8-9-2016

Real-time In-situ Seismic Tomography in Sensor Network

Lei Shi

REAL-TIME IN-SITU SEISMIC TOMOGRAPHY IN SENSOR NETWORK

by

LEI SHI

Under the Direction of Wenzhan Song, PhD

ABSTRACT

Seismic tomography is a technique for illuminating the physical dynamics of the Earth by seismic waves generated by earthquakes or explosions. In both industry and academia, the seismic exploration does not yet have the capability of imaging seismic tomography in real-time and with high resolution. There are two reasons. First, at present raw seismic data are typically recorded on sensor nodes locally then are manually collected to central observatories for post processing, and this process may take months to complete. Second, high resolution tomography requires a large and dense sensor network, the real-time data

retrieval from a network of large-amount wireless seismic nodes to a central server is virtually impossible due to the sheer data amount and resource limitations. This limits our ability to understand earthquake zone or volcano dynamics.

To obtain the seismic tomography in real-time and high resolution, a new design of sensor network system for raw seismic data processing and distributed tomography computation is demanded. Based on these requirements, three research aspects are addressed in this work. First, a distributed multi-resolution evolving tomography computation algorithm is proposed to compute tomography in the network, while avoiding costly data collections and centralized computations. Second, InsightTomo, an end-to-end sensor network emulation platform, is designed to emulate the entire process from data recording to tomography image result delivery. Third, a sensor network testbed is presented to verify the related methods and design in real world. The design of the platform consists of hardware, sensing and data processing components.

INDEX WORDS: Sensor Network, Seismic Tomography, Distributed Computing, In-network Processing.

REAL-TIME IN-SITU SEISMIC TOMOGRAPHY IN SENSOR NETWORK

by

LEI SHI

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2016

REAL-TIME IN-SITU SEISMIC TOMOGRAPHY IN SENSOR NETWORK

by

LEI SHI

Committee Chair: Wenzhan Song

Committee: Xiaojun Cao

Xiaolin Hu

Xiaojing Ye

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

August 2016

DEDICATION

This dissertation is dedicated to Georgia State University. This dissertation is also dedicated to my parents Tingli Shi and Cuizhen Zhang, for their lasting love, sacrifice and support.

ACKNOWLEDGEMENTS

This dissertation work would not have been possible without the support of many people. I want to express my appreciation to my advisor Dr. Wenzhan Song for his kind support and guidance on my study, research and life. I also want to thank all my committee members, Dr. Xiaojun Cao, Dr. Xiaolin Hu and Dr. Xiaojing Ye for their kind help and support in the past several years.

I would like to express my gratitude to Dr. Jonathan M. Lees, Dr. Zhigang Peng and Dr. Yao Xie for their sincere help and guidance on my research, particularly on some areas that I have never touched before.

I also want to thank all my friends in the lab for their friendship and help during my study and research, especially Dr. Mingsen Xu, Dr. Debraj De, Dr. Qingjun Xiao, Dr. Song Tan, Dr. Liang Zhao and Mr. Goutham Kamath.

Finally I thank NSF for their support with grants NSF-CNS-1066391, NSF-CNS-0914371, NSF-CPS-1135814 and NSF-CDI-1125165.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

- GSU - Georgia State University
- WSN - Wireless Sensor Network
- NSF - National Science Foundation

PART 1

INTRODUCTION

Volcanic eruption is one of the most dangerous threats to life on the Earth. In recent years, more volcano activities have drawn the attention of public and scientists. Most existing volcano monitoring systems employ expensive broadband seismometer as instrumentation. Also at present raw seismic data are typically collected at central observatories for post processing. Seismic sampling rates for volcano monitoring are usually in the range of 16-24 bit at 50-200 Hz. With such high-fidelity sampling, it is virtually impossible to collect raw, real-time data from a large-scale dense sensor network, due to severe limitations of energy and bandwidth at current, battery-powered sensor nodes. As a result, at some most threatening, active volcanoes, fewer than 20 nodes [1] are thus maintained. With such a small network and post processing mechanism, existing system do not yet have the capability to recover physical dynamics with sufficient resolution in real-time. This limits our ability to understand volcano dynamics and physical processes inside volcano conduit systems. Substantial scientific discoveries on the geology and physics of active volcanism would be imminent if the seismic tomography inversion could be done in real-time and the resolution could be increased by an order of magnitude or more. This requires a large-scale network with automatic in-network processing and computation capability.

To date, the sensor network technology has matured to the point where it is possible to deploy and maintain a large-scale network for volcano monitoring and utilize the computing power of each node for signal processing and distributed tomography inversion in real-time. The methods commonly used today in the procedure of seismic tomography computation cannot be directly employed under field circumstances proposed here because they rely on centralized algorithms and require massive amounts of raw seismic data collected on a central processing unit. Thus, real-time seismic tomography of high resolution requires a new

mechanism with respect to system design, information processing and tomography inversion computation.

In this chapter, we first give an introduction on seismic tomography. Then we present the background and focus of this work and the organization of this dissertation.

1.1 First-arrival Traveltime Tomography

Seismic tomography is a technique for imaging the subsurface of the Earth with seismic waves produced by earthquakes or explosions. The first-arrival traveltime tomography uses P-wave first arrival times at sensor nodes to derive the internal velocity structure of the subsurface. The basic workflow of traveltime tomography illustrated in Figure 1.1 involves four steps.

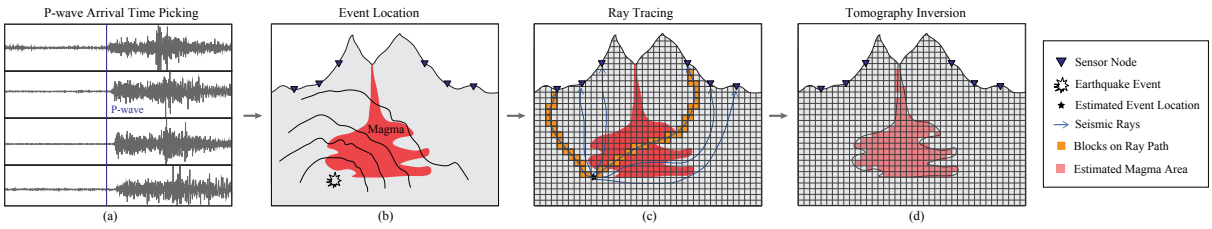


Figure 1.1 Workflow of 3D First-arrival Traveltime Tomography.

(a) P-wave Arrival Time Picking. Once an earthquake event happens, the sensor nodes that detect seismic disturbances record the signals. The P-wave arrival times need to be extracted from the raw seismic data.

(b) Event Location. The P-wave arrival times and locations of sensor nodes are used to estimate the event hypocenter and origin time in the volcanic edifice.

(c) Ray Tracing. Following each event, seismic rays propagate to nodes and pass through anomalous media. These rays are perturbed and thus register anomalous residuals. Given the source locations of the seismic events and current velocity model, ray tracing is to find the ray paths from the event hypocenters to the nodes.

(d) Tomography Inversion. The traced ray paths, in turn, are used to image a 3D tomography model of the velocity structure. As shown in Figure 1.1 the volcano is partitioned

into small blocks and the seismic tomography problem can be formulated as a large, sparse matrix inversion problem.

