J. Parallel Distrib. Comput. 89 (2016) 50–64

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

Distributed travel-time seismic tomography in large-scale sensor networks

Goutham Kamath [a,*], Lei Shi [a], Wen-Zhan Song [a], Jonathan Lees [b]

[a] Department of Computer Science, Georgia State University, United States
[b] Department of Geological Sciences, University of North Carolina at Chapel Hill, United States

HIGHLIGHTS

- We present a distributed method to obtain seismic image on wireless sensor networks.
- We perform in-network computing and avoid data gathering.
- High resolution image can be obtained using proposed multi-resolution scheme.
- We evaluate the method using synthetic seismic data in a CORE emulator.
- Our method is able to recover velocity model of Mt. St. Helens.

ARTICLE INFO

Article history:
Received 20 July 2014
Received in revised form
19 November 2015
Accepted 4 December 2015
Available online 15 December 2015

Keywords:
Distributed computing
Seismic tomography
Sensor network
Mt. St. Helens

ABSTRACT

Current geophysical techniques for visualizing seismic activity employ image reconstruction methods that rely on a centralized approach for processing the raw data captured by seismic sensors. The data is either gathered manually, or relayed by expensive broadband stations, and then processed at a base station. This approach is time-consuming (weeks to months) and hazardous as the task involves manual data gathering in extreme conditions. Also, raw seismic samples are typically in the range of 16–24 bit, sampled at 50–200 Hz and transferring this high fidelity sample from large number of sensors to a centralized station results in a bottleneck due to bandwidth limitations. To avoid these issues, a new distributed method is required which processes raw seismic samples inside each node and obtains a high-resolution seismic tomography in real time. In this paper, we present a component-averaged distributed multi-resolution evolving tomography algorithm for processing data and inverting volcano tomography in the network while avoiding centralized computation and costly data collection. The algorithm is first evaluated for the correctness using a synthetic model in a CORE emulator. Later, our proposed algorithm runs using the real data obtained from Mt. St. Helens, WA, USA. The results validate that our distributed algorithm is able to obtain a satisfactory image similar to centralized computation under constraints of network resources, while distributing the computational burden to sensor nodes.

1. Introduction

Current volcano data collection and monitoring systems lack the capability of obtaining real time information and recovering the physical dynamics of seismic activity with sufficient resolution. At present, the seismic tomography process involves aggregation of 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. This process of 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 constraints, volcanologists worldwide use less than 20 stations on many of the threatening active volcanoes [30], limiting our ability to understand dynamics and physical processes of volcanoes in real-time. The centralized solution also introduces a computational bottleneck and increases the risk of data loss in case of node failures. With the advancement in sensor technology it is now possible to deploy and maintain a large-scale network for environmental monitoring and surveillance. However, currently used

* Corresponding author.
E-mail addresses: gkamath1@student.gsu.edu (G. Kamath), lshi1@student.gsu.edu (L. Shi), wsong@gsu.edu (W.-Z. Song), jonathan.lees@unc.edu (J. Lees).

http://dx.doi.org/10.1016/j.jpdc.2015.12.002
0743-7315/

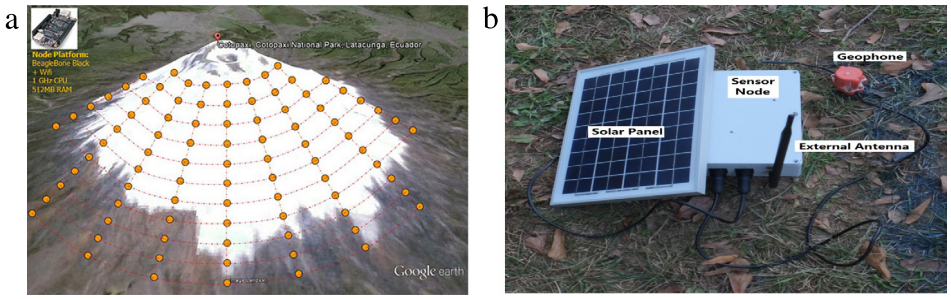


Fig. 1. (a) Illustration of our proposed deployment. (b) Our custom designed station (OpenNode) along with solar panel, geophone and external antenna.

tomography inversion algorithms cannot be easily implemented under this distributed scenario, as they are designed for the centralized processing. Thus, real-time volcano tomography requires a practical approach which is distributed, scalable, and efficient with respect to current tomographic methods.

1.1. State of the art

At present, tomography inversion methods rely on centralized methods where data gathered by instruments is manually collected and transferred to a centralized unit for further processing. This scheme has been implemented on a number of volcanoes including Mount St. Helens [24] and Mount Rainier. Currently, the seismic community faces two major hurdles in order to obtain both real-time and high-resolution tomography. First, the data logged at each station is of high fidelity, and due to limited communication bandwidth, information from only a small number of sensors can be obtained for real time imaging. With limited sensors, high resolution tomographic images cannot be obtained. On the other hand, large arrays of sensors can be deployed to obtain high-resolution tomography as proposed by iMush[1] project. However, limited bandwidth restricts the transfer of large volume of data from these sensors, in which case seismologists resort to manual data gathering which takes months for post processing and imaging. To obtain high resolution and real time imaging we need a large number of sensor stations which have the capability of performing in-network computing and also avoid expensive data collection. The authors in [26] have discussed the use of low-cost sensors for P-wave arrival time picking and earthquake location. 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 have low computational power e.g. Raspberry Pi/Beagle-Bone/Android.

1.2. Contributions and outline

Our project aims at constructing high resolution seismic tomography using large dense array of stations as shown in Fig. 1(a). Fig. 1(b) shows a seismic station that consists of a geophone (vibration recording sensor), MSP430/BeagleBone Black acts as a computational unit and Xbee radio which acts as a low power communication module. The stations are powered by battery and solar panel.

This paper mainly focuses on building a distributed algorithm suitable for performing in-network tomographic inversion. Here we present a Component-Averaged Distributed Multi-resolution Evolving Tomography (CA-DMET) algorithm to compute tomographic inversion in a distributed fashion without sending the raw data to a centralized location. The sensor nodes deployed use the arrival times of seismic events and earthquake locations to derive a 3D tomographic model of the velocity structure within the volcano. As more earthquakes are recorded, the velocity model evolves over time to refine the existing one. To our best knowledge of literature, our work is the first attempt to perform seismic tomography in sensor networks. The algorithm proposed here has application to fields far beyond the specifics of volcanoes, e.g., oil field explorations have similar problems and needs.

The rest of the paper is organized as follows. Section 2 presents the background knowledge of seismic tomography, while in Section 3 we formulate the tomography problem. In Section 4, we discuss the related works of distributed algorithms in wireless sensor networks. The algorithm and design of distributed tomography inversion are presented in Section 5. Synthetic data simulation results are obtained in Section 6. Results from Mt. St. Helens are presented in Section 7. Discussions and future work are presented in Section 8. We then conclude the paper in Section 9.

2. Seismic tomography background

Tomography can be defined as the science of computing reconstructions in 2D and 3D from projections, i.e., solving the system of linear equations obtained by integrations along the rays that penetrate a domain Ω, typically a rectangle in 2D, and a box in 3D. In this paper we use first-arrival travel time of the p-wave (primary wave) to derive the internal velocity structure of a volcano. Fig. 2(a) shows the sample of p-wave obtained at four different stations and the blue line indicates its corresponding arrival time. Next, we explain four basic principles involved in travel-time seismic tomography.

(i) P-wave arrival-time picking: P-waves travel faster than any other waves though the earth and are the first to be recorded in the seismic sensors. Inside the earth, density varies due to the presence of different layers and materials. These cause the seismic waves to travel at different velocities in different directions as shown in Fig. 2(b). By picking the arrival time of the p-wave at different stations we can obtain the difference in the propagation delay. Picking arrival time is inherently distributed and authors in [26] have proposed methods to automatically pick the arrival time which can be implemented on each station.

(ii) Event location: Once the arrival times of p-waves have been detected by each station, their differences can be used to obtain the exact location and the origin time of the earthquake. Geiger's [12] method is used to calculate the event location along with origin time and it requires travel time differences from at least four different stations. This method is one of the classic and widely used event localization schemes to obtain the exact location and time.

(iii) Ray tracing: This is the technique used to find the ray paths between the seismic source locations (earthquake) and the receiver nodes with minimum travel time, following an event.

[1] http://imush.org/.

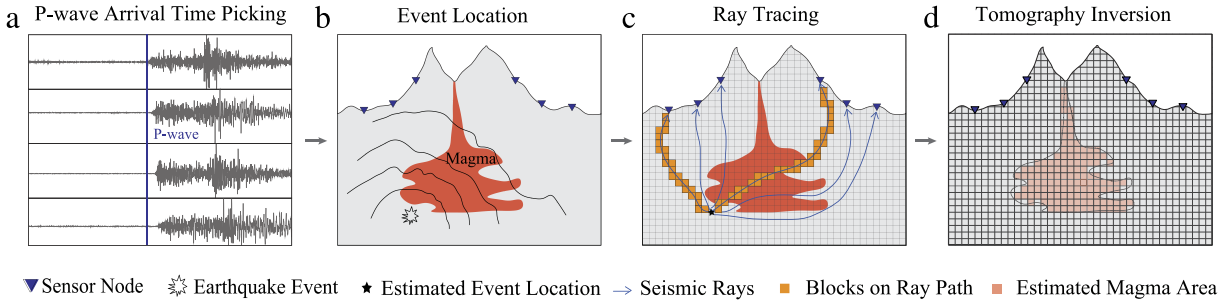


Fig. 2. Procedures of seismic tomography inversion. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Once an earthquake occurs, the seismic rays originating from these events perturb and register the anomalous residuals. Given the source location of the seismic events and current velocity model of the volcano, the ray tracing method finds the ray paths from the event source locations to the nodes, as shown in Fig. 2(c). The p-waves traveling in different mediums under the earth tend to bend due to reflection and refraction resulting in a curved path during the ray tracing [11]. However, for the sake of simplicity in this paper we assume that the rays travel in a straight path.

(iv) Tomography inversion: The traced ray paths are in turn used to estimate the velocity model of the volcano. Inheriting the concept from medical tomography, the volcano can be partitioned into smaller blocks called pixels (2D) and voxels (3D), as shown in Fig. 2(d), to form a large, sparse system of linear equations. In the next section, we show the formulation of the travel-time seismic tomography problem using the perturbation model.

