k \|x_0 - x_\star\|_2^2 \tag{11}$$

Using (11) in (9) and (10) we get,

$$E\|\tilde{x}_k - x_\star\|_2^2 \leq E\|x_k - x_\star\|_2^2 \tag{12}$$

Now after taking fewer iteration to solve $\tilde{A}_i y_i = B_i$ we will show how to combine the partial solution to form the next iterate. We borrow the idea from (Gordon, 2006) who proposed parallel-ART which is an extension of Cimmino's method to a generalized case. (Gordon and Gordon, 2005) and (Gordon, 2006) proved that the intermediate projection $\{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_P\}$ can be averaged to obtain next iterate x^{k+1}. Algorithm (3) describes the parallel-ART.

The line 4 in algorithm (3) performs averaging of all the partial solution to form the next iterate x^{k+1} which will be

Algorithm 3 Parallel-ART

INPUT
1: $\mathbb{A} = \{A_1, \cdots, A_P\}$; $\mathbb{B} = \{B_1, \cdots, B_P\}$
OUTPUT
1: $x \leftarrow \min_x \|\mathbb{A}x - \mathbb{B}\|$
REPEAT
1: **for**$k \leftarrow 0$ until convergence or maximum number of iteration
2: **do parallel in all blocks** $i \in \{1, \cdots, P\}$
3: $\bar{x}_i^k$ = ART(λ, A_i, B_i, x^k)
4: $x^{k+1} = \frac{1}{P}\sum_{i=1}^P \bar{x}_i^k$
5: **end**

used as a initial input to ART in the next iteration. This process continue until convergence is met or for fixed maximum number of iterations. The partial solution in (Gordon, 2006; Gordon and Gordon, 2005) have same size i.e. $x_i \in \mathrm{R}^n$. However, in our case due to AMR the partial solution $\tilde{x}_i \in \mathbb{R}^{n_i}$ and we show how we can perform averaging over partial solution of different dimension using generalized averaging lemma .

Lemma 1: *(Generalized Averaging Lemma)*
Let $\tilde{x}_i \in \mathrm{R}^{n_i}$ be the projection obtaining after performing ART($\lambda, \tilde{A}_i, B_i, \tilde{x}^{k-1}$) on ith sensor node with irregular grid. The next iterate takes the form $\tilde{x}^{k+1} \leftarrow (D \cdot \tilde{X}^k)/P$ where, $D = (\Delta_1, \cdots, \Delta_P)$ and $\tilde{X}^k = (\tilde{x}_1^k, \cdots, \tilde{x}_P^k)^T$

Proof 2: *Performing ART simultaneously on all the P nodes having irregular mesh gives*
$\tilde{x}_i^k \leftarrow$ *ART*$(\lambda, \tilde{A}_i, B_i, \tilde{x}^{k-1})$, $A_i \in \mathbb{R}^{m_i \times n_i}, B_i \in \mathbb{R}^{m_i}, \tilde{x}_i^k \in \mathbb{R}^{n_i}$
From (4.2) we have, $\Delta_i : \mathbb{R}^{n_i} \to \mathbb{R}^n$ where, n is the dimension of the regular mesh. Therefore, to form the generalized average we need
$\tilde{x}^{k+1} = ((\Delta_1 \tilde{x}_1^k), \cdots, (\Delta_P \tilde{x}_P^k))/P$
$\tilde{x}^{k+1} = ((\Delta_1, \cdots, \Delta_P)(\tilde{x}_1^k, \cdots, \tilde{x}_P^k)^T)/P$
$\tilde{x}^{k+1} = (D \cdot \tilde{X}^k)$

The complete DT-AMR is given in algorithm (4). In this algorithm the initialization, ray tracing and adaptive mesh is performed only once and done simultaneously on each node. The steps that involves communication is highlighted in **bold**. Sending Δ_i to SINK in step 4 is done only once and can be done cheaply as Δ_i can be encoded efficiently as mentioned in section 4. The actual communication in the network occurs in the line 5 and 7 of distributed tomography which involves aggregation of partial solution of size n_i from all the nodes and then broadcast averaged result back. The worst case communication cost for this given by $P\sum_i dim(x_i)$. Since this needs to be broadcasted back and algorithm converges after k iterations, the worst case communication cost is $2kP\sum_i dim(x_i)$. Although this cost may seem to be high, we will see in the next section that it is infact lesser than centralized methods where each row of $\{A_i, B_i\}$ needs to transferred. Moreover, the averaging over the networks can be done by communicating only with the neighbors and we leave this to the future work.