1.2 Research Background and Focus

This work is part of the project VolcanoSRI (Volcano Seismic Realtime Imaging). This project aims to create a new paradigm for imaging 4D (four-dimensional) tomography of an active volcano in real-time. VolcanoSRI is a large-scale sensor network of low-cost geophysical stations that analyzes seismic signals and computes real-time, full-scale, three-dimensional fluid dynamics of the volcano conduit system within the active network. The computed 4D tomography model will illuminate complex, time-varying dynamics of an erupting volcano, providing a deeper scientific understanding of volcanic processes, as well as a basis for rapid detection of volcanic hazards.

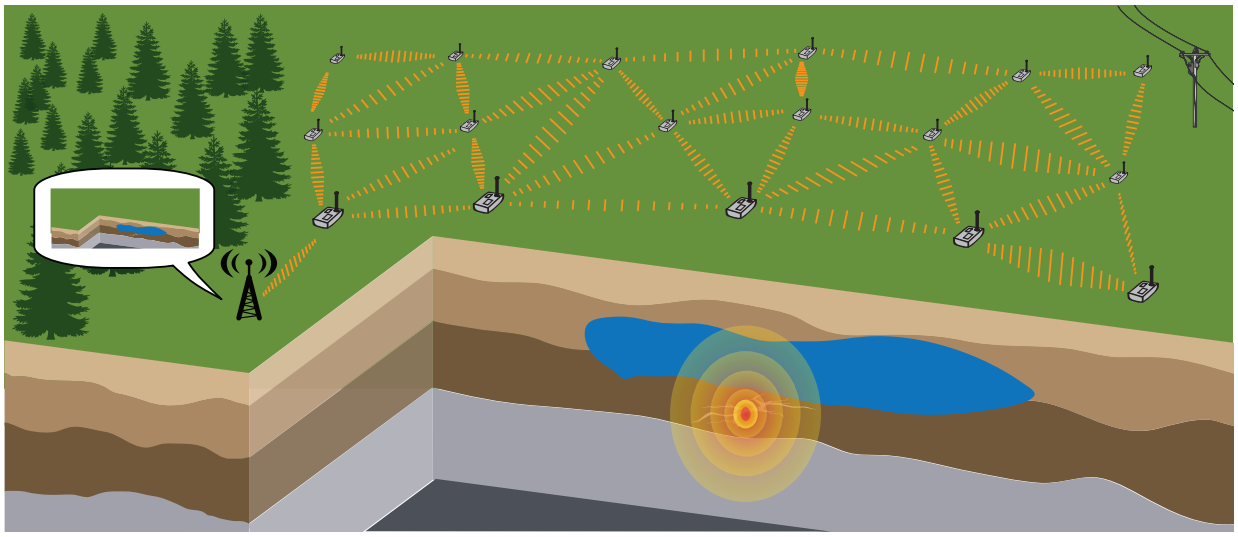


Figure 1.2 Sensor Network for Seismic Tomography.

Realizing the VolcanoSRI system requires a transformative study on the science of complex volcano systems and the design of large-scale sensor networks. Our approach integrates innovations on distributed tomographic algorithms, collaborative signal processing and situation-aware networking technology for large-scale real-time sensor systems. The distributed tomography algorithm disperses the computational burden to the sensor nodes and

performs real-time tomographic inversion within the network. Such an approach has never been attempted before and represents a major achievement for both earth and computer science. The team is composed of computer and earth scientists including early pioneers of wireless sensor networks as applied to volcano monitoring. Figure 1.2 shows the idea for real-time in-situ 3D tomography computation with VolcanoSRI.

The rest of this dissertation is organized as follows. In chapter 2, we give a survey of the related works. In chapter 3, we propose the distributed multi-resolution evolving tomography algorithm for tomography computation. In chapter 4, we present InsightTomo - an in-situ seismic tomographic imaging system framework in sensor network. In chapter 5, we give the design and implementation of the testbed system platform. Finally we conclude this dissertation in chapter 6.

PART 2

RELATED WORKS

In this chapter we discuss the related works in the literature of seismic tomography, solutions for event location, ray tracing, arrival time picking and the least-squares problem.

Static tomography inversion for 3D structure, applied to volcanoes and oil field explorations, has been explored since the late 1970's [2] [3] [4]. In volcano applications, tomography inversion used passive seismic data from networks consisting of tens of nodes, at most. The development and application to volcanoes include Mount St. Helens [5] [6] [7], Mt. Rainier [8], Kliuchevskoi, Kamchatka, Russia [9], and Unzen Volcano, Japan [10]. At the Coso geothermal field, California, researchers have made significant contributions to seismic imaging by coordinating tomography inversions of velocity [11], anisotropy [12], attenuation [13] and porosity [14].

Sensor network has been deployed for monitoring in many different areas. In [15], the sensor network was deployed to collect dense environmental and ecological data about populations of rare species and their habitats. Another sensor system was used by the researchers to monitor the habitat of the Leach's Storm Petrel at Great Duck Island [16]. The Zebranet [17] project uses sensor network nodes attached to zebras to monitortheir movements via GPS. It is composed of multiple mobile nodes and a base station with occasional radio contact. The sensor network was also used to monitor the bridge health [18] [19] and weather condition as well [20].

For volcano monitoring, The first volcanic monitoring work using WSN was developed in July 2004 [21], by a group of researchers from the Universities of Harvard, New Hampshire, North Carolina, and the Geophysical Institute of the National Polytechnic School at Reventador in Ecuador. Data collection was performed with continuous monitoring during 19 days. In 2008, a smart solution was proposed for collecting reliable information aiming to

improve the collection of real-time information. The sensor network was deployed on Mount St. Helens [1] for volcano hazard monitoring and run for months.

P-wave arrival picking has been studied by the community. One widely used approach is the STA/LTA method [22] that has been using in real deployment [1] on volcano monitoring. The STA/LTA method continuously monitors the ratio of short-term average over long-term average on a signal. Since it is based based on RSAM (Realtime Seismic Amplitude Measurement), which is calculated on raw seismic data samples every second, the accuracy of STA/LTA method is not enough for tomography computation. Some methods either determined through joint AR modeling of the noise and the seismic signal [23] or based on in-network collaborative signal processing [24] are proposed. In seismic tomography, the event location can also be formulated as a least-squares problem by Geiger's Method [25]. This problem can be solved by conjugate gradient method or row action method [26]. For ray tracing, each node can trace the ray paths based on a reference model with either bending or shooting methods [27] [28] [29] [30] [31]. This can be naturally distributed since the ray tracing computation is entirely local.

The methods to solve least-squares problem mainly fall into two categories, direct methods and iterative methods. Iterative methods for solving large sparse linear systems of equations are advantageous over the classical direct solvers, especially for huge systems [32]. Methods of parallelizing least-squares solutions on distributed memory architecture have been studied for both direct and iterative methods, but there are few studies on distributing the least-squares solutions from a wireless sensor network point of view.

A class of direct solver for least-square problem is through QR decomposition. Straková, Gansterer and Zemen investigate randomized algorithms based on gossiping for the distributed computation of the QR factorization [33]. The algorithm is based on modified Gram-Schmidt orthogonalization (mGS). To distribute the mGS algorithm, they pursue a randomized decentralized QR factorization algorithm based on gossiping. Gossip-based (or epidemic) algorithms are characterized by asynchronous randomized information exchange, usually only within the local neighbors of each node. The advantage of this method is that

the communication is entirely local, the problem is gossiping based algorithm converges slow and after QR decomposition a back back substitution is required to get the least-square solution where the nodes need to perform the computation sequentially.