3. Problem formulation

Let us assume that the time taken by seismic signal to travel from the source (earthquake) to a receiver (station), in a given medium, is a function of the seismic velocity of the material and the ray path. Given the velocity model, source and receiver locations, calculating the travel time is called a forward problem, given by

$$T(v, ray) = \int_{ray} \frac{1}{v(y)} d\tau, \tag{1}$$

where, T is the travel time, $v(y)$ represents velocity of the material and spatial position y, and $d\tau$ is a differential line element along the path. Since, T does not depend linearly of the velocity, $v(y)$, for convenience we introduce *slowness* where, $s(y) = 1/v(y)$. Therefore Eq. (1) becomes

$$T(v, ray) = \int_{ray} s(y) d\tau. \tag{2}$$

Since the ray path traveled depends on the slowness, it is non-linear and can be linearized using the perturbation approach [25]. Suppose, we have a reference model $s_0(y)$, the slowness model $s(y)$ we intend to obtain is given by,

$$s(y) = s_0(y) + \delta s(y). \tag{3}$$

Substituting Eq. (3) in Eq. (2), we get,

$$T = \int_{ray} s_0(y) d\tau + \int_{ray} \delta s(y) d\tau, \tag{4}$$

and by rearranging we get,

$$\delta T = T - T_0 \approx \int_{ray} \delta s(y) d\tau, \tag{5}$$

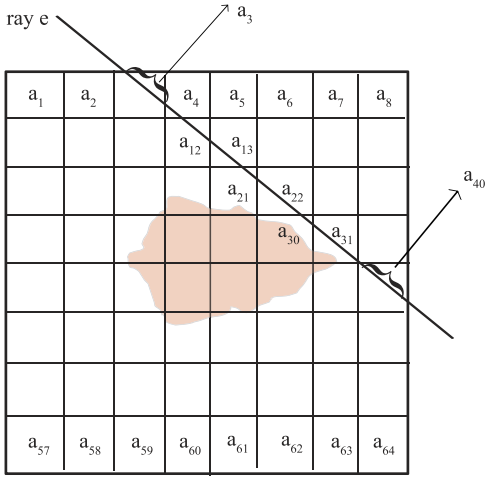


Fig. 3. Example of discretized tomography with $N = 8$ in sensor i. The eth ray intersects a total of 9 pixels, and thus the eth row of matrix A_i has 9 non-zero elements (in columns 3, 4, 12, 13, 21, 22, 30, 31, and 40).

where, δT is called travel time residual. This gives the linear relationship between δT and slowness perturbation δs (see Fig. 3).

Next we will show how the continuous function $\delta s(y)$ can be discretized by dividing a square domain $\Omega = [0, 1] \times [0, 1]$ into a grid of pixels. For simplicity we assume 2D scenario with regular $N \times N$ grids. When Eq. (5) is discretized on $N \times N$ array of pixels, let us assume $s(y)$ takes a constant value $s_{k\ell}$ over the pixels (k, ℓ), where k and $\ell \in \{1, \ldots, N\}$.

Let us consider a single ray $e \in \{1, \ldots, m_i\}$, where m_i denotes total number of rays in sensor node i. Also let the travel time residual for ray e is given by $b_e = \delta T_e$. Now, by substituting the discretized $s_{k\ell}$ into (5) we get,

$$b_e = \sum_{(k,\ell) \in ray_e} s_{k\ell} \Delta L^e_{k\ell}, \tag{6}$$

where, $\Delta L^e_{k\ell}$ = length of ray_e 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. $x_j = s_{k\ell}$ and $a_{ej} = \Delta L^e_{k\ell}$, where $j = (\ell - 1)N + k$, we get

$$b_e = \sum_{j=1}^{n} a_{ej} x_j, \quad e \in \{1, \ldots, m_i\};\ n = N^2. \tag{7}$$

In the matrix form the system of equations formed at the ith sensor node is given by,

$$A_i x = B_i, \quad A_i \in \mathbb{R}^{m_i \times n},\ x \in \mathbb{R}^n \tag{8}$$

where,

$$B_i = [b_1, \ldots, b_{m_i}]^T,$$

$$a_{ej} = \begin{cases} \Delta L^e_{k\ell} & (k, \ell) \in ray_e \\ 0 & \text{otherwise.} \end{cases}$$

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

$$Ax = B \tag{9}$$

where,

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

In this paper, we consider solving Eq. (9) in a distributed way over a loosely connected decentralized network. We assume that each node i has $\{A_i, B_i\}$ obtained from steps such as arrival time picking, event location and ray tracing.

4. Related works

The tomography inversion process mainly involves solving a large sparse system of linear equations (Eq. (9)). To solve such large sparse system, iterative methods become almost mandatory as they are more efficient in-terms of memory and computational requirements compared to direct methods [29]. Several parallel and distributed iterative methods have been developed and are currently used to solve a large variety of problems [16,4]. These methods are developed mainly for GPU computing and assume large bandwidth and reliable communication. Communication in sensor networks are unreliable and has a limited bandwidth. These constraints prevent us from using existing methods on sensor network for tomography inversion process.

Another method originally proposed for parallel computing is the multi-splitting solution for a least squares problem i.e. $\min_x \|Ax - B\|^2$ [28]. This method partitions the system into columns instead of rows letting each processor apply a well-known fixed point iteration methods such as Jacobi, Gauss–Seidel and successive over-relaxation to the normal equation. The solutions to the local problems are recombined using weighted matrices to pick out the appropriate components of each subproblem solution. Column splitting of Eq. (9) in seismic tomography refers to the splitting of the travel time vector B. Since we have the information of the total travel time from event source to node, we cannot divide B exactly as mentioned in [28]. Now splitting the ray using any heuristic method can introduce an error on top of existing measurement noise. Apart from that, the communication cost is very expensive as it requires exchanging B which in our case increases with the occurrence of an earthquake. Due to this reason, column partitioning is not suitable for distributed seismic tomography.

A popular iterative method for solving system of linear equations was proposed by Kaczmarz (KACZ) [20]. This method is also known under the name Algebraic Reconstruction Technique (ART) in computer tomography [17]. 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.

In order to handle large data, many iterative methods suitable for parallel computing have been developed recently [10]. Among them we found Component Averaging (CAV) [7] and Component Averaged Row Projections (CARP) [14] methods to be suitable for distributed tomography computation over sensor networks [21]. CAV [7] is a Cimmino-type method [8] that projects current iterates simultaneously onto all the systems' hyperplanes. In this method, each projection is scaled with a sparsity-related weight, rather than fixed weights used in Cimmino, exhibiting faster numerical convergence. CAV retains the desired convergence properties of the Cimmino's method, in the sense that it converges in the inconsistent case.

In CARP, the equations are divided into blocks, and each block is assigned to one processor. The processors operate in parallel and each processor performs one or more Kaczmarz projections on its assigned equations. The different solutions are then merged to become the next iterate by averaging the values of each component across all the blocks in which the component appears [14]. This process is repeated until convergence. CARP does not place any restrictions on the system matrix or on the selection of the blocks and is shown to be robust and memory efficient [14]. Authors in [10] have compared the performance of various block parallel methods on GPU's.

The proposed CA-DMET is primarily designed to work on loosely coupled systems such as wireless sensor network [21]. In the proposed method we inherit the idea of averaging projections introduced in CARP [14] to make CA-DMET distributed, however it differs with CARP in the following way. (i) Row projection method KACZ is replaced with Bayesian ART (BART) which is found to be suitable for seismic data [25]. (ii) A Multi-level scheme to solve problem in different resolution level depending on the availability of events is presented. (iii) We apply a smoother i.e. a low pass filter after each iteration to obtain satisfactory seismic result. To our best knowledge of literature, our work is the first attempt to compute seismic tomography in sensor networks.

5. Algorithm design

Image reconstruction problems have large sparse linear systems and iterative methods are routinely used to solve them [15]. Algebraic Reconstruction Technique (ART) [17] also known as Kaczmarz (KACZ) [20] was the first iterative method used to solve such problems. This algorithm is inherently sequential, where initial vector x is projected onto rows of A in a cyclic manner. At each iteration, the previous iterate is projected orthogonally onto the hyperplane defined by the equation $\langle a_i, x\rangle = b_i$. Here, we denote a_i and b_i as the ith row vector of the matrix A and ith element of b respectively. The ART algorithm is given by where, ρ_i is a cyclic

Algorithm 1 ART Algorithm

1: Initialize: $x^0 \leftarrow 0$
2: **for** $k \leftarrow 0$ until convergence or max iteration **do**
3: $\quad i \leftarrow k \bmod m + 1$
4: $\quad x^{(k+1)} = x^{(k)} + \rho_i \frac{b_i - \langle a_i, x^{(k)}\rangle}{\|a_i\|^2} a_i$
5: **end**

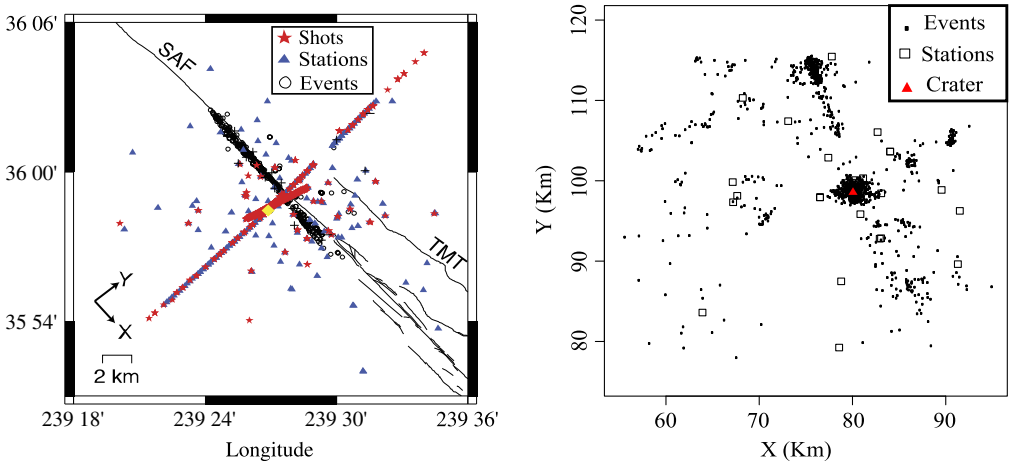
relaxation parameter that extends the projections either in front of the hyperplane ($\rho_i < 1$), on the hyperplane ($\rho_i = 1$), or beyond the hyperplane ($\rho_i > 1$). The convergence of KACZ with relaxation parameter ($0 < \rho < 2$) for a consistent system has been shown in [9,31].

Cimmino in 1938 introduced a parallel iterative method [8] similar to ART. This method is highly parallel and is guaranteed to converge even in an in-consistent case. In practice, the method is slower than ART; however, it has the potential to lower the computation burden by splitting the task onto different processors as shown in Algorithm 2.