Algorithm 4 Distributed Tomography using Adaptive Mesh Refinement (DT-AMR)

Initialize

1: Number of seismic sensors P and AMR threshold T
2: Finest resolution dimension $Q = d \times d$ or $Q = d \times d \times d$
3: Initialize a $SINK$ node for aggregation.

Ray Tracing at each node i

1: Upon the detection of events
2: Perform ray tracing on each node simultaneously to obtain A_i and B_i

Adaptive Mesh at each node i

1: Obtain Density matrix I from A_i
2: S = QTDecomp(I,T) ; Δ_i = TransMat(S)
3: $\tilde{A}_i = A_i \Delta_i^T$
4: **Send(** Δ_i**)** $\rightarrow$ **SINK**

Distributed Tomography

1: $k \leftarrow 0, x^k \leftarrow 0$
2: **while** not converged **do**
3: In Every node i for $1 \leq i \leq P$ do in parallel
4: $\bar{x}_i \leftarrow$ ART($\lambda, \tilde{A}_i, B_i, x^k$,Iteration)
5: **Send(** $\bar{x}_i$**)** $\rightarrow$ **SINK to average**
6: $x^{(k+1)} = \{ \frac{1}{P} \sum_{i=1}^{P} \Delta_i \bar{x}_i$
7: **Broadcast** $x^{(k+1)}$ **to all the node** P
8: $k \leftarrow k + 1$
9: **end while**
10: Update slowness model: $x = x^{k-1}$
11: TERMINATE

6 Communication Architecture

In this section, we present the communication architecture to evaluate DT-AMR algorithm on large scale sensor network. The protocol is tested on CORE[2] and EMANE[3] network emulators Ahrenholz et al. (2011). The advantage of emulation on CORE is that the code developed here can be transplanted to a Linux-based device, e.g., BeagleBone Black board, virtually without any modifications.

DT-AMR is designed to compute the tomography in a wireless mesh network and requires both unicast and broadcast communication according to the system architecture and the algorithm requirements. On most remote deployment sites, it is hard to rely on the pre-existing infrastructures (e.g. cellular infrastructure). Therefore, we need to utilize the wireless mesh networking which creates its own infrastructure by multi hop relays. However, such systems may experience erratic link qualities and intermittent disconnections among nodes. These characteristics, combined with unpredictable environmental conditions, make it difficult to maintain efficient and reliable end-to-end connectivity that spans many hops. For example, the traditional end-to-end protocol like TCP is not suitable for a wireless mesh network in challenging environment because the packet lost ratio is higher than a wired network. For example in a multi hop transmission using TCP, the source node needs to retransmit the packets through all hops if the packet gets lost on the path. This decreases the data rate after several hops due to packet loss and congestion control.

To address the challenges in wireless mesh networking, we adopt Disruption-Toleration Networks (DTN) techniques to maintain efficient and reliable end-to-end connectivity that