More works have been done on parallelizing iterative algorithms. Renaut proposed a multisplitting solution of the least-squares problem [34] where the solutions to the local problems are recombined using weighting matrices to pick out the appropriate components of each subproblem solution. This algorithm updates the solution by finding the optimal update with respect to the weights of the recombination. The problem to distribute this algorithm in the network is that it requires each node to broadcast the residual updates per iteration. Yang and Brent describe a modified CGLS (MCGLS) method to reduce inner products global synchronization points, respectively, then improve the parallel performance accordingly [35]. This can also be potentially distributed over the network but the broadcast communication is still required per iteration.

In the literature of signal processing, there are a few studies on consensus-based Distributed Least Mean Square (D-LMS) algorithms [36] [37] [38] [39] [40] [41] in sensor networks. These algorithms adopted the weighted sum of local estimations to achieve consensus. Each sensor node maintains its own local estimation and, to reach the consensus, needs to continuously exchange the estimation with only its neighbors in the network. Since if the process is ergodic and stationary, the least-square estimator approaches the least mean square estimator as the size of the data set grows, this can also be used for least-square solutions statistically. The problem is that the consensus-based methods can be slow [42] on convergence and the communication cost increases along the the estimation vector dimension increases (in our problem, high resolution seismic has a high-dimension estimation $\mathbf{s}$). Besides, in a large-scale sensor network, the hop distance between two sensor nodes might be very long. In such situation, to achieve consensus, any pair of nodes need to exchange high-dimension estimations frequently. This not only means high communication overhead but also introduces long delays involving many multi-hop communications. Sayed and Lopes developed a Distributed Recursive Least-Squares (D-RLS) strategy by appealing to collaboration tech-

niques that exploit the space-time structure of the data, achieving an exact recursive solution that is fully distributed [43]. This method requires a cyclic path in the network to perform the computation on the nodes sequentially and exchanging a dense matrix between nodes.

The most popular iterative method was proposed by Kaczmarz (KACZ) [44] which is a form of alternating projection method. This method is also known under the name Algebraic Reconstruction Technique (ART) in computer tomography [45]. This algorithm do not require the full design of 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 till it converges. The other variants of iterative methods are symmetric ART (symART) [46] and Simultaneous ART (SART) [47]. In SymART, one cycle in ART is followed by another cycle in reverse order while in SART, block of equations are projected instead of just single equation. All these methods are centralized and cannot be directly applied to distributed seismic tomography.

The block parallel versions of ART have also been proposed and widely used algorithms among them are component averaging (CAV) [48], Block Iterative- Component Averaging (BI-CAV) [49] and component-averaged row projections (CARP) [50]. These block-parallel algorithms use string averaging technique to combine the intermediate result of each block by taking regular weighted average. The main idea here is to utilize the sparsity of the system matrix as the weight for averaging. A survey paper comparing various block parallel methods based on their performance on GPU's are discussed in [51]. From the above methods, CARP is more generalized and places no restriction on the system matrix or the selection of blocks. CARP does not require any pre-processing or pre-ordering of the matrix and provides very robust method unlike any other block parallel algorithm for solving large sparse linear system. In CARP, a finite number of KACZ method is applied in each block and the resulting point in each block is averaged to get next iterate. It is also proved to converge for large scale systems [50].

In [52] we 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 [53]. However, only until early 90's it was used by seismology community to solve inverse problem on small set of data [54]. [31] used SVD to interactively change the boundaries while, [55] used genetic algorithm to optimize the parametrization. These algorithms were suitable for small size data sets and required high computational power to run efficiently. [56] 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.

Here is an analysis of the communication upper bound to distribute some of the methods mentioned above. For more details about how to distribute the algorithms and the communication analysis, please refer to our technical report on a survey of distributed least-square computing in networks [57]. Consider a linear system $Ax = b$ where $A \in \mathbb{R}^{m \times n} (m \geq n)$ and $x \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$, a least-squares problem is defined as $\min_{x \in \mathbb{R}^n} ||Ax - b||_2$. The distributed least-square algorithms can be implemented in a multi-hop networks with N nodes and converge in $O(k)$ iterations. In D-LMS algorithm, D_{avg} denotes the average node degree of the network. All the methods above have been proved to be convergent mathematically, but there are several system design problems for directly applying them in the wireless sensor networks. In the seismic tomography inversion problem, we intend to solve a large sparse system (n may be as large as tens of thousands even millions) in a large-scale network (hun-

	Algorithm	Communication Cost
1	distributed QR	$O(kNn^2)$
2	Multisplitting	$O(kN^2m)$
3	distributed MGLS	$O((k+1)N(m+n))$
4	D-LMS	$O(kN(D_{avg}+1)n)$
5	D-RLS	$O(Nn^2)$
6	CARP	$O(2kNn)$

Table 2.1 Communication Cost Analysis

dreds of nodes), usually $N \ll m$ and $N \ll n$. Considering this, the communication cost of Multisplitting method and CARP is less than other methods if the iteration numbers are in the same order among these algorithms. But it is hard to bound the iteration numbers of the methods since it highly depends on the system itself (matrix condition number). Besides, except for the D-LMS method, other methods either requires broadcast communication per iteration or a path in the network to perform the computation sequentially on the nodes. Broadcast communication brings not only high communication cost but also difficulties to maintain a stable protocol for communication. Since the convergence analysis of these methods is based on the information completeness, the result is unpredictable if data loss happens in the communication among iterations. On the other hand, sequential computation along a path in the network or too many iterations with communication will introduce delays and may not meet the real-time requirement of the system. In next chapter, we will discuss how to address these challenges and present an distributed least-square solution which avoids long-term broadcast communication and delay, at the same time, can still approximate the least-square solution.

PART 3

DISTRIBUTED MULTI-RESOLUTION EVOLVING TOMOGRAPHY ALGORITHM

In this chapter, we first give the formulation of the seismic tomography computation. Then we present the distributed multi-resolution evolving tomography algorithm and gives the formal description of the algorithm. Also, we give an initial evaluation of the proposed algorithm on correctness, computation and communication cost and tolerance at the end of this chapter.

3.1 Problem Formulation

Based on the first-arrival traveltime tomography principle discussed in chapter 1. In Tomography Inversion the volcano is partitioned into small blocks and the seismic tomography problem can be formulated as a large, sparse matrix inversion problem.

Suppose that there are N nodes and J earthquakes, we consider a perturbation approach here. Let $\mathbf{s}^*$ be the slowness (reciprocal of velocity) model of the volcano with resolution M (blocks). $\mathbf{s}^*$ can be assumed to be a reference model, $\mathbf{s}^0$, plus a small perturbation $\Delta\mathbf{s}$, i.e., $\mathbf{s}^* = \mathbf{s}^0 + \Delta\mathbf{s}$. For simplicity, we use $\mathbf{s}$ denote $\Delta\mathbf{s}$ in the following discussion. The initial reference model is usually given by the interior layered structure of the earth in seismic tomography.