Algorithm 2 Cimmino's Algorithm

1: Initialize: $x^0 \leftarrow 0$; $D \leftarrow \frac{1}{m}\,\mathrm{diag}(\frac{1}{\|a_1\|^2}, \frac{1}{\|a_2\|^2}, \cdots, \frac{1}{\|a_m\|^2})$
2: **for** $k \leftarrow 0$ until convergence or max iteration **do**
3: $\quad x^{(k+1)} = x^{(k)} + \rho_k A^T D(b - Ax^{(k)})$
4: **end**

According to Cimmino, the initial point x^0 and its reflections with respect to all the projected hyperplanes lie on a hypersphere whose center is precisely the point common to all the hyperplanes;



(a) Events from Parkfield, California.

(b) Events from Mt. St. Helens, Washington.

Fig. 4. Non-uniform distribution of events; In (a) red stars (Shots) denote explosions (an alternate way to create events) with known location and origin times, black lines represent the faults i.e. SAF, San Andreas Fault; TMT, Table Mountain Thrust. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
Source: modified from *Zhang* et al. [33].

namely, the solution of the linear system. Now, since the center of gravity of the system of masses falls within the hypersphere, it follows that the new iterate x^1 is a better approximation to the solution than x^0, i.e. $\|x^1 - x_*\| < \|x^0 - x_*\|$, where x_* denotes the true solution.

In order to perform distributed tomography in sensor network, reconstruction algorithms such as CAV and CARP were found to be suitable [21]. These algorithms belong to parallel reconstruction techniques, where during each iteration, all the blocks (single equation in case of CAV) are processed in parallel (by some algorithm operating on each block independently of the others), and then the next iterate is formed by applying some operation to the partial solutions from the different blocks. These algorithms are also known as *string averaging* methods [7]. CARP in particular is designed as a general method and places no restriction on the system matrix or the selection of the blocks. CARP uses ART to perform local iteration. The partial solutions from all the blocks are then combined to form the next iterate using component wise averaging. CARP is shown to be equivalent to KACZ in some superspace and this makes it attractive for distributed tomography. The row projection property of KACZ, i.e. processing each row at a time, makes it promising to be implemented on tiny devices like BeagleBone black due to its small memory footprint (each row being sparse).

ART and its variants were mainly developed for medical tomography. Although the concept behind seismic and medical tomography are similar, there are major differences between them, such as: (i) Acoustic signals are used to probe the internal structure of the earth and they travel in a highly curved path due to reflection/refraction and also due to large dimensions of the observing zone. (ii) The measured travel time is a non-linear function of the velocity field unlike the medical tomography which has a linear relationship. (iii) Sources (i.e. earthquakes) and receivers (stations) are distributed in a non uniform manner and the ray coverage cannot be controlled and is often highly inhomogeneous (Fig. 4(a) and (b)). (iv) The source location and origin time are determined using the observations themselves which introduce uncertainties. These differences indicate that CARP, which uses ART for local computation, cannot be directly adopted to solve seismic tomography problems. In the next subsection we use another variant of ART called Bayesian ART (BART) to develop a distributed algorithm, and later we show how it is suitable for seismic tomography.

5.1. Seismic tomography using Bayesian ART

The travel time residual computed at each station contains noise due to measurement and also due to uncertainty in the initial velocity model. This causes large sparse linear system to be inconsistent. Seismic tomography problems are often ill-posed and ill-conditioned i.e. small changes in the travel time residual B will lead to large changes in the estimation of the slowness parameter x [15]. To show this effect mathematically, let us assume x^{exact} and x to be the exact and the perturbed solutions satisfying,

$$Ax^{exact} = b^{exact}; \qquad Ax = b = b^{exact} + e, \tag{10}$$

where, e denotes the perturbation or noise. Then the classical perturbation theory (see appendix A of [15]) leads to the bound

$$\frac{\|x^{exact} - x\|_2}{\|x^{exact}\|_2} \leq cond(A)\frac{\|e\|_2}{\|b^{exact}\|_2}, \tag{11}$$

where, $cond(A)$ denotes condition number of A. Now with large $cond(A)$, x can be very far from x^{exact} and this problem arises more often in the real experiments. Due to these difficulties associated with computing tomography, naive solutions are often expected to be useless. To obtain satisfactory results, we add additional constraints such as regularity or smoothness which suppresses some unwanted noise. This is called as *regularization*, and is required to avoid strong, undesired influence of small singular values dominating the solutions. The solution x_λ is defined as a solution to the regularized problem

$$x_\lambda = \arg\min_x \|Ax - b\|_2^2 + \lambda^2\|x\|_2^2. \tag{12}$$

Here, the *regularization parameter* λ is a positive number that controls the weight between $\|Ax - b\|_2^2$ which measures the goodness fit and $\|x\|_2^2$ which measures the regularity of the solution. The larger the value of λ, the more weight is given to the minimization of the solution norm, while on the other hand, small λ means more weight is given to fit the noisy data. This type of regularization is commonly known as the Tikhonov regularization [13].

To solve Eq. (12), we use the Bayesian ART proposed by G.T Herman [18,17]. Since the system is inconsistent, we consider, $Ax + r = b$, where r is chosen in such that given any x, $r = b - Ax$. With this assumption the equation now becomes well-posed. Therefore, we solve for x and r simultaneously given by,

$$\begin{bmatrix}\lambda I & A\end{bmatrix}\begin{bmatrix} r \\ x \end{bmatrix} = b. \tag{13}$$

The solution to the above consistent system is also the solution to the regularized Eq. (12). Now, by applying simple ART to Eq. (13) we obtain Bayesian ART (BART) given in Algorithm 3, where, a_i is the ith row of A (a_i^T is its transpose), $\hat{e}_i$ is a unit vector with ith element set to one, ρ_k relaxation parameter and λ is the regularization parameter. This method was applied to seismic tomography by Lees and Crosson [25] in which they compared BART with the CG method and found that they generate similar results.

Algorithm 3 Bayesian ART (BART)

1: Initialize: $r^0 \leftarrow 0$; $x^0 \leftarrow 0$
2: **for** $k \leftarrow 0$ until convergence or maximum number of iteration **do**
3: $\quad i \leftarrow k \bmod m + 1$
4: $\quad d^{(k)} = \rho_k \frac{b_i - \lambda(r^{(k)}) - (a_i^T \cdot x^{(k)})}{\lambda^2 + \|a_i\|^2}$
5: $\quad x^{(k+1)} = x^{(k)} + d^{(k)} a_i$
6: $\quad r^{(k+1)} = r^{(k)} + \lambda d^{(k)} \hat{e}_i$
7: **end**

Bayesian ART described in Algorithm 3 has similar structure to that of ART and is equal to ART for $\lambda = 0$. For $\lambda \neq 0$, the regularization parameter introduces an explicit weighting which was found to be desirable for seismic tomography problems due to large uncertainties [32]. The relaxation parameter ρ is used to stabilize the convergence and may remain constant or may change between iterations. While handling real data we have allowed $\rho_k = \frac{\kappa_1}{\kappa_2 + k}$, where κ_1 and κ_2 are suitably chosen constants. This way of reducing ρ_k is in accordance with convergence outlined by [31].

In reality, geological structure exhibits a smooth pattern (i.e. a gradual change in structural property). A-priori information such as location of faults or presence of specific buried features can be added to fine tune the computation and obtain a desired meaningful result. This process of adding additional constraints is called *smoothing*. Smoothing is introduced by applying a low-pass filter after each iteration, i.e. after all the equations in the system have been used. In this study, the updated version of the model is formed by taking a linear combination of the present and the smoothed version,

$$x^{k+1} = (1 - \Psi_k)x^k + \Psi_k x_{smooth}, \tag{14}$$

where, $\Psi_k = \kappa_3 \rho_k$ and $x_{smooth} = \sum_{j=1}^{m} a_j x^k$. This reduction in smoothing by relating it to the decreasing relaxation parameter ensures that the smoothness constraint does not dominate over reduction of data misfit. Different smoothing schemes that are popular in tomography computation are described in [17].

5.2. Distributed Bayesian ART

Until now, BART was used only in a centralized setup to generate seismic tomography. All the raw data sampled from each station had to be manually collected, pre-processed, analyzed, and then finally interpreted. Data gathering is the most challenging step, and this motivated us to develop in-network distributed algorithms. In this section, we provide details regarding Distributed Bayesian ART (D-BART) that can be implemented on sensor nodes to perform distributed tomographic inversion.

Suppose there are P sensor nodes in the network. Let the ith sensor perform event detection and ray tracing as mentioned in Section 3 to form $A_i \in \mathbb{R}^{m_i \times n}$ and $B_i \in \mathbb{R}^{m_i}$. Here n denotes the resolution of the tomography image and m_i is the number of rows of A_i, which is the number of p-waves detected and traced. Now we let A_i^j represent the jth column of the matrix A_i, and we identify the non-zero columns of A_i, i.e. for ith station and $1 \leq j \leq n$ we let

$$s_i^j = \begin{cases} 1 & \text{if } A_i^j \neq 0 \\ 0 & \text{if } A_i^j = 0. \end{cases} \tag{15}$$

This s_i^j is sent to a *SINK* node where $S^j = \sum_{i=1}^{P} s_i^j$ $1 \leq j \leq n$ is calculated. S^j denotes total number of blocks that have at least one non-zero element in the jth column. This S^j will be used to perform component averaging, where partial slowness obtained from each station will be combined with others. Next, we show how component averaging is done to obtain the next iterate.

Let $A = \{A_1, \ldots, A_P\}$ and $\bar{x}_i^j$ denote the jth component of partial slowness obtained from ith station after solving the linear equation $A_i x = B_i$. The component averaging operator relative to A is the transfer operator $CA_A : (\mathbb{R}^n)^P \to (\mathbb{R}^n)$ and is defined as follows: Let $\{\bar{x}_1, \ldots, \bar{x}_P\} \in \mathbb{R}^n$ be partial solution from all P sensor nodes. Then $CA_A(\bar{x}_1, \ldots, \bar{x}_P)$ is the point in $\mathbb{R}^n$ whose jth component is given by

$$CA_A(\bar{x}_1, \ldots, \bar{x}_P)^j = \frac{1}{S^j} \sum_{t=1}^{P} \bar{x}_t^j.$$

Algorithm 4 Distributed Bayesian ART (D-BART)

1: Initialize: $x^0 \leftarrow 0$; λ; κ_1; κ_2
2: **for** $k \leftarrow 0$ until convergence or maximum number of iteration **do**
3: $\quad$ **for** $i \leftarrow 1, \cdots, P$ execute in Parallel
4: $\quad\quad x_i^k = \text{BART-sweep}^t(A_i, b_i, x^{k-1}, \lambda, \rho_k)$
5: $\quad$ **end**
6: $(x^j)^k = \frac{1}{S^j} \sum_{i=1}^{P} (x_i^j)^k \quad \forall\, j = 1, \cdots, n$
7: **end**

Algorithm 4 presents Distributed Bayesian ART (D-BART) where BART-sweep^t denotes t internal iterations of BART algorithm 3. More details regarding component averaging mentioned in line-6 is provided in Algorithm 5.