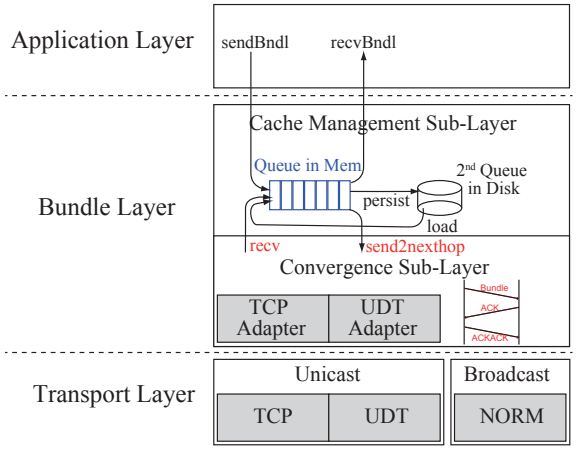


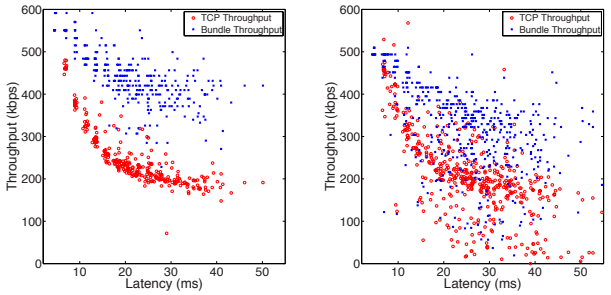
Figure 7 Bundle Layer Architecture

spans many hops for data delivery. In our design, the data is buffered in a bundle and then transferred hop by hop in a store-and-forward manner until it arrives at the destination. Our implementation of DTN technique does not make any changes to underlying network services, it uses TCP for one-hop reliable bundle transfer, and uses routing table to indicate the next hop. Figure 7 shows the application interfaces on each node for the integration of DTN and routing protocol. For DT-AMR emulation, we implemented a naive data aggregation and dissemination protocol on top of bundle layer. In this setup, we specified a SINK node to which all the other nodes sent its partial solution. Next, the SINK node computed the $(k+1)$-th iterate and disseminated the solution back to all nodes.

Figure 8 shows that the Bundle Layer outperforms TCP with routing protocol. The test is done using CORE and EMANE for 100 nodes multi-hop network settings. Besides unicast, we implemented a delay-tolerant broadcasting service based on the NACK-Oriented Reliable Multicast (NORM) protocol. Using NORM interface, one node can push a bundle reliably to its one-hop neighbors. Our cache component can receive and store this broadcast bundle and rebroadcast it again with NORM to the nodes that are two hops away and so on so forth. A redundancy check module is developed in the cache component which guarantees that each node receives the same bundle at most once. The implementation of all the algorithms of CA-DMET is in ANSI C. The event location and tomography inversion related code are cross-compiled to run on BeagleBone Black board. All other code can be directly ported to embedded system such as ARM-based CPU or MCU.

7 Evaluation and Validation

In this section, we evaluate the DT-AMT algorithm and present the simulation results. First we give the description of the synthetic model used for the simulation along with the experimental setup and later we compare DT-AMR with the existing distributed algorithms such as, CARP (Gordon and Gordon, 2005) and CAV (Censor et al., 2001). Typically, to test tomography inversion algorithm a synthetic model is used. This serves two purpose: a) the real data set such as from Mt. St Helens do not have a ground truth and it is still uncertain which

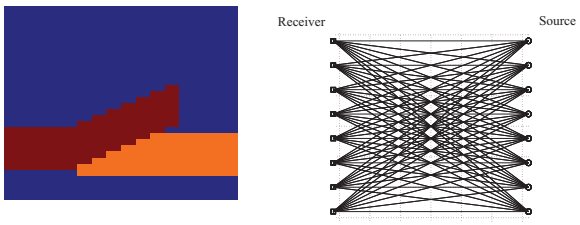


(a) B.A.T.M.A.N. routing (b) OLSR routing

Figure 8 Performance of Bundle layer vs TCP.

model is reliable. b) The simulation using synthetic model enables us to investigate individually various phenomena which cannot be separated physically. For example, p-wave data always contain noise due to measurement and scattering, but simulation can indicate the specific effect separately. For this reason, we adopt a synthetic data of a fault model from (Hansen and Saxild-Hansen, 2012) which has been widely used for cross bore-hole tomography (Curtis and Snieder, 1997). This fault model is created with velocities of $0.75V$ for the right fault and $1.0V$ for the left fault as shown in the Figure 9(a).