We can estimate the ray travel times in Event Location by the arrival times and estimated event origin times. Let $\mathbf{t}_i^* = [t_{i1}^*, t_{i2}^*, \ldots, t_{iJ}^*]^T$, where t_{ij}^* is the travel time experienced by node i in the j-th event. Based on the ray paths traced in Ray Tracing, the travel time of a ray is the sum of the slowness in each block times the length of the ray within that block, i.e., $t_{ij}^* = \mathbf{A}_i[j,m] \cdot \mathbf{s}^*[m]$ where $\mathbf{A}_i[j,m]$ is the length of the ray from the j-th event to node i in the m-th block and $\mathbf{s}^*[m]$ is the slowness of the m-th block. Let $\mathbf{t}_i^0 = [t_{i1}^0, t_{i2}^0, \ldots, t_{iJ}^0]^T$

be the unperturbed travel times where $t_{ij}^0 = \mathbf{A}_i[j,m] \cdot \mathbf{s}^0[m]$. In the rest of this section, we use *observed travel time* and *predicted travel time* to indicate the same meanings of t_{ij}^* and t_{ij}^0. In matrix notation we have the following equation,

$$\mathbf{A}_i\mathbf{s}^* - \mathbf{A}_i\mathbf{s}^0 = \mathbf{A}_i\mathbf{s} \tag{3.1}$$

where $\mathbf{A}_i[j,m]$ represents the element at the j-th row and m-th column of matrix $\mathbf{A}_i \in \mathbb{R}^{J\times M}$. Let $\mathbf{t}_i = [t_{i1}, t_{i2}, \ldots, t_{iJ}]^T$ be the travel time residual such that $\mathbf{t}_i = \mathbf{t}_i^* - \mathbf{t}_i^0$, equation (3.1) can be rewritten as,

$$\mathbf{A}_i\mathbf{s} = \mathbf{t}_i \tag{3.2}$$

We now have a linear relationship between the travel time residual observations, $\mathbf{t}_i$, and the slowness perturbations, $\mathbf{s}$. Since each ray path intersects with the model only at a small number of blocks compared with M, the design matrix, $\mathbf{A}_i$, is sparse. The seismic tomography inversion problem is to solve the system,

$$\mathbf{A}\mathbf{s} = \mathbf{t} \tag{3.3}$$

where $\mathbf{A} = [\mathbf{A}_1^T, \mathbf{A}_2^T, \ldots, \mathbf{A}_N^T]^T$ and $\mathbf{t} = [\mathbf{t}_1^T, \mathbf{t}_2^T, \ldots, \mathbf{t}_N^T]^T$. This system is usually overdetermined and the inversion aims to find the least-squares solution $\mathbf{s}$ such that,

$$\mathbf{s} = \arg\min_{\mathbf{s}} \parallel \mathbf{t} - \mathbf{A}\mathbf{s} \parallel^2 \tag{3.4}$$

In seismic tomography, the event location can also be formulated as a least-squares problem by Geiger's Method [25], and the estimation vector is of length 4 (event origin time and 3D coordinates). Since the dimension of the estimation vector is fixed and small, a centralized solution can be applied in the network for this problem. In Ray Tracing, each node traces ray path based on a reference model. This can be naturally distributed since the ray tracing computation is entirely local. The fourth step is the most computationally intensive and time consuming aspect of high resolution seismic tomography. The sparse system

can be solved by conjugate gradient method or row action method [26]. However, designed for high-performance computers, these centralized approaches need significant amount of computational/memory resources and require the knowledge of global information. As a result, they cannot be directly distributed in wireless sensor network. Thus, the key research challenge here is how to solve the least-squares problem in Tomography Inversion distributedly under the severe constraints of wireless sensor network. In this chapter, we focus on distributed tomography inversion algorithm, while assuming that the event arrival timing at each node has been extracted from the raw seismic data by each node itself [23] [24], as well as that the event location and ray tracing have been done. We will discuss more about arrival timing, event location and ray tracing in chapter 4.

3.2 Algorithm

We developed a new tomography partition and computation distribution algorithm with a multi-resolution evolving scheme to distribute the computation load, reduce the communication cost and approximate the least-squares solution of the seismic tomography inversion problem in the network. To distribute the computation load, we first partition the volcano structure geometrically and the system $\mathbf{As} = \mathbf{t}$ correspondingly. Then some nodes are selected as *landlord*s to compute part of the tomography model. The computation on each landlord is entirely local so that the communication cost is bounded. Since the computation on each landlord only uses part of the system $\mathbf{As} = \mathbf{t}$, the result is not equivalent to the solution of the original system. To approximate the optimal solution, we introduce the multi-resolution evolving scheme: the network initially computes a coarse resolution tomography without partition when small amount of earthquake events arrive; as more and more earthquake events arrive, the network will compute finer and finer resolution tomography with more partitions. The intuition behind this is that the network first computes an outline of the volcano structure in an low resolution then fills up with finer details inside. With the multi-resolution evolving scheme, we do not need to wait for all computation done and can retrieve the intermediate results under low resolutions in a real-time manner.

In this section, first we use an example to show the idea of the tomography partition and the computation distribution over the network; second we introduce the multi-resolution evolving scheme and give the description of the algorithm.

3.2.1 Tomography Partition

The example in Figure 3.1 illustrates that how to partition the volcano structure geometrically (vertically) and the corresponding system $\mathbf{As} = \mathbf{t}$ for distributing the computation in the network.

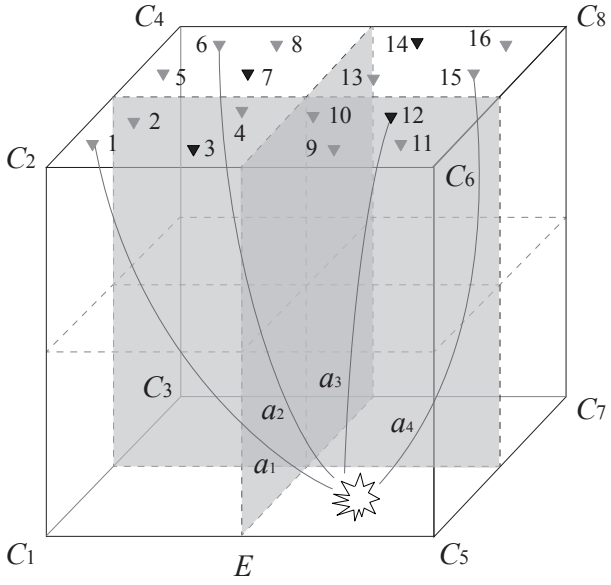


Figure 3.1 Tomography Partition. The resolution of cube E is $2 \times 2 \times 2$ (8 blocks C_1 to C_8) and 16 sensor nodes are deployed on top of E.

In this example, cube E is vertically partitioned into 4 parts (E_1 to E_4) and there are 4 sensor nodes in each partition, e.g., node 1, 2, 3 and 4 are on top of E_1 consisting of blocks C_1 and C_2. Suppose that one earthquake happens in block C_5 and node 1, 6, 12 and 15 detect this event. Once the event location is done, these 4 nodes do ray tracing individually and get 4 ray paths a_1, a_2, a_3 and a_4. Assume that a_1 penetrates C_5, C_1 and C_2, a_2 penetrates C_5, C_1, C_3 and C_4, a_3 penetrates C_5 and C_6, and a_4 penetrates C_5, C_7 and C_8. These 4 ray

paths can form a system $\mathbf{As} = \mathbf{t}$ as following,

$$\begin{bmatrix} a_{1,1} & a_{1,2} & 0 & 0 & a_{1,5} & 0 & 0 & 0 \\ a_{2,1} & 0 & a_{2,3} & a_{2,4} & a_{2,5} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{3,5} & a_{3,6} & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{4,5} & 0 & a_{4,7} & a_{4,8} \end{bmatrix} \cdot \begin{bmatrix} s_1, & s_2, & s_3, & s_4, & s_5, & s_6, & s_7, & s_8 \end{bmatrix}^T = \begin{bmatrix} t_1, & t_2, & t_3, & t_4 \end{bmatrix}^T$$