5.3. Multi-resolution evolving tomography

As discussed in Section 2, the computation of travel time residual dependents highly on the slowness reference model. Initially, with no system information, we generate the unperturbed model x^0 based on historic data. Although the ground structure may have changed (shifting of magma beneath), it provides a good initial guess for the computation. From the experiments, researchers have found that tomography inversion is highly sensitive to the reference model [23]. Keeping this in mind, we introduced a multi-resolution evolving tomography scheme which computes a coarse resolution tomography when few earthquake events occur using the initial model x^0. This model is then refined based on coarser tomography and used to compute a finer resolution with the occurrence of more earthquakes.

This scheme was developed mainly to handle: (1) extremely large problem size for high dimensions approximately $160 \times 200 \times 24$; (2) occurrence of new events (earthquakes) and (3) continuously evolving magma model. Due to sparse occurrence of earthquake events we may not be able to generate high resolution tomography initially. But with the occurrence of earthquakes we must develop a mechanism to use the previous earthquake information and also the partial solution to obtain a new high resolution tomography. Therefore, we need to design a suitable interpolation operator that maps the solution from lower to higher dimensions and vice versa. This multi-level method was previously used to generate 3D tomography of local and deep velocity structure [3]. In this method, since the initial guess for the computation of higher resolution problem is obtained from the previous level, the initial guess in general will be closer to the actual solution rather than a random or zero vector. This can accelerate the convergence and reduce the computation cost, especially while solving problems involving high resolution (large column size). With the multi-resolution evolving scheme, the intermediate results can be retrieved in real-time without having

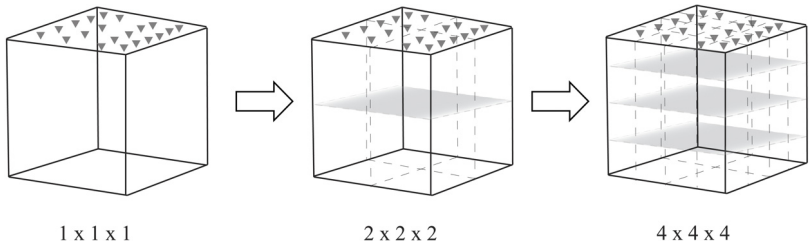
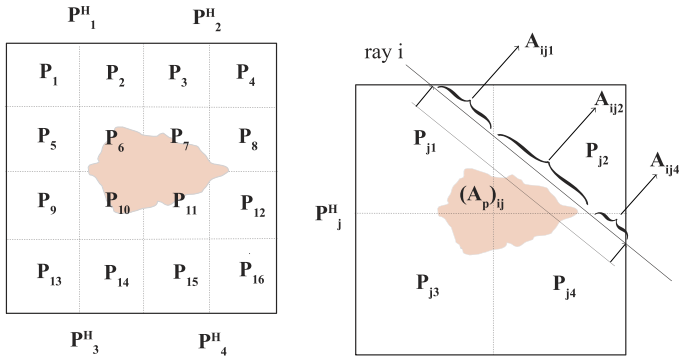


Fig. 5. Multi-resolution evolving scheme.



(a) Fine grid. (b) Coarse grid.

Fig. 6. Relation between fine and coarse grid.

to wait until all the computation has been completed. Fig. 5 shows the schematic diagram of our proposed multi-resolution scheme for seismic tomography.

We will now develop a suitable interpolation operator, and for simplicity we describe the 2D scenario.

Let n be the number of columns in A, suppose that $n = 4p$, and let $P_1, \ldots, P_n$ be the pixels on the *fine grid*. The *coarse grid* is obtained by combining four adjacent pixels of the fine grid as shown in Fig. 6(a). Let $S(j), j \in \{1, \ldots, p\}$ be the set of indices of the fine grid that form the coarse grid P_j^H, i.e.,

$$Q(j) = \{j_1, j_2, j_3, j_4\}; \quad \forall j = 1, \ldots, p;\ j_1 < j_2 < j_3 < j_4;$$
$$P_j^H = \{P_{j1} \cup P_{j2} \cup P_{j3} \cup P_{j4}\}.$$

From the above equation the coarse grid matrix A_p will be

$$A_p^{ij} = \sum_{k \in S(j)} A_{ik}, \quad \forall i \in \{1, \ldots, m\},\ j \in \{1, \ldots, p\}. \tag{16}$$

Now the interpolation operator I_p^n is given by

$$I_p^n = \begin{cases} 1 & \text{if } i \in Q(j) \\ 0 & \text{if } i \notin Q(j). \end{cases} \tag{17}$$

We now see that $A = A_p \times I_p^n$ satisfies the interpolation property. We also observe that I_p^n has full column rank.

Remark 1. Notice that the interpolation operator only increases the number of columns in matrix A. We can consider a similar operator which also reduces the rows by weighting them, however this is beyond the scope of this paper.

Remark 2. The above intergrid operators are designed for 2D cases, however the 3D case can be easily derived using $n = 8p$ i.e. a cuboid.

5.4. Component-averaged distributed multi-resolution evolving tomography

In this section, we present the complete seismic tomography algorithm with a multi-resolution evolving scheme (Algorithm 5) to distribute the computation load, and we compute the slowness model of the seismic tomography inversion problem in the sensor network.

For the convergence of CA-DMET, we refer to the averaging lemma of [14], according to which averaging operation in step 14 is equivalent to certain KACZ row projections with $\rho = 1$ in the superspace. Now for consistent case, BART in CA-DMET becomes equivalent to KACZ ($\lambda = 0$) and converges with positive number of internal sweeps in each block for the relaxation parameter ($0 < \rho < 2$) [1]. Similarly for inconsistent system, BART converges with cyclic relaxation parameters given by [31]. By combining these two results with averaging lemma, convergence of CA-DMET for both consistent and inconsistent can be obtained [14].

6. Evaluation and validation

In this section, we evaluate CA-DMET algorithm by performing extensive experiments both with the synthetic and the real data. The experiments were carried on CORE[2] and EMANE[3] network emulators [2] to emulate the sensor network with sensor nodes. 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.

6.1. System implementation

CA-DMET 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,

[2] http://cs.itd.nrl.navy.mil/work/core/.
[3] http://cs.itd.nrl.navy.mil/work/emane/.

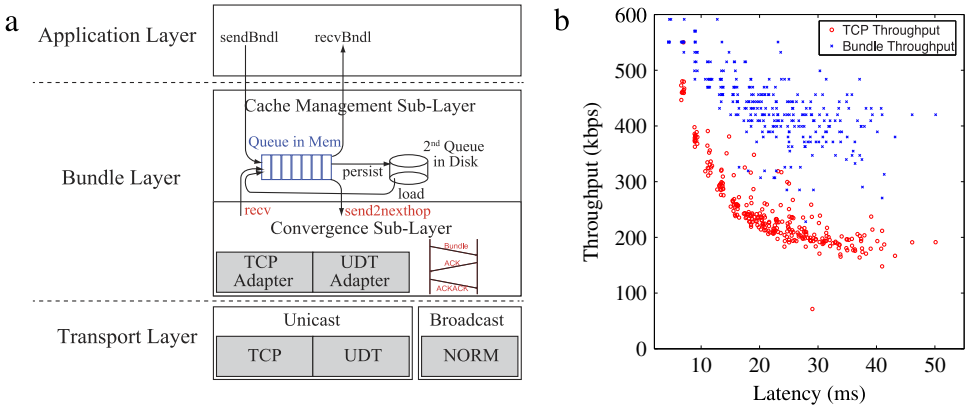


Fig. 7. (a) Bundle layer architecture (b) Comparison of TCP vs. Bundle layer using B.A.T.M.A.N. routing protocol.

Algorithm 5 Component-Averaged Distributed Multi-resolution Evolving Tomography

Initialize

Node ID *id*, resolution level $\ell \leftarrow 1$, START dim d, FINAL level L, Total Nodes P.
Current resolution dimension $Q = 2^{(\ell-1)}d \times 2^{(\ell-1)}d \times 2^{(\ell-1)}d$
Send initial slowness model $\mathbf{x}^{\ell}$ of resolution Q to each node

Repeat until Q reaches FINAL level L

Upon the detection of an event
  Trace the ray path a_e for every node
For each node i **do in parallel**
  For each $1 \leq j \leq Q$, calculate s_j^i
  Where $s_j^i = |I_j| = \{1 \leq i \leq P | x_j$ has nonzero coefficient in some equation of node i.
end for
Calculate $s_j = \sum_{i=1}^{P} s_j^i$
  $k \leftarrow 0, x^k \leftarrow x^{\ell}$
    while not converged **do**
      In every node for $1 \leq i \leq P$ do in parallel
      Perform BART-sweep^t on system of equations with initial guess x^k
      Aggregate the partial slowness perturbation $\bar{x}^t$ from all nodes at SINK and find the next iterate:

$$x_j^{(k+1)} = \begin{cases} \bar{x}_j^t & \text{if } s_j = 1 \\ \frac{1}{s_j} \sum_{t=1}^{P} \bar{x}_j^t & \text{if } s_j > 1 \end{cases}$$

      Send $x_j^{(k+1)}$ to all the nodes
      $k \leftarrow k + 1$
    end while
  $\mathbf{x} \leftarrow x^{k-1}$
Upon completing BART and obtaining final x
  Update slowness model: $\mathbf{x}^{\ell} = \mathbf{x}^{\ell} + \mathbf{x}$
  $\ell \leftarrow \ell + 1$
  if ℓ is equal to L
    TERMINATE
  else
    $Q = 2^{(\ell-1)}d \times 2^{(\ell-1)}d \times 2^{(\ell-1)}d$
    Increase the resolution of $\mathbf{x}^{\ell}$ to Q
  endif

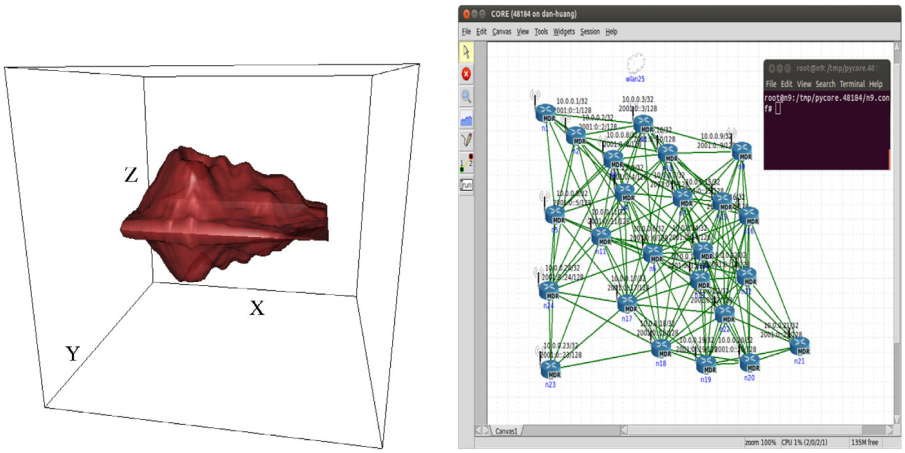
such systems may experience erratic link qualities and intermittent disconnections among nodes. These characteristics, combined with unpredictable environmental conditions, make it difficult to maintain efficient and reliable end-to-end connectivity that spans many hops. For example, the traditional end-to-end protocol like TCP is not suitable for a wireless mesh network in a challenging environment as the packet loss 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 Network (DTN) techniques to maintain efficient and reliable end-to-end connectivity that spans many hops for data delivery. In our design, the data is buffered into 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. Fig. 7(a) shows the application interfaces within each node which is used for the integration of DTN and the routing protocol. To generate these routing tables in this paper we use B.A.T.M.A.N (better approach to mobile ad-hoc networking) [19]. This routing protocol does not maintain full route to the destination and maintains only the information about the next link through which a best route can be found. In our paper, we use BATMAN routing protocol along with the bundle layer to ensure reliable multi-hop delivery.