We evaluated the communication cost of DT-AMR algorithm using CORE network emulator (Ahrenholz et al., 2011). We select CORE as the development and evaluation platform because the sensors that will be deployed on the real volcano will be some-low powered linux based devices such as android, beagle-bone or raspberry pi. Code developed in CORE emulator can be transferred to these devices without any modification. A network of 32 nodes are deployed which detects the event and traces the ray as shown in the Figure 9(b). We add Gaussian noise to the obtained travel time to model the receiver error. The finest resolution of dimension 32×32 is used as a regular grid to construct adaptive mesh using quadtree. The threshold for the hitcount is chosen to be 20. For the iterative methods, the selection of relaxation parameter ρ are critical and in all of our experiments this remains constant throughout the iterations, i.e., $\rho^k = \rho = 0.25$ for all $k \geq 0$. Rate of convergence of different algorithms are compared using relative updates, $\phi = |x^{(k+1)} - x^{(k)}|/|x^{(k)}|$, residuals $\chi = \|Ax^k - b\|$ and absolute error $\epsilon = \|x^* - x^k\|$ where x^* is the ground truth.

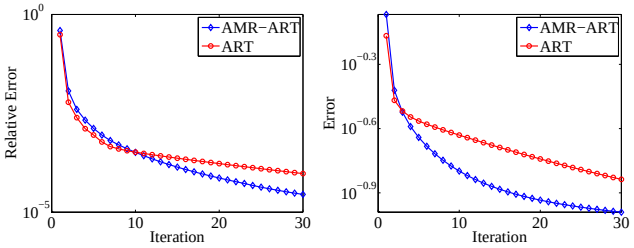


(a) Magma/Fault Model (b) Ray Traversal

Figure 9 Synthetic Model

7.1 Centralized AMR

Adaptive mesh has been applied on seismic tomography in a centralized setup earlier and has proven to perform better (Vesnaver, 1996)(Spakman and Bijwaard, 2001). However, quadtree based adaptive mesh for seismic tomography has been applied for the first time and we validate our approach using similar steps. Quadtree based AMR has been developed specifically to work in distributed environment and we do not expect it to perform better than centralized algorithms mentioned in (Vesnaver, 1996)(Spakman and Bijwaard, 2001). We perform quadtree based AMR on synthetic model and run ART which is a common centralized algorithm used for computing tomography. We see the advantage of AMR in Figure 10(a) and (b), where the relative and absolute error decreases significantly when AMR is used. AMR makes the system well conditioned i.e reduce the zero eigenvalue and when the system is well-conditioned the solution obtained will be closer to the ground truth. From these tests we can conclude that ART with AMR is better and we can obtain better solution.



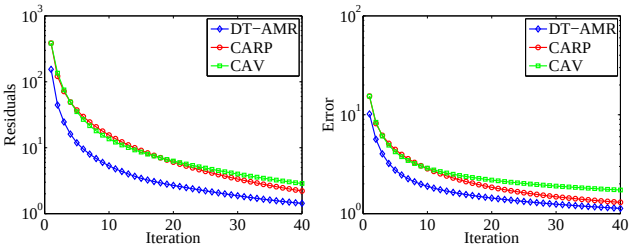
(a) Relative Error in ART (b) Absolute Error in ART

Figure 10 Comparing effect of AMR in Centralized ART

7.2 Distributed AMR

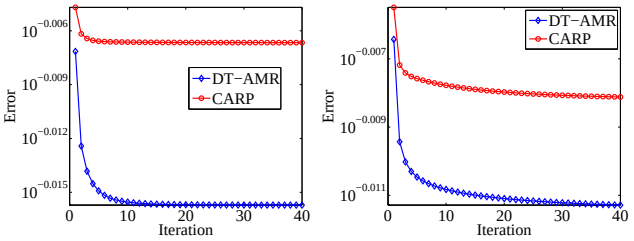
After validating the performance of our quadtree based AMR in centralized setup, we now perform series of experiment on distributed network. We compare our DT-AMR with standard distributed algorithms such as CARP and CAV. We use residuals and absolute error as the parameter for comparison and results are shown in Figure 11. These plots demonstrate that there is a difference in the initial convergence behavior in these algorithms both in-terms of residuals and absolute error. The final solution obtained from DT-AMR is also more closer to the ground truth and can be obtained with fewer iteration. The iteration on the x-axis represents the total number exchange of partial solution required during the intermediate step.