where $a_{l,m}$ is the intersecting length of the l-th ray path and the m-th block, s_m is the slowness perturbation of m-th block, and t_l is the travel time residual observation of the l-th ray path. Notice that each column in $\mathbf{A}$ contains the lengths of all the ray paths which penetrate the corresponding block. So the vertical partition of cube E can be mapped to a column partition on the system $\mathbf{As} = \mathbf{t}$,

$$\begin{bmatrix} a_{1,1} & a_{1,2} & 0 & 0 & a_{1,5} & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 \end{bmatrix}^T = \begin{bmatrix} t_{1,1} & + & t_{1,2} & + & t_{1,3} & + & t_{1,4} \end{bmatrix}$$

where $t_{l,m}$ is the partial travel time residual of the l-th ray path in the m-th block. The system can be expressed as,

$$[\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \mathbf{A}_4] \cdot [\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \mathbf{s}_4]^T = [\mathbf{t}_1 + \mathbf{t}_2 + \mathbf{t}_3 + \mathbf{t}_4]$$

where $\mathbf{A}_p$ is column partition of $\mathbf{A}$ corresponding to tomography partition p, $\mathbf{t}_p$ is the partial time residuals for $\mathbf{A}_p$, and $\mathbf{s}_p$ is the partial slowness perturbation model of the blocks in partition p. Then in each partition, one node is selected as the landlord, e.g., 3, 7, 12 and 14 are landlords in Figure 3.1. The landlord in partition p solves the subsystem $\mathbf{A}_p \cdot \mathbf{s}_p = \mathbf{t}_p$,

and the global tomography can be obtained by combining all $\mathbf{s}_p$.

Next is how to partition the travel time residual of each ray. Since the travel time residual is based on the observation of P-wave arrival time which usually contains noise, it is difficult to partition the travel time residual exactly for each partial ray. Here an approximation method is employed to derive the partial travel time residuals based on the reference model. For example, assume that the reference slowness model for E in Figure 3.1 is $\mathbf{s}^0 = [\hat{s}_1, \hat{s}_2, \hat{s}_3, \hat{s}_4, \hat{s}_5, \hat{s}_6, \hat{s}_7, \hat{s}_8]^T$. Let T_2 be the observed travel time of ray a_2 and the predicted travel time of a_2 is $T_2(0) = a_2 \cdot \mathbf{s}^0$, thus the travel time residual $t_2 = T_2 - T_2(0)$. Then the partial predicted travel time can be approximated by the reference slowness model,

$$T_{2,1}(0) = \vec{a}_{2,1} \cdot [\hat{s}_1, \hat{s}_2]^T \qquad T_{2,2}(0) = \vec{a}_{2,2} \cdot [\hat{s}_3, \hat{s}_4]^T$$
$$T_{2,3}(0) = \vec{a}_{2,3} \cdot [\hat{s}_5, \hat{s}_6]^T \qquad T_{2,4}(0) = \vec{a}_{2,4} \cdot [\hat{s}_7, \hat{s}_8]^T$$

where $T_{2,1}$ is the partial travel time of ray a_2 in partition E_1, $\vec{a}_{2,1}$ is the part of ray path a_2 in E_1. The travel time residual then is proportionally partitioned according to the predicted travel time,

$$t_{2,1} = t_2 \cdot \frac{T_{2,1}(0)}{T_2(0)} \qquad t_{2,2} = t_2 \cdot \frac{T_{2,2}(0)}{T_2(0)}$$
$$t_{2,3} = t_2 \cdot \frac{T_{2,3}(0)}{T_2(0)} \qquad t_{2,4} = t_2 \cdot \frac{T_{2,4}(0)}{T_2(0)}$$

Here we formalize the estimation of the partial travel time residuals. Let $\vec{a}_{l,p}$ be the partial ray path of the l-th ray in partition E_p and let $t_{l,p}$ be the corresponding partial travel time residual, $t_{l,p}$ can be estimated as,

$$t_{l,p} = t_l \cdot \frac{T_{l,p}(0)}{T_l(0)}$$

where $T_{l,p}(0)$ is the predicted travel time of l-th ray in partition E_p. Assume that $\hat{\mathbf{s}}_p(0)$ is

the partial slowness reference model of partition E_p then,

$$T_{l,p}(0) = \vec{a}_{l,p} \cdot \hat{\mathbf{s}}_p(0)$$

since $t_l = T_l - T_l(0)$ and $T_l(0) = a_l \cdot \mathbf{s}^0$, the partial travel time residual $t_{l,p}$ is,

$$t_{l,p} = T_l \cdot \frac{\vec{a}_{l,p} \cdot \hat{\mathbf{s}}_p(0)}{a_l \cdot \mathbf{s}^0} - \vec{a}_{l,p} \cdot \hat{\mathbf{s}}_p(0)$$

The previous discussion explains how to partition the tomography as well as $\mathbf{As} = \mathbf{t}$. Next we will show that how the partition and the computation distribution can be done in the network. First, after the ray tracing done, each node needs to send the partial ray path and the estimated partial travel time residual to the corresponding landlords for constructing the subsystems. Then each landlord can compute the partial slowness perturbation and broadcast it to the network so that each node can update its reference model part by part for future ray tracing. So there are two communication patterns in the network, unicast for sending partial rays and broadcast to synchronize the reference model. Both of them only happen once in one computation round.

3.2.2 Multi-resolution Evolving Tomography

In this section, we discuss the details about the multi-resolution evolving scheme and give the description of the proposed algorithm. Figure 3.2 illustrates how the multi-resolution evolving scheme works following the example in Figure 3.1. Notice that the computation of the partial travel time residuals highly depends on the slowness reference model. In the multi-resolution evolving scheme, the tomography model is not partitioned initially so that a good initial guess of the slowness model can be derived. This initial guess is used to estimate the partial travel time residuals later for approximating the optimal solution.

Suppose that the resolution of the tomography model is $d \times d \times d$ in the beginning, a single landlord (node 10 in the example) will compute the first perturbation for the reference model. Then the resolution increases to $2d \times 2d \times 2d$, and the tomography model is partitioned into

4 parts and distributed to 4 landlords (node 3, 7, 12 and 14) for computation as illustrated in Figure 3.2. Notice that, here the resolution of each partition is $d \times d \times 2d$. The aforementioned partition procedure will be recursively applied in each partition when sufficient more new earthquake events arrive, until the required resolution is achieved. Thus, at the $(r+1)$-th $(r = 0, 1, 2, \ldots)$ resolution, the tomography model has resolution $2^r d \times 2^r d \times 2^r d$ and is partitioned into 4^r parts and evenly distributed to $\max(N, 4^r)$ landlords.