Fig. 7(b) shows that the Bundle Layer outperforms TCP with B.A.T.M.A.N. routing protocols. 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. For CA-DMET 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 send its partial solution. Next, the SINK node computes the $(k+1)$th iterate and disseminates the solution back to all nodes. The implementation of all the algorithms are 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.

6.2. Tomographic imaging result

Typically, to test tomography inversion algorithm, a synthetic model is used. This serves two purposes: (a) the real data set such as from Mt. St Helens do not have an exact ground truth (b) simulations 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. Therefore, we first test our algorithm using a synthetic model and later with the real data trace from Mt. St. Helens.



(a) 3D Synthetic phantom. (b) Station setup in CORE.

Fig. 8. (a) 3D synthetic magma model (b) Snapshot of the CORE GUI which displays the nodes along with communication link.

Table 1
The four different test cases.

Case no.	Dimension(D)	Events	Total rays(R)	Ratio(R/D)
1	$8 \times 8 \times 8$	100	10 000	19.53
2	$16 \times 16 \times 16$	300	30 000	7.32
3	$32 \times 32 \times 32$	500	50 000	1.52
4	$32 \times 32 \times 32$	900	90 000	2.74

6.2.1. Synthetic phantom

To evaluate the algorithm, the data generator is implemented to generate a magma area and earthquake events assuming the tomography model is a cube of dimension $10 \times 10 \times 10$ km. Then we set a predefined magma area as the ground truth as shown in Fig. 8(a). The velocities of seismic waves inside and outside the magma area are V and $0.9V$ where V is 4.5 km/s which is a typical P-wave velocity.

A network of 100 nodes (Fig. 8(b)) are setup in CORE emulator to monitor the magma area. We set the final tomography resolution to be $32 \times 32 \times 32$ where each block is of the size 0.315 km^3. The data generator then generates earthquake events with random location and time, and calculates ray travel time from event location to all sensor nodes. To simulate the event location estimation and ray tracing errors, a white Gaussian noise is added to the travel time. Each node can calculate the predicted travel time based on the initial model in different resolution.

To evaluate the CA-DMET algorithm, three different resolution levels are used i.e. 8, 16 and 32. At first, we start with lower resolution $8 \times 8 \times 8$ and the data generator generates 100 earthquake events and since there are 100 sensor nodes the total number of rays add up to $100 \times 100 = 10\,000$. Next the resolution is increased to $16 \times 16 \times 16$ and finally increased to $32 \times 32 \times 32$. Case (4) in Table 1 denotes a centralized setup with 100 nodes that receives rays from 900 events. Here, the 100 nodes later relay their rays to a single station for centralized computation.

Note: Events column in Table 1 refers to the new events generated for a particular resolution, and in our case since we use ray information interpolated from a lower resolution the total events for case (2) and case (3) becomes 400 and 900 respectively, i.e. sum of previous events. For this reason, to have a fair comparison we generate a total of 900 events for the case (4).

In the implementation, the Bayesian ART method is performed for 10 iterations locally to solve the equation system on each node. We use the relative slowness perturbation updates of the estimation between the two sweeps (one sweep means that all partial slowness perturbation is averaged to calculate next iterate) as the stopping criteria. If the relative update (ϕ) is less than a tolerance, the CA-DMET stops. To compare the performance of CA-DMET, we also used the centralized Bayesian ART to solve the system at target resolution with all 900 events (case (4)).

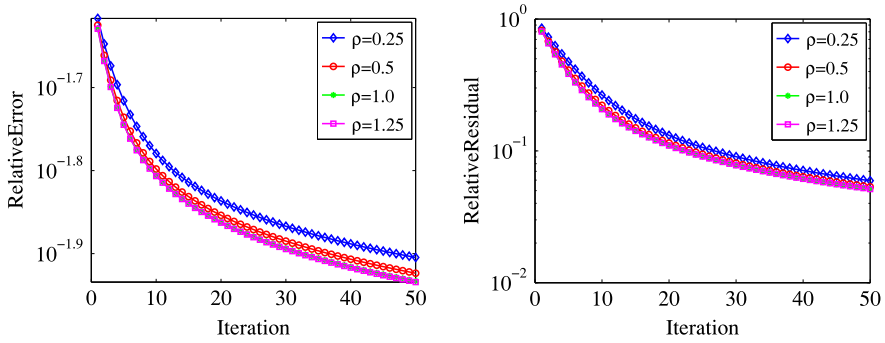
Performance of CA-DMET is compared using their relative update ($\phi = |x^{(k+1)} - x^{(k)}|/|x^{(k)}|$), relative residual ($\chi = \|Ax^k - B\|/\|B\|$) and relative error ($\delta = |x^{(k)} - x^{truth}|/\|x^{truth}\|$).

6.3. Correctness and accuracy

We first run experiments to show the behavior of different relaxation parameters on CA-DMET and also choose an optimal ρ for a given set of synthetic data. CA-DMET is performed on case 3 by varying ρ from 0.25 to 1.25 and the plots are shown in Fig. 9. Starting from a small relaxation parameter, each successive value of ρ increases the convergence rate until an optimal value is reached. We found that further increase in the relaxation parameter worsened the result and an optimal relaxation parameter was found to be 1.25 for the given synthetic magma model. In all our experiments with synthetic data, the relaxation parameter remained constant throughout the iterations, i.e., $\rho^k = \rho$ for all $k \geq 0$.

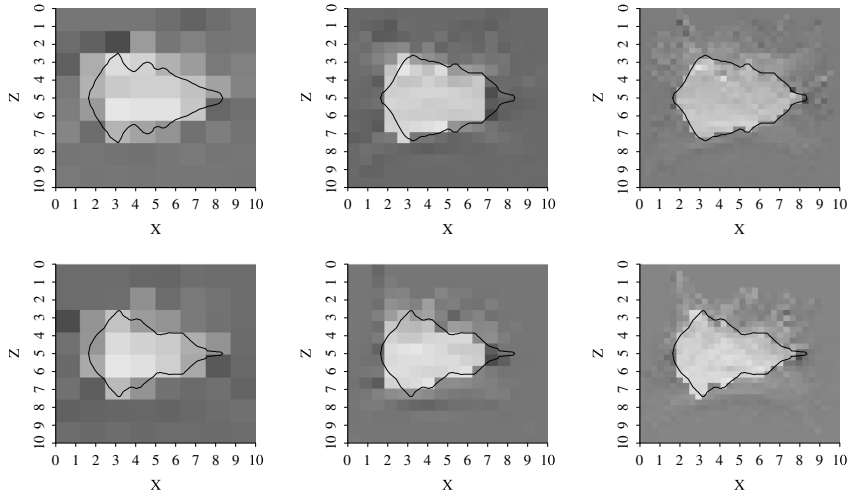
In the next set of experiments we intend to demonstrate the correctness of our algorithm through visualization. CA-DMET is stopped when relative update $\phi \leq 0.001$. Fig. 10 shows the result slice by slice along the X and Y axes. Fig. 10(a),(b) and (c) are the results from CA-DMET with resolution dimension 8, 16 and 32 respectively. Fig. 10(d) is obtained from centralize Bayesian ART computation and in Fig. 10(e) we have the ground truth. Each row of figure shows the same tomography slice on some layer along with X and Y axis (the total layers of each figure is equal to the resolution dimension of the result). The black polygons give the cross section outline of the surface of magma area represented in Fig. 8. Although CA-DMET fails to construct an accurate image in low resolution, it still provides a good starting point for the higher resolution computation to further refine the result. By obtaining the region of interest and initial slowness, the velocity model is updated and with resolution dimension 16, the result can closely show the outline of the magma area. By comparing Fig. 10(c) and (d), we observe that the result obtained from CA-DMET and centralized BART are almost similar. This suggests that CA-DMET can be a good candidate for distributed tomographic inversion.

In the next set of experiments, we compare the relative performance of CA-DMET with three different algorithms: CAV, Cimmino and un-weighted DROP [6]. We use relative error as the parameter for comparison and results shown in Fig. 11(a) demonstrate that there is a difference in the initial convergence

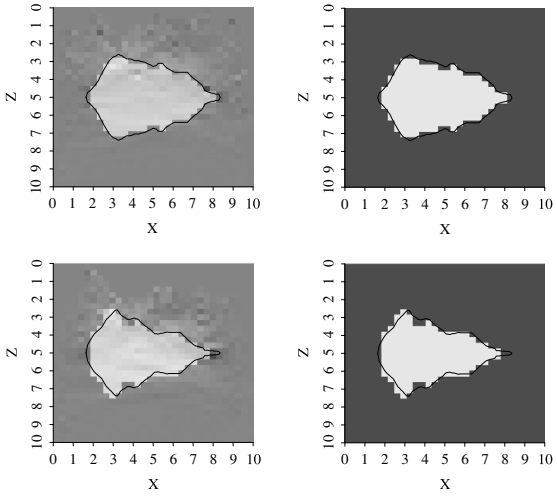


(a) Relative error. (b) Relative residual.

Fig. 9. Error measurement for CA-DMET with various relaxation parameters—Case 1.



(a) CA-DMET Res: 8^3. (b) CA-DMET Res: 16^3. (c) CA-DMET Res: 32^3.



(d) Centralize Res: 32^3. (e) Phantom Res: 32^3.