DT-AMR takes the advantage of partitioning the system of equation based on its resolving power at each node making it more well conditioned. The local computation on the well conditioned system on single node will accelerate the convergence and this is shown in Figure 12, where we analyze the performance on node 1 and 4. We observe that even at each node the partial solution obtained from DT-AMR is significantly better than CARP (Theorem 1) and this is the main reason for the improved performance of DT-AMR in distributed environment.



(a) Residuals (b) Absolute Error

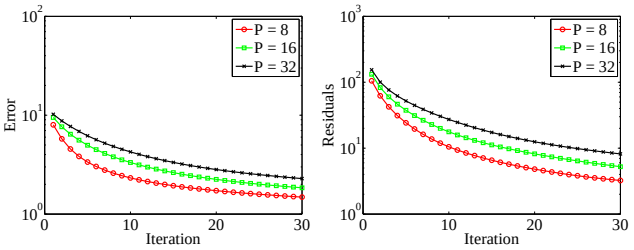
Figure 11 Comparing effect of DT-AMR with CARP and CAV



(a) Absolute Error in Node 1 (b) Absolute Error in Node 4

Figure 12 Comparing effect of DT-AMR in single Node

The DT-AMR is designed to run both in clustered and also in fully distributed setup. To test this, we group the nodes into clusters each having cluster heads which collects A_i and b_i from its neighbors and run DT-AMR. The simulation was done with total 32 nodes and number of clusters varying from 8 to 32 (each cluster has one node i.e., fully distributed). From Figure 13 we observe that as cluster number increases the convergence rate decreases. This is one of the drawbacks of distributed iterative methods and the reason behind this is the way these algorithm projects and combine the partial results. Centralized methods are always better in-terms of computation and final solution especially for inverse problems.



(a) Absolute Error in DT-AMR (b) Residual in DT-AMR

Figure 13 DT-AMR with different partitioning

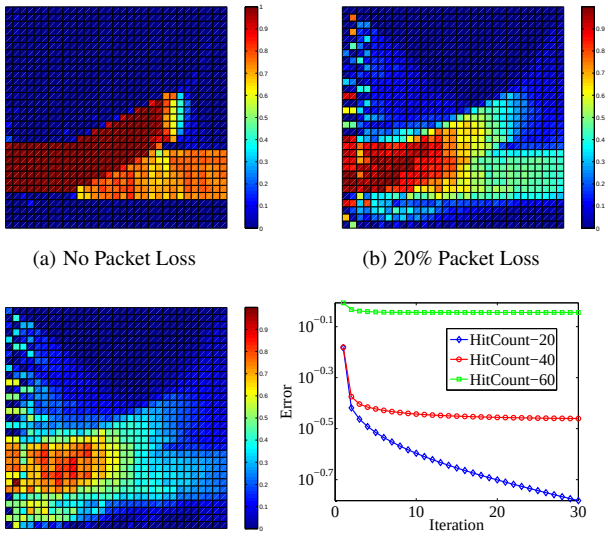
7.3 Communication Cost and Robustness

In section 5 we have seen the summation of partial solution on the network and dissemination of the computed result constitutes the major part of communication in DT-AMR. In this setup, we specified a SINK node to which all the other nodes send its partial solution. Next, the SINK node computed the $(k+1)$-th iterate and disseminated the solution back to all nodes. For fair comparison we implemented similar data aggregation protocol (without the dissemination part) for centralized scheme where all nodes sent its ray information to the SINK placed 1) at the corner CENT(C) and 2) in the middle CENT(M).

To compare the communication cost we increased the number of receivers from 32 to 49 which formed 7×7 grid of mesh network. From Figure 14(a) and (b), we can see that communication cost in a centralized setup is high near the base station as all the ray information is transferred over the network and its volume depend on the number of earthquake events.