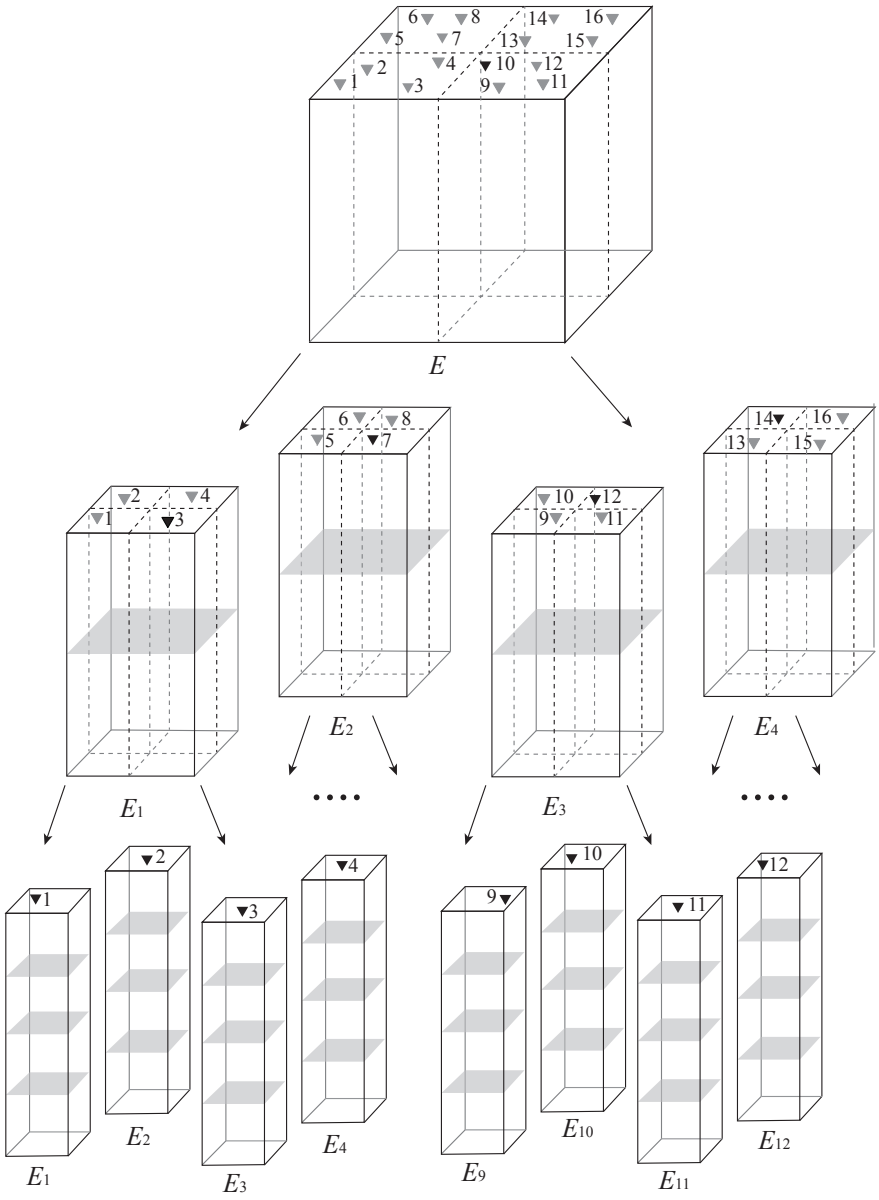


Figure 3.2 Multi-resolution Evolving Tomography.

Algorithm 1 Distributed Multi-resolution Evolving Tomography

Initialize

Node ID id, the resolution level q and dimension d
Landlord lists $\{\mathcal{H}_1, \mathcal{H}_2, ..., \mathcal{H}_q\}$
$r \leftarrow 0$, $Q_r \leftarrow d \times d \times d$, $P_r \leftarrow 1$
Slowness reference model $\mathbf{s}^0$ of resolution Q_r
Set landlord list to $\mathcal{H}_1$, $p' \leftarrow$ the partition index this node locates
if id is equal to $h_{p'}$
  $rc \leftarrow 0$, set the time out threshold T_{th}
endif

Repeat

Upon the detection of an event
  Trace the ray path a_l, compute $\vec{a}_{l,p}$ and $t_{l,p}$ for $1 \leq p \leq P_r$
  Send $\vec{a}_{l,p}$ and $t_{l,p}$ to landlord h_p.
Upon the reception of $\vec{a}_{l,p}$ and $t_{l,p}$
  if id is equal to h_p
    if rc is equal to 0
      $rc \leftarrow rc + 1$, start timer T_c
    else
      if $T_c < T_{th}$
        Add $\vec{a}_{l,p}$ and $t_{l,p}$ to $\mathbf{A}_{p'}\mathbf{s}_{p'} = \mathbf{t}_{p'}$
      else
        Solve least-squares problem $\mathbf{A}_{p'}\mathbf{s}_{p'} = \mathbf{t}_{p'}$
        Broadcast $\mathbf{s}_{p'}$ to all other nodes
        $rc \leftarrow 0$, reset T_c, clear system $\mathbf{A}_{p'}\mathbf{s}_{p'} = \mathbf{t}_{p'}$
      endif
    endif
  else
    Transfer $\vec{a}_{l,p}$ and $t_{l,p}$ to h_p
  endif
Upon the reception of $\mathbf{s}_p$ from landlord h_p
  Update the corresponding part of $\mathbf{s}^0$ with $\mathbf{s}_p$
  if all $\mathbf{s}_p (1 \leq p \leq P_r)$ have been received
    if $r + 1$ is equal to q
      TERMINATE
    else
      $r \leftarrow r + 1$, $Q \leftarrow 2^r d \times 2^r d \times 2^r d$, $P \leftarrow 4^r$
      Increase the resolution of $\mathbf{s}^0$ to Q
      Partition the tomography model into P_r parts
      Set landlord list to $\mathcal{H}_{r+1}$
      $p' \leftarrow$ the partition index this node locates
      if id is equal to $h_{p'}$
        $rc \leftarrow 0$, set time out threshold T_{th}
      endif
    endif
  endif

Now we give the formal description of the Distributed Multi-resolution Evolving Tomography (DMET) algorithm, see Algorithm 1. Suppose that there are q different resolution levels in our multi-resolution scheme, let d be the initial resolution dimension.

At the $(r+1)$-th resolution, the tomography model E with resolution Q_r is vertically and evenly partitioned into P_r different parts, then the p-th part E_p contains Q_r/P_r blocks ($Q_r = 2^r d \times 2^r d \times 2^r d$ and $P_r = 4^r$). The system $\mathbf{As} = \mathbf{t}$ can be partitioned by columns accordingly,

$$[\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_{P_r}] \cdot [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{P_r}]^T = [\mathbf{t}_1 + \mathbf{t}_2 + \ldots + \mathbf{t}_{P_r}]$$

and for E_p the following subsystem is constructed on a landlord,

$$\mathbf{A}_p \mathbf{s}_p = \mathbf{t}_p$$

where $1 \leq p \leq P_r$.

Initialize line 1-4: Each node initialize its ID, the resolution level q and initial dimension d. Besides, each node initializes a landlord list $\{\mathcal{H}_1, \mathcal{H}_2, ..., \mathcal{H}_q\}$ where $\mathcal{H}_{r+1}$ is a node list $\{h_p | 1 \leq p \leq P_r\}$ and h_p indicates the landlord for partition p at the $(r+1)$-th resolution. This tells each node where to send the partial ray information. Then the resolution and partition parameters and a slowness reference model for ray tracing are initialized.

Initialize line 5-8: Set the landlord list as $\mathcal{H}_1$ (only one landlord in it). If this node is the landlord, it also initializes a ray counter and a time out threshold T_{th} which controls the waiting time for a landlord to receive partial ray information from other nodes. Because it is hard to know how many partial rays will be received by the landlord since the event activity is unpredictable.