Fig. 10. 2D Tomography Rendering of a 3D phantom. Each row shows 2D slice of certain layer.

between CAV, Cimmino, DROP and CA-DMET. A visual verification of all the algorithms is shown in Fig. 12. All the algorithms are run for same number of iterations. The reconstructed image from different algorithms reveals that CA-DMET is able to obtain a better magma image with less perturbation noise outside the region compared to other algorithms.

6.3.1. *Seismic tomography with single and multi-resolution*

Previously, in Section 5.3 we mentioned few advantages of multi-resolution evolving tomography scheme over single resolution tomography. In this section, we will demonstrate this using synthetic data simulation. For this experiment, we use the settings mentioned in Table 1. In case of multi-resolution, we start

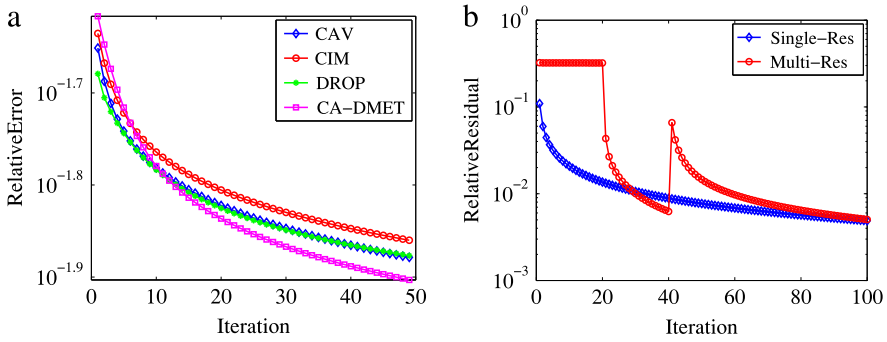
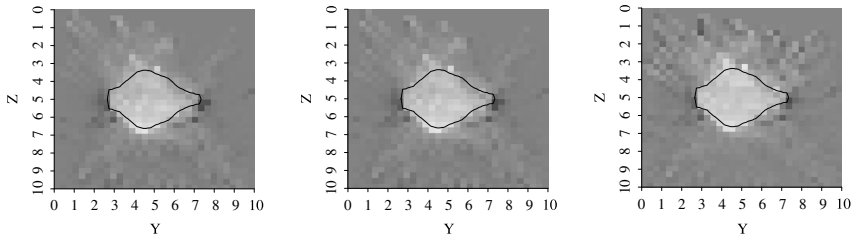
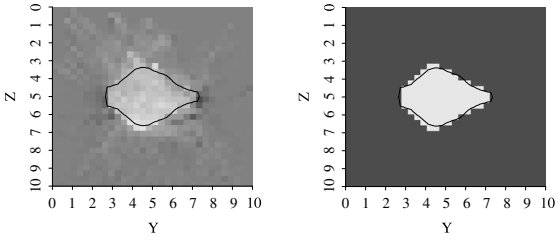


Fig. 11. (a) Comparison of different algorithm (b) Comparison of relative residual between single and multi-resolution.



(a) Cimmino Res: 32^3. (b) CAV Res: 32^3. (c) DROP Res: 32^3.



(d) CA-DMET Res: 32^3. (e) Phantom Res: 32^3.

Fig. 12. 2D tomography rendering from different algorithms.

with $8 \times 8 \times 8$ (Table 1 case-1) with zero initial guess and run 20 iterations to generate x. Next, we interpolate this x to 16^3 (case-2) and again run for another 20 iterations. Finally, we perform case-3 with an initial guess obtained from 16^3 and run the algorithm for 60 iterations. In case of single resolution, we use case-4 i.e. 32^3 with zero initial guess and run for 100 iterations. Fig. 11(b) compares the convergence rate of both single and multi-resolution. We see that the multi-resolution can achieve similar residual as that of single-resolution with only 60 iterations, provided it gets initial values from previous resolutions. Note that, the total computation cost for multi-resolution will be lower than that of single-resolution. This is due to the fact that the computation cost is lesser for lower dimension system for the same number of iterations.

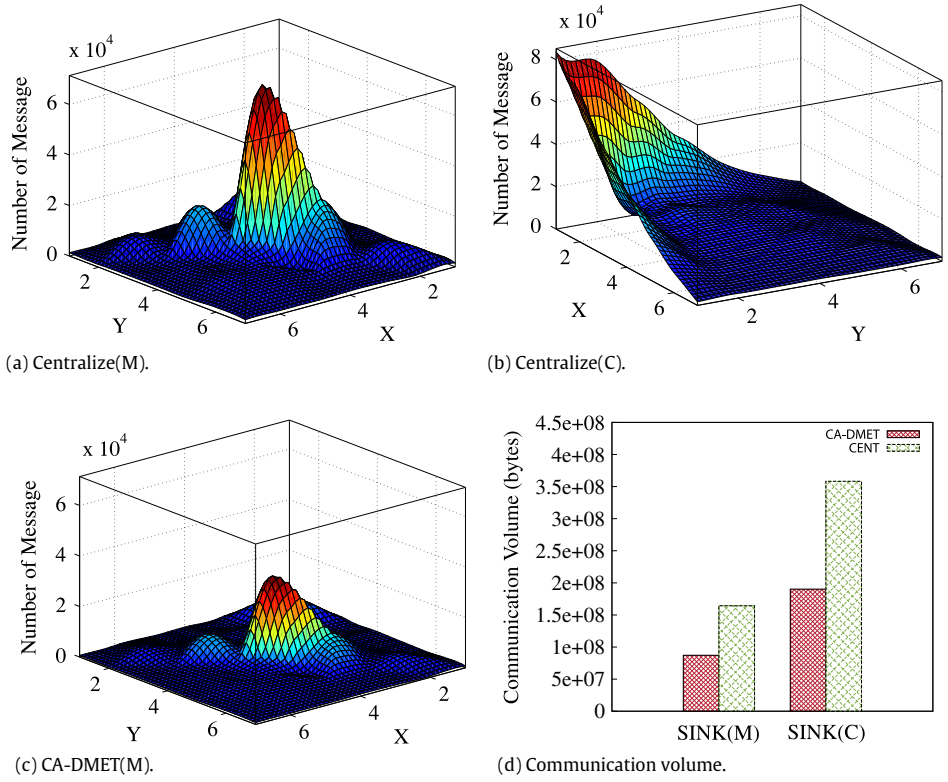
6.3.2. Communication cost and robustness

Earlier we had seen the summation of partial solution on the network using bundle layer. In CA-DMET, aggregation and dissemination of the computed result constitutes the major part of communication. In this section, we evaluate the communication cost and compare it to the centralized algorithm. In the centralized scheme, each station send its corresponding ray information to a base station or SINK placed (1) at the corner CENT(C) and (2) in the middle CENT(M) using similar aggregation protocol

From Fig. 13(a) and (b), we can see that communication cost in a centralized setup is high near the SINK as all the rays are transferred over the network. The volume of data are dependent on the number of earthquake events and also the number of stations. Fig. 13(c) shows the communication pattern for CA-DMET and from this we can see that the communication cost in CA-DMET is lesser than the centralized scheme. This is mainly because communication cost in CA-DMET depends on number of iterations and typically with semi-convergent property of iterative methods the number of iterations are much less compared to number of earthquake events. We validate this effect by comparing volume (bytes) transferred by CA-DMET to that of centralized (CENT). Fig. 13(d) shows that in both cases SINK(M) and SINK(C) the volume of bytes transferred is less in case of CA-DMET. The communication cost of the centralized algorithm increases with occurrence of each event and also with increase in station number.

In CA-DMET although we are able to distribute the computation part, the communication part still relies on a SINK node to perform summation which might create bottleneck. We are currently investigating methods to perform average using gossip methods [5] that involves information exchange only with the neighbors. Since the variables shared by two or more nodes need to be averaged, we intend to use gossip methods that converge to average consensus such as [5]. In this way, we can achieve a decentralized way to compute average that will reduce the communication cost providing a flat and balanced communication pattern.

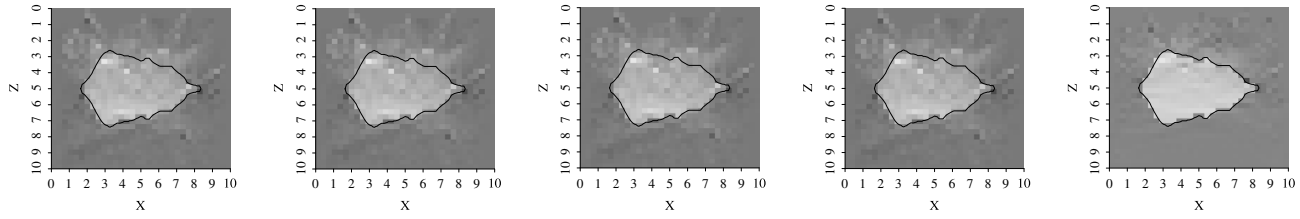
In the next set of experiments, loss tolerance and robustness of CA-DMET are evaluated. The algorithm runs with the same configuration for packet loss ratios of 10%, 20% and 40% in the emulator. Fig. 14 gives part of a 2D slice rendered along Y axes



(a) Centralize(M). (b) Centralize(C).

(c) CA-DMET(M). (d) Communication volume.

Fig. 13. Comparison of communication cost between centralized data gathering and CA-DMET.



(a) 10% loss. (b) 20% loss. (c) 40% loss. (d) CA-DMET no loss. (e) Centralized no loss.

Fig. 14. Impact of packet loss in CA-DMET.

with packet loss. We can see that with 10%, 20% or even 40% packet loss, there is no significant difference in terms of the magma area outline when compared to the results with no packet loss. Since the computation is distributed and all the nodes are involved in slowness calculation, the proposed algorithm is tolerant to a severe packet loss.

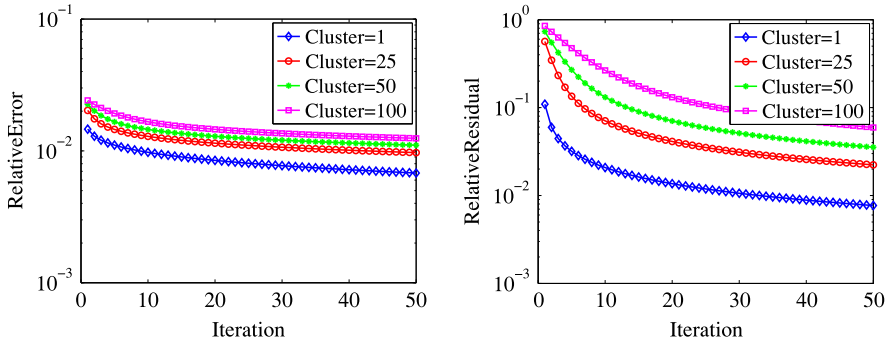
Synthetic tomography experiments until now had all the 100 nodes participating in sensing, computation and communication, i.e., with no cluster. Now, we will compare the efficiency of our algorithm by partitioning 100 nodes into different clusters. Here, all nodes are responsible for sensing, however, only cluster heads take part in computation and communication. Cluster heads receive ray information from the nodes in their cluster. We run the simulation for different number of cluster heads i.e. 1 (centralized) through 100 (no cluster). From Fig. 15 we can see that as the number of cluster heads increases the convergent rate decreases. This is mainly due to the increase in the communication overhead. Although the centralized solution has a higher rate of convergence, we can observe from Fig. 10(c) and (d) that there is no significant difference between them in terms of visualization. From this we can conclude that there is an optimal partition for a given set of nodes that balances the communication cost and the computing load over the network.