In case of DT-AMR we used data aggregation and dissemination as mentioned above with aggregation SINK node in the middle. Figure 14(c) shows the communication pattern and from this we can see that the communication cost for DT-AMR is less than centralized scheme. This is mainly because distributed method depends on number of iteration and typically with semi-convergent property of iterative method the number of iteration is much less compared to number of earthquake events. The advantage of DT-AMR method is that the communication depends only on the averaging over the network and this can be done using gossip methods which involves information only with neighbors and can provide flat balanced communication pattern. This is beyond the scope of this paper and will be investigated in future. AMR decreased the size of partial solution $\tilde{x}_i$ from $n \to n_i$. We validate this effect by comparing volume(bytes) transferred by DT-AMR to that of without AMR. Figure 14(d) shows that in both cases SINK(M) and SINK(C) the volume of bytes transferred is less in case of DT-AMR.

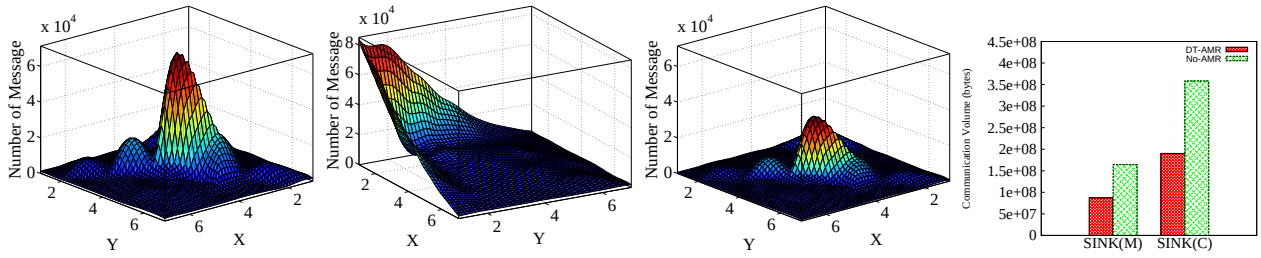
Next we performed experiments to validate the robustness of DT-AMR with same data set but with two different packet loss ratios of 20% and 40% at each node. Figure 15 shows the final tomography obtained and from this we can see that due to the lost partial solution at each iteration the solution is not very accurate compared to one with no lost packet. However, with Figure 15(b) and (c) we are still able to identify the region of fault although it is not very accurate.



(a) No Packet Loss (b) 20% Packet Loss

(c) 40% Packet Loss (d) DT-AMR with varying hit-count

Figure 15 Loss tolerance and threshold sensitivity of DT-AMR



(a) Centralize(M) (b) Centralize(C) (c) DT-AMR(M) (d) Communication Volume

Figure 14 Communication Cost

Finally, we evaluate the performance of DT-AMR by varying the threshold of the hit-count from 20 through 60. From Figure 15(d) we can see that the convergence speed and the final result depends on the selection of hit-count threshold as mentioned in section 4. To select this threshold we need domain knowledge and if not selected carefully may result in removal of good singular values that leads to different or bad solution.

8 Conclusion

In this paper, we presented a distributed approach based on adaptive mesh refinement that performed tomography inversion over sensor networks. This algorithm first converted the problem to a well-conditioned system by eliminating the null spaces and retaining the information based on the nodes resolving power. Later, a distributed method was presented which solved the inverse problem over sensor networks by exchanging on the partial solution to form next iterate. The experimental evaluation showed that our proposed method converges faster than CARP and CAV also obtained better slowness closer to ground truth. Centralized methods are more computationally stable and distributed method cannot perform better in terms of getting solution, however in terms of communication and message exchanged DT-AMR algorithm performed better than centralized algorithm. DT-AMR is designed to run on devices with low memory and computational power. Further enhancement of this algorithm can be done by studying its performance on real data and applying different constraints to form adaptive mesh. Our future research will focus on developing a distributed averaging operator that can balance the communication cost.

Acknowledgements

Our research is partially supported by NSF-CNS-1066391, NSF-CNS-0914371, NSF-CPS-1135814 and NSF-CDI-1125165.

References

Ahrenholz, J., Goff, T., and Adamson, B. (2011). Integration of the CORE and EMANE Network Emulators. In *MILITARY COMMUNICATIONS CONFERENCE, 2011 - MILCOM 2011*, pages 1870–1875.