Repeat line 1-3: After the initialization, each node will act based on the event detection and message reception. Once an event is detected by a node, either a landlord or a common node, it will trace the ray path (as assumed in previous discussion, the event location has already been estimated in the network and every node knows it). Then the node computes and sends the partial ray information to the corresponding landlords according

the landlord list $\mathcal{H}_{r+1}$.

Repeat line 4-19: If the node is a landlord in partition p', when it receives the first partial ray for current resolution, the node will start a timer T_c. Otherwise, if the timer is less than the threshold T_{th}, the landlord will add the received partial ray into the subsystem $\mathbf{A}_{p'}\mathbf{s}_{p'} = \mathbf{t}_{p'}$. Once the timer T_c is out, the node will compute the partial slowness perturbation $\mathbf{s}_{p'}$ from constructed $\mathbf{A}_{p'}\mathbf{s}_{p'} = \mathbf{t}_{p'}$. Then the landlord broadcasts $\mathbf{s}_{p'}$ to all other nodes and reset the parameters. If the node is not a landlord, it will transfer the received partial ray information to the corresponding landlord.

Repeat line 20-35: Once a node receives the slowness perturbation $\mathbf{s}_p$ from landlord h_p. It will update the corresponding part of the slowness reference model $\mathbf{s}^0$ with $\mathbf{s}_p$. After the partial slowness perturbations from all landlords have been received, i.e., the entire slowness reference model has been updated. The algorithm will TERMINATE if the required resolution is met; otherwise, the node will set r to $r + 1$, calculate current Q_r and P_r, and increase the resolution of reference model $\mathbf{s}^0$ to Q_r. There are different ways to increase the resolution of the model, e.g., all the blocks in higher resolution just use the slowness value of the block it belongs to in the lower resolution. Then each node will partition the model into P_r parts. Also, the node will set the appropriate landlord list. If the node is a landlord at $(r + 1)$-th resolution, it will initial a ray counter and set a timeout threshold (the threshold is larger in higher resolution since more rays are needed for higher resolution tomography computation).

Notice that the algorithm here is based on a cube tomography model, in reality the model is not always a cube and the partition may depend on the deployment of the sensor network. This algorithm can be applied to different models, it only needs to change the resolution evolving and partition scheme, i.e., how to set Q_r and P_r in resolution level r.

3.3 Evaluation and Validation

We implemented and evaluated the algorithm with our InsightTomo emulation system, we will discuss more details about this in next chapter, here we only validate that the

proposed algorithm not only balances the computation load, but also achieves low communication cost and good data loss tolerance.

3.3.1 Experiment Setup and Implementation

Our evaluation uses synthetic magma and earthquake events data. First, a data generator is implemented to generate a magma area and earthquake events. Assume that 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 Figure 3.3. The velocities of seismic waves inside and outside the magma area are V and $0.9V$ where V is 4.5km/s which is a typical P-wave velocity.

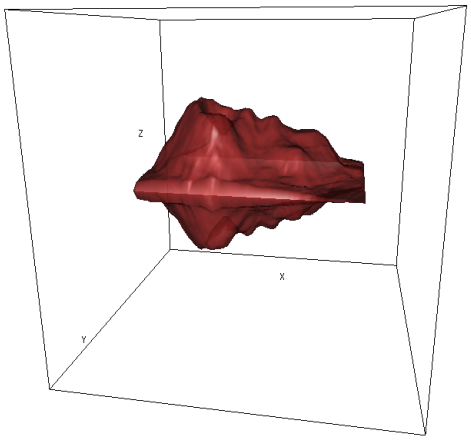


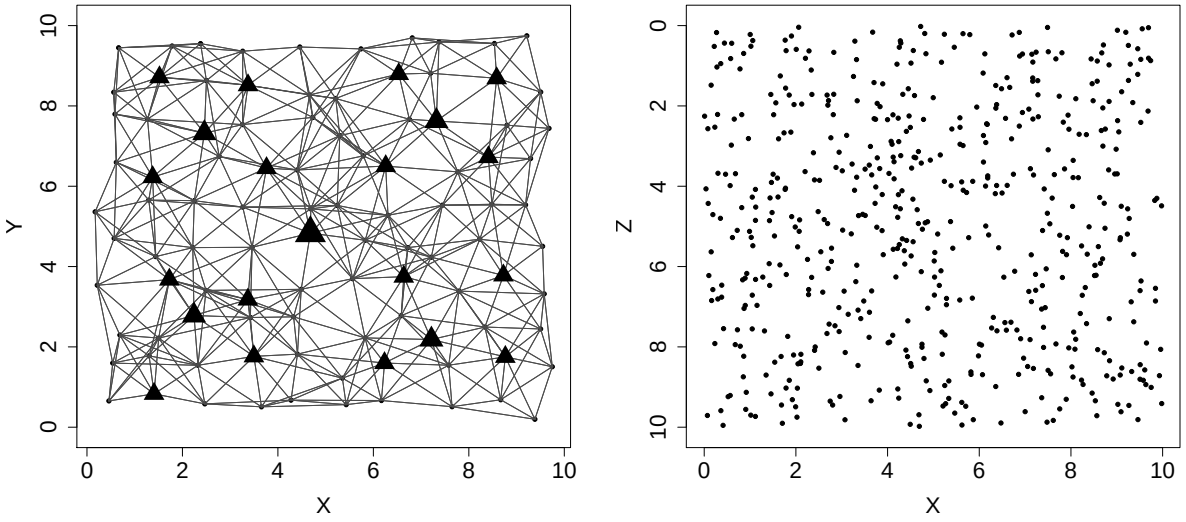
Figure 3.3 3D Model in the Simulation.

In InsightTomo emulator, a network of 100 nodes is deployed on the top of the monitoring area. We set the target tomography resolution to be $32 \times 32 \times 32 = 32768$ where each block is of size 0.3215 km^3. The data generator then generates earthquake events with random location and time, and calculates ray travel times from event locations to all sensor nodes. To simulate the event location estimation and ray tracing errors, a White Gaussian Noise is added to the travel time to generate the sensor node observation (arrival times).

Each node can calculate the predicted travel time based on the initial model in different resolutions.

Notice that based on the predefined slowness model for the cubic area, the generator is supposed to calculate the exact travel time for each ray. But the magma area itself is a discrete model and the surface shown in Figure 3.3 is constructed from a discrete data set. So the generator partitions the cube area with a much higher resolution $128 \times 128 \times 128$ as the ground truth and assigns the slowness value to the blocks according whether their center points are in the magma area or not.

100 stations are randomly distributed on top of the cube and form a mesh network as shown in Figure 3.4(a) (the black triangles indicate the landlords in different resolution levels). 650 events are similarly distributed, Figure 3.4(b) which shows the vertical view of the event sources distribution.



(a) Network Topology (b) Event Sources

Figure 3.4 Stations and Events Distribution

We evaluate the algorithm starting with resolution 8^3 and one partition, evolving to resolution 16^3 with 4 partitions and complete at resolution 32^3 with 16 partitions. 50, 100 and 400 events are generated for resolution from low to high to make sure the system is

overdetermined. To simulate the event location estimation and ray tracing errors, a White Gaussian Noise is added to the travel time to generate the sensor node observations (arrival times).