7. Imaging of Mount St. Helens, Washington, USA

Mount St. Helens (MSH), WA, USA situated in pacific northwest cascade region erupted on May 18, 1980 and was one of the deadliest volcano in the history of United States, killing 57 people and destroying several homes. MSH is one of the most widely studied volcano in the world mainly due to its recent volcanic activity and also due to its unusual sideways eruption, which surprised many scientists and questioned many of the existing theories. Due to its location close to human habitat there is an increasing effort to understand the dynamics of this volcano and obtain higher resolution imagery.

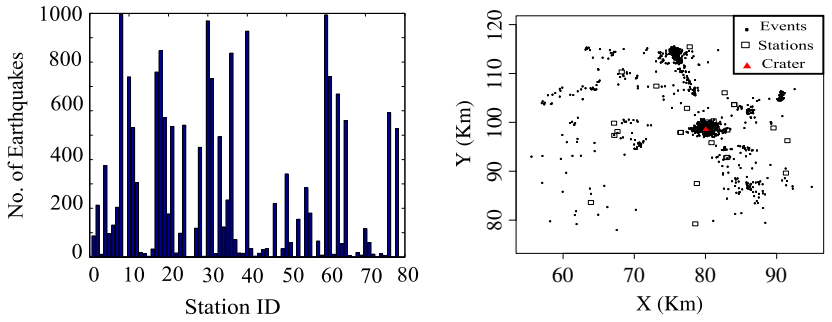
7.1. Experiment setting

In this section, we study and present a P-wave velocity model for Mount St. Helens (MSH) using our proposed distributed algorithm. This result is verified by comparing it with the centralized algorithm similar to [24]. We used the data set from [24] which has data from 78 stations spread over 160 × 200 kms in area. The depth we will analyze in this paper is up to 24 km. A total of 1141 earthquake events are used which gave around 18 161 rays. Each station can record up to 1141 earthquake, however due to factors such as location and accuracy of stations



(a) Relative error. (b) Relative residual.

Fig. 15. Performance of CA-DMET with different partition.



(a) Node wise event distribution. (b) Events from Mt. St Helens, Washington.

Fig. 16. Non-uniform distribution of rays and events at Mt St Helens. (a) Number of event from Mt. St. Helens detected by different nodes. (b). Black dots and squares denote the location of the earthquake and station location respectively and the red triangle denotes the location of the crater. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the distribution of events varies significantly. Fig. 16 shows the distribution of rays traced across each station. For distributed tomography ray count at each station is equivalent to the number of rows of the linear system at that station. This is very important as it determines whether the system is under-determined or not which can effect the rate of convergence as seen earlier in Fig. 15. If the number of rows in each station is less than a threshold then CA-DMET can work in a clustered scheme. In this paper we partition 78 stations into 4 and 8 clusters with each cluster containing approximately 4540 and 2270 rows respectively. The column size (i.e. resolution dimension) is around $160 \times 200 \times 24 = 768\,000$ much higher compared to [24]. Although this system is highly under-determined the results show that CA-DMET is able to obtain result close to that of the centralized solution.

7.2. Results

Geophysicists calculate relative seismic velocity variations ($Vrel_P$) to monitor the volcano edifice. This velocity can be represented as a 3D map of slices of earth, depicting the regions of higher or lower seismic wave velocity. The region where seismic waves move more slowly, is due to the presence of warm, partially melted rock; that is the magma storage region. In Fig. 17, negative value represents a slower velocity region and positive value represents a higher velocity region. This value is relative to initial velocity model. To calculate $Vrel_P$, we need to know the initial slowness model (Sls) and slowness perturbation (Sls_P).

From CA-DMET, we obtain x the slowness perturbation (Sls_P) i.e. how much did the slowness perturb from the initial slowness model (Sls). Real slowness model is now obtained by; $Sls_R = Sls + Sls_P$. Since, velocity is inversely related to slowness, we can obtain initial velocity and real velocity model by $Vel = 1/Sls$ and $Vel_R = 1/Sls_R$ respectively. Similarly, perturbed velocity can be found using $Vel_P = Vel_R - Vel$. Finally, we obtained relative velocity perturbation using $Vrel_P = Vel_P/Vel$. Fig. 17 shows the $Vrel_P$ map of MSH obtained from CA-DMET and compared with one obtained from the centralized method. Notice that unlike synthetic data used in the previous section, there is no ground truth for the velocity of MSH. Hence, we focus on the comparison of the proposed methods with the centralized processing scheme, which can be used as a benchmark that fully utilizes the data available. Interpretation of this data requires in-depth knowledge of geophysics and is out of the scope of this paper.

8. Discussion and future work

The inversion process in seismic tomography is highly nonlinear and is often influenced with measurement noise. This measurement noise along with noise in initial velocity model can lead to incorrect event location estimation. To obtain final tomography many steps like raw data collection, earthquake detection, ray tracing are equally necessary. These pre-processing steps are all inherently distributed and can be calculated by single station without communicating with other stations. We have implemented and tested these pre-processing steps on the embedded device and submitted the findings else where.

The proposed distributed algorithm for obtaining seismic image involves some kind of aggregation of partial slowness model at every iteration step. The communication complexity depends on number of iteration which cannot be predicted in these kind of inverse problems. Although, in experiments we saw that we need fewer than 50 global iterations, this might still cause a bottleneck at the SINK node. To avoid this we are currently developing a gossip based aggregation method which performs average operation over the network and still able to obtain satisfactory result. These type

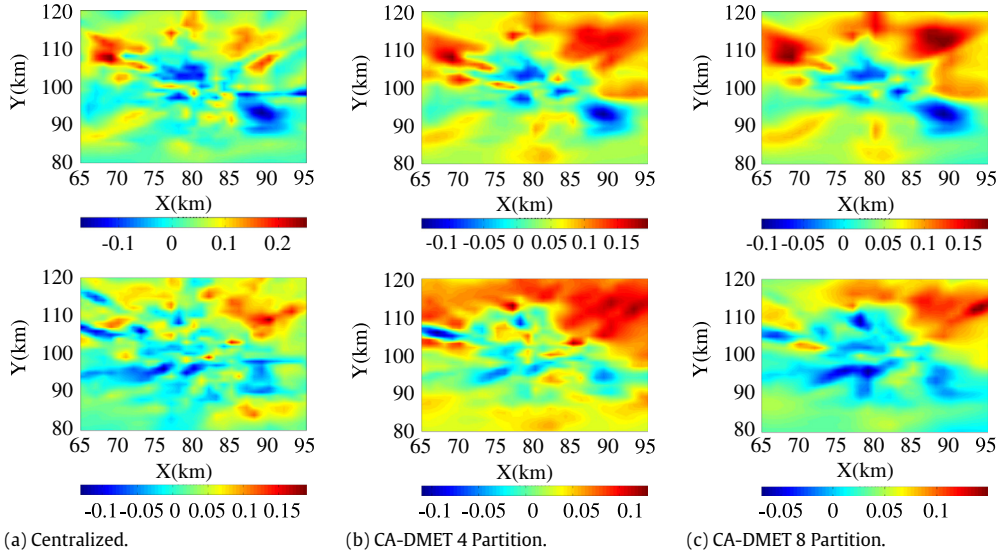


Fig. 17. Relative velocity perturbation map of MSH at different depths. Top row shows $Vrel_p$ at depth 2.9 km,bottom row is at depth 4.9 km.

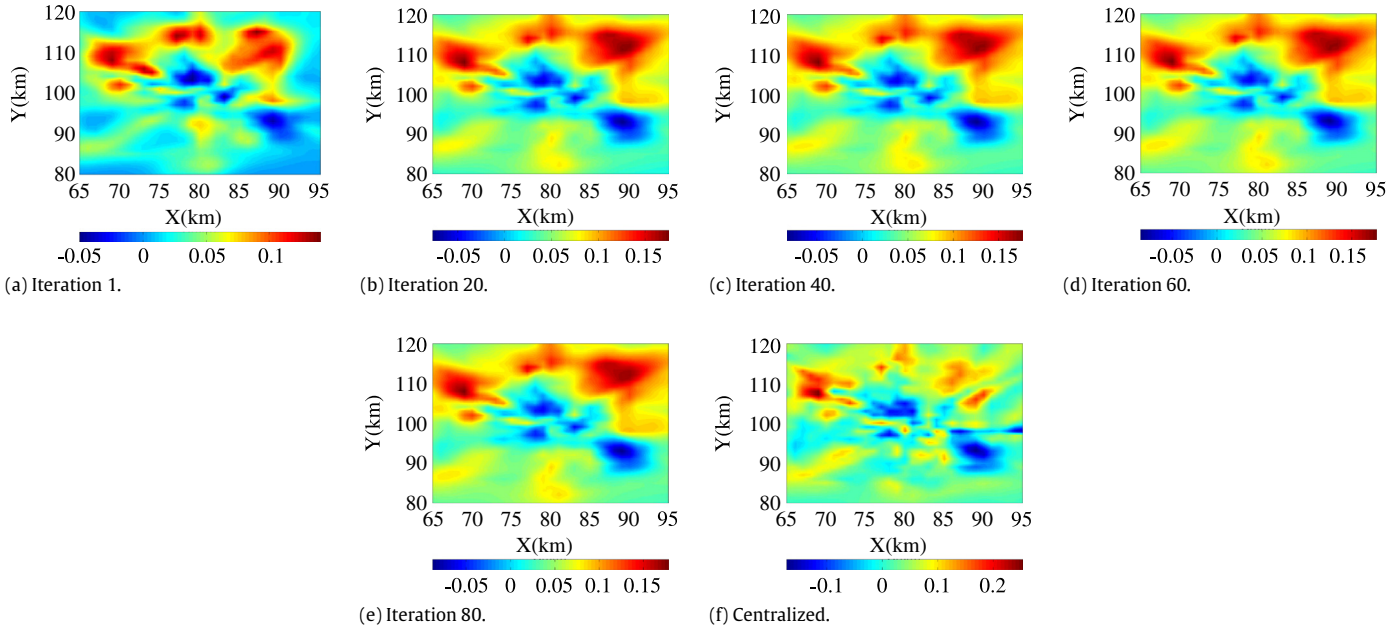


Fig. 18. Relative velocity perturbation map of MSH (depth 2.9 km) at different iteration. (a)–(e) refer model using CA-DMET and (f) using the centralized BART.

of methods can also make the entire method asynchronous with no routing overhead.