Berger, M. J. and Oliger, J. (1984). Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics*, 53(3):484–512.

Bertero, M., Mol, C. D., and Pike, E. R. (1985). Linear inverse problems with discrete data. I. General formulation and singular system analysis. 1(4):301–330.

Bertsekas, D. P. and Tsitsiklis, J. N. (1991). Some aspects of parallel and distributed iterative algorithms âŁ" a survey. *Automatica*, 27(1):3–21.

Censor, Y., Gordon, D., and Gordon, R. (2001). Component averaging: An efficient iterative parallel algorithm for large and sparse unstructured problems. *Parallel Computing*, 27(6):777–808.

Curtis, A. and Snieder, R. (1997). Reconditioning inverse problems using the genetic algorithm and revised parameterization. *Geophysics*, 62(5):1524–1532.

Elble, J. M., Sahinidis, N. V., and Vouzis, P. (2010). GPU computing with Kaczmarz's and other iterative algorithms for linear systems. *Parallel Computing*, 36:215–231.

Gargantini, I. (1982). An Effective Way to Represent Quadtrees. *Commun. ACM*, 25(12):905–910.

Gordon, D. (2006). Parallel ART for image reconstruction in CT using processor arrays. *International Journal of Parallel, Emergent and Distributed Systems*, 21(5):365–380.

Gordon, D. and Gordon, R. (2005). Component-averaged row projections: a robust, block-parallel scheme for sparse linear systems. *SIAM Journal on Scientific Computing*, 27:1092–1117.

Hansen, P. C. and Saxild-Hansen, M. (2012). AIR Tools âŁ" A MATLAB package of algebraic iterative reconstruction methods. *Journal of Computational and Applied Mathematics*, 236(8):2167–2178.

Heath, M. T., Ng, E., and Peyton, B. W. (1991). Parallel algorithms for sparse linear systems. *SIAM review*, 33(3):420–460.

Herman, G. T. (1980). *Reconstruction from Projections: The Fundamentals of Computerized Tomography*. Academic Press.

Iyer, H. M. and Dawson, P. B. (1993). *Imaging volcanoes using teleseismic tomography*. Chapman and Hall.

Kaczmarz, S. (1937). Angenäherte Auflösung von Systemen linearer Gleichungen. *Bulletin International de l'Académie Polonaise des Sciences et des Lettres*, 35:355–357.

Kamath, G., Shi, L., and Song, W.-Z. (2013). Component-Average based Distributed Seismic Tomography in Sensor Networks. In *IEEE DCOSS*.

Lees, J. M. (1992). The magma system of Mount St. Helens: non-linear high-resolution P-wave tomography. *Journal of Volcanology and Geothermal Research*, 53:103–116.

Liu, G., Tan, R., Zhou, R., Xing, G., Song, W., and Lees, J. (2013). Volcanic Earthquake Timing using Wireless Sensor Networks. pages 91–102.

Michelini, A. (1995). An adaptive-grid formalism for traveltime tomography. *Geophysical Journal International*, 121(2):489–510.

Moran, S. C., Lees, J. M., and Malone, S. D. (1999). P wave crustal velocity structure in the greater Mount Rainier area from local earthquake tomography. *Journal of Geophysical Research*, 104(B5):10775–10786.

Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, second edition.

Song, W.-Z., Huang, R., Xu, M., Ma, A., Shirazi, B., and Lahusen, R. (2009). Air-dropped Sensor Network for Real-time High-fidelity Volcano Monitoring. In *The 7th Annual International Conference on Mobile Systems, Applications and Services (MobiSys)*.

Spakman, W. and Bijwaard, H. (2001). Optimization of Cell Parameterizations for Tomographic Inverse Problems. *Pure and Applied Geophysics*, 158(8):1401+.

Strohmer, T. and Vershynin, R. (2009). A randomized Kaczmarz algorithm with exponential convergence. *J. Fourier Anal. Appl*, 15:262–278.

Vesnaver, A. L. (1996). Irregular grids in seismic tomography and minimum-time ray tracing. *Geophysical Journal International*, 126(1):147–165.