In Fig. 18 we plot the velocity map at different iterations. We can see that the velocity map gets updated with increase in the iterations. However, we observed that after iteration 40 (Fig. 18(c)) the result starts to diverge from the centralized solution (Fig. 18(f)). This reminds us the role of regularization and smoothing, i.e. increase in the iteration can make the image more smooth or algebraically get different solution. To avoid this we need an appropriate stopping criteria which can stop the iterative methods automatically when the result is achieved. The methods like L-curve and generalized cross validation are used to automatically pick the suitable stopping criteria [15]. In practice, seismologists do not have a standard method and they decide through visualization and domain knowledge. Our future research will also focus on deciding a suitable stopping criteria for distributed tomography inversion. The application of this algorithm in reservoir monitoring [27] can also be very useful for oil and gas exploration [22] which can bring down their exploration time from weeks to days or even hours.

9. Conclusion

In this paper, we presented a component-averaged multi-resolution evolving tomography that distributes the tomographic inversion and balances computational load to the network, while computing real-time high-resolution 3D tomography in the network. We first presented the design of distributed algorithm which used BART and performed component averaging over sensor networks. The multi-resolution was introduced keeping in mind the need for real time imaging. We also provided the suitable intergrid operator which acted as a transformation matrix that mapped lower dimension matrix to higher and vice versa depending on the need. We also showed that it was equivalent to ray tracing. This algorithm was motivated by the results of CARP

and it was modified to suite the seismic tomography. Algorithm was first tested using synthetic data on CORE emulator and the result shows that this method converges. We also compared the message exchange overhead, which was found to be less compared to the centralized solution. Later, we also tested our algorithm using real data traces from Mt. St. Helens and the result obtained from CA-DMET was close to that of the centralized method. The robustness and loss tolerance feature of CA-DMET is very attractive which makes it very suitable for sensor network applications.

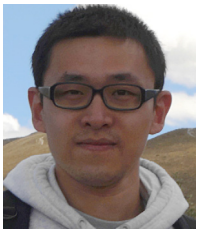
Acknowledgments

This research is partially supported by NSF-CNS-1066391, NSF-CNS-0914371, NSF-CPS-1135814 and NSF-CDI-1125165. We also thank Dr. Greg Waite for providing the recent data set of Mt. St Helens and help us visualize the velocity model. We thank Dr. Dan Gordon for few valuable suggestions regarding component averaging on sensor networks.

References

[1] R. Aharoni, Y. Censor, Block-iterative projection methods for parallel computation of solutions to convex feasibility problems, Linear Algebra Appl. 120 (1989) 165–175. http://dx.doi.org/10.1016/0024-3795(89)90375-3.
[2] J. Ahrenholz, T. Goff, B. Adamson, Integration of the CORE and EMANE network emulators, in: Military Communications Conference, 2011—MILCOM 2011, 2011, pp. 1870–1875.
[3] C.-y. Bai, S. Greenhalgh, 3D multi-step travel time tomography: Imageing the local, deep velocity structure of Rabaul volcano, Papua New Guinea, Phys. Earth Planet. Inter. 151 (3–4) (2005) 259–275. http://dx.doi.org/10.1016/j.pepi.2005.03.009.
[4] D.P. Bertsekas, J.N. Tsitsiklis, Some aspects of parallel and distributed iterative algorithms—A survey, Automatica 27 (1) (1991) 3–21.
[5] S. Boyd, A. Ghosh, B. Prabhakar, D. Shah, Randomized gossip algorithms, IEEE Trans. Inform. Theory 52 (6) (2006) 2508–2530. http://dx.doi.org/10.1109/tit.2006.874516.
[6] Y. Censor, T. Elfving, G.T. Herman, On diagonally-relaxed orthogonal projection methods, SIAM J. Sci. Comput. 30 (1) (2007) 473–504. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.333.960.
[7] Y. Censor, D. Gordon, R. Gordon, Component averaging: An efficient iterative parallel algorithm for large and sparse unstructured problems, Parallel Comput. 27 (6) (2001) 777–808.
[8] G. Cimmino, Calcolo approssimato per le soluzioni dei sistemi di equazioni lineari, XVI (9) (1938) 326–333, La Ric. Sci. 16 (9) (1938) 326–333.
[9] P.P.B. Eggermont, G.T. Herman, A. Lent, Iterative algorithms for large partitioned linear systems, with applications to image reconstruction, Linear Algebra Appl. 40 (1981) 37–67. http://dx.doi.org/10.1016/0024-3795(81)90139-7.
[10] J.M. Elble, N.V. Sahinidis, P. Vouzis, GPU computing with Kaczmarz's and other iterative algorithms for linear systems, Parallel Comput. 36 (2010) 215–231.
[11] R. Fischer, J.M. Lees, Shortest path ray tracing with sparse graphs, Geophysics 58 (7) (1993) 987–996.
[12] L. Geiger, Probability method for the determination of earthquake epicenters from the arrival time only, Bull. St. Louis. Univ. 8 (1912) 60–71.
[13] G.H. Golub, P.C. Hansen, D.P. O'Leary, Tikhonov regularization and total least squares, SIAM J. Matrix Anal. Appl. 21 (1) (1999) 185–194. http://dx.doi.org/10.1137/s0895479897326432.
[14] D. Gordon, R. Gordon, Component-averaged row projections: a robust, block-parallel scheme for sparse linear systems, SIAM J. Sci. Comput. 27 (2005) 1092–1117.
[15] P.C. Hansen, Discrete Inverse Problems, Society for Industrial and Applied Mathematics, 2010, http://dx.doi.org/10.1137/1.9780898718836.
[16] M.T. Heath, E. Ng, B.W. Peyton, Parallel algorithms for sparse linear systems, SIAM Rev. 33 (3) (1991) 420–460.
[17] G.T. Herman, Reconstruction from Projections: The Fundamentals of Computerized Tomography, Academic Press, 1980.
[18] G.T. Herman, H. Hurwitz, A. Lent, H.-P. Lung, On the Bayesian approach to image reconstruction, Inf. Control 42 (1979) 60–71.
[19] D. Johnson, N. Ntlatlapa, C. Aichele, A simple pragmatic approach to mesh routing using BATMAN, in: 2nd IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries, 2008.
[20] S. Kaczmarz, Angenäherte Auflösung von systemen linearer Gleichungen, Bull. Int. Acad. Pol. Sci. Lett. 35 (1937) 355–357.
[21] G. Kamath, L. Shi, W.-Z. Song, Component-average based distributed seismic tomography in sensor networks, in: 2013 IEEE International Conference on Distributed Computing in Sensor Systems, DCOSS, 2013, pp. 88–95. http://dx.doi.org/10.1109/DCOSS.2013.17.
[22] P. Kearey, M. Brooks, I. Hill, An Introduction to Geophysical Exploration, Blackwell Science, 2002, URL: http://www.worldcat.org/isbn/9780632049295.
[23] E. Kissling, W.L. Ellsworth, D. Eberhart-Phillips, U. Kradolfer, Initial reference models in local earthquake tomography, J. Geophys. Res. 99 (B10) (1994) 19635–19646. http://dx.doi.org/10.1029/93jb03138.
[24] J.M. Lees, The magma system of Mount St. Helens: non-linear high-resolution P-wave tomography, J. Volcanol. Geotherm. Res. 53 (1992) 103–116.
[25] J.M. Lees, R.S. Crosson, Bayesian art versus conjugate gradient methods in tomographic seismic imaging: An application at Mount St. Helens, Washington, Inst. Math. Stat. 20 (1991) 186–208.
[26] G. Liu, R. Tan, R. Zhou, G. Xing, W. Song, J. Lees, Volcanic earthquake timing using wireless sensor networks, 2013, pp. 91–102.
[27] D. Lumley, Time-lapse seismic reservoir monitoring, Geophysics 66 (1) (2001) 50–53. http://dx.doi.org/10.1190/1.1444921.
[28] R.A. Renaut, A parallel multisplitting solution of the least squares problem, Numer. Linear Algebra Appl. 5 (1) (1998) 11–31.
[29] Y. Saad, Iterative Methods for Sparse Linear Systems, second ed., Society for Industrial and Applied Mathematics, 2003.
[30] W.-Z. Song, R. Huang, M. Xu, A. Ma, B. Shirazi, R. Lahusen, Air-dropped sensor network for real-time high-fidelity volcano monitoring, in: The 7th Annual International Conference on Mobile Systems, Applications and Services, MobiSys, 2009.
[31] M.R. Trummer, Reconstructing pictures from projections: On the convergence of the ART algorithm with relaxation, vol. 26, no. 3, 1981, pp. 189–195. http://dx.doi.org/10.1007/bf02243477.
[32] A. van der Sluis, H.A. van der Vorst, Numerical solution of large, sparse linear algebraic systems arising from tomographic problems, in: G. Nolet (Ed.), Seismic Tomography, in: Seismology and Exploration Geophysics, vol. 5, Springer, Netherlands, 1987, pp. 49–83. http://dx.doi.org/10.1007/978-94-009-3899-1_3.
[33] H. Zhang, C. Thurber, P. Bedrosian, Joint inversion for Vp, Vs, and Vp/Vs at SAFOD, Parkfield, California, Geochem. Geophys. Geosyst. 10 (11) (2009) Q11002+. http://dx.doi.org/10.1029/2009gc002709.

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 Ph.D. 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 B.Sc. degree in Software Engineering from Tongji University in 2007, and the M.Sc. degree in Computer Science from Shanghai Jiao Tong University in 2010. He is currently pursuing the Ph.D. degree in the Department of Computer Science of Georgia State University. His research interests include wireless sensor networks, in-network processing and distributed systems.

Wen-Zhan Song is a tenured associate 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, etc. since 2005, and resulted in 80+ journal articles, conference articles and book chapters in this area.

Jonathan M. Lees did his undergraduate training at the University of Illinois in Physics/Maths. After graduating he worked as a Geophysicist at Shell Oil, quit after 5 years and continued his education at University of Washington, Seattle, in Geophysics, graduating in 1989. He then did two short Post-Docs at UC Santa Barbara and Japan before landing a faculty position at Yale University where he taught for 10 years. In 2000 he moved to UNC-Chapel Hill, where he has been teaching since. His work is mainly focused on quantitative analysis in seismology and volcanology, and occasional time series analysis for climate change.