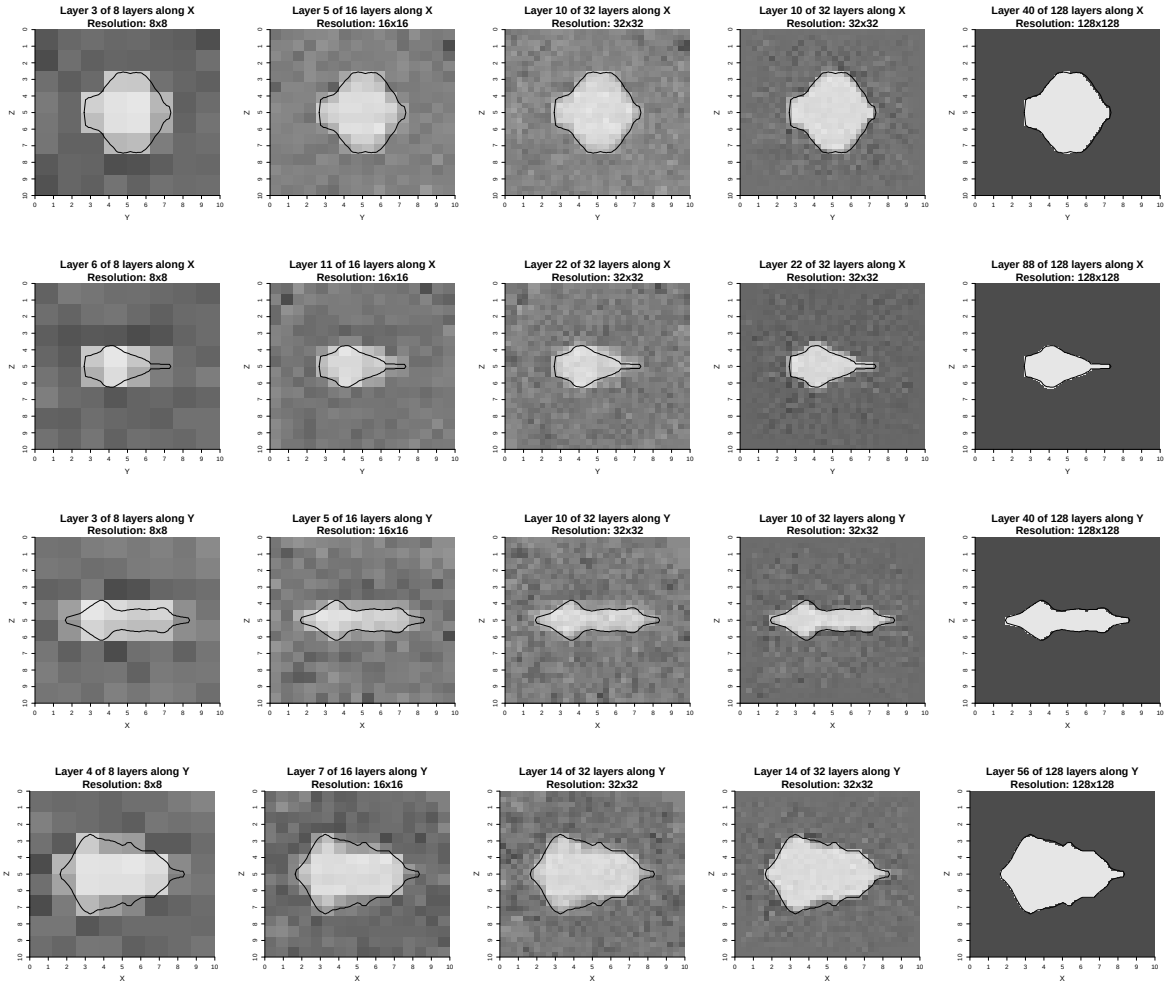
To simulate the sensing behavior of the nodes in the network, two subnetworks are built in the emulator. One is the wireless mesh network in Figure 3.4(a) with a link and physical layer connectivity model. The other is a simple wired network where one generator node wired connects with each node in the mesh network. This generator node will generate event at random time, compute the travel times from this event to each node based on the ground truth, and send the event location and travel time to corresponding node with noise (suppose that the event location has been done as we discussed above). So the nodes in mesh network are blind to the event activity and thus simulate the sensing behaviour.

The Bayesian version of ART method [58] is used in the experiments to compute the tomography. We use the relative update of the estimation between two sweeps (one sweep means that all the ray paths in the system are used once for estimation update) as the stopping criteria. Besides, a centralized collection scheme (one node collects all the ray information and perform centralized computation) is implemented in the emulator with the same data set to compare with our DMET algorithm. Notice that the Bayesian ART method solves the system $\mathbf{As} = \mathbf{t}$ to minimize $\| \mathbf{t} - \mathbf{As} \|^2 + \lambda^2 \| \mathbf{s} \|^2$ where λ is the trade-off parameter that regulates the relative importance we assign to models that predict the data versus models that have a characteristic, a priori variance.

3.3.2 Correctness and Accuracy

To validate the correctness and accuracy of the algorithm, we first visualize the tomography result in vertical slices in Figure 3.5. Each row of figures shows the same tomography slice on corresponding layer along with X or Y axes (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 magma chamber surface. We can see that at the lowest resolution 8^3, the result can hardly indicate the outline of the magma chamber since the block size is big, especially for

the small cross section in the first row. But it gives a good start point for the higher resolution to further refine the result. At resolution dimension 16, the result can closely show the outline of the magma chamber. The result in column (c) at resolution 32^3 is already very close to the centralized solution in column (d).



(a) Partition Res: $8 \times 8 \times 8$ (b) Partition Res: $16 \times 16 \times 16$ (c) Partition Res: $32 \times 32 \times 32$ (d) Centralize Res: $32 \times 32 \times 32$ (e) Ground Truth Res: $128 \times 128 \times 128$

Figure 3.5 2D Tomography Rendering.

Using $\tilde{\mathbf{s}}$, $\mathbf{s}^*$ and $\bar{\mathbf{s}}$ to represent the synthetic model, the reconstructed model and the mean value of $\mathbf{s}^*$ respectively, we use the following quantitative measures of distance from

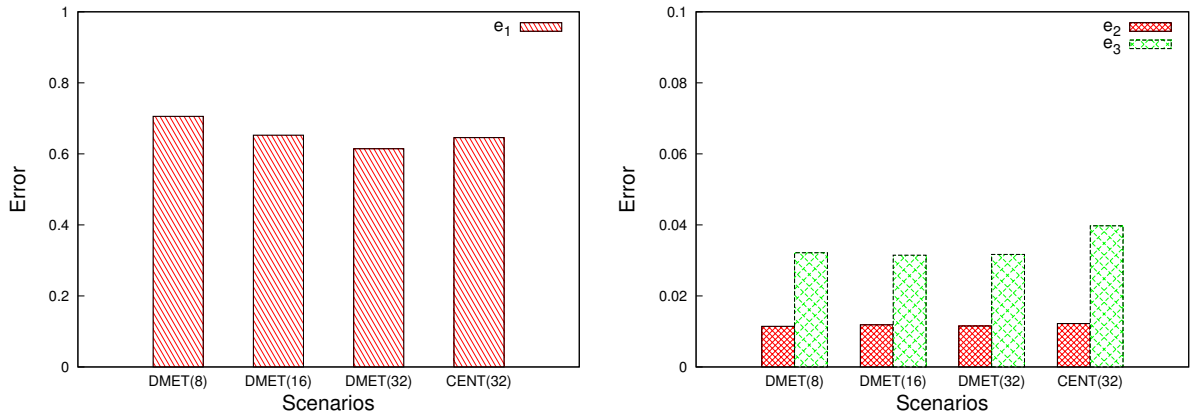


Figure 3.6 Measures of Distance from Synthetic Model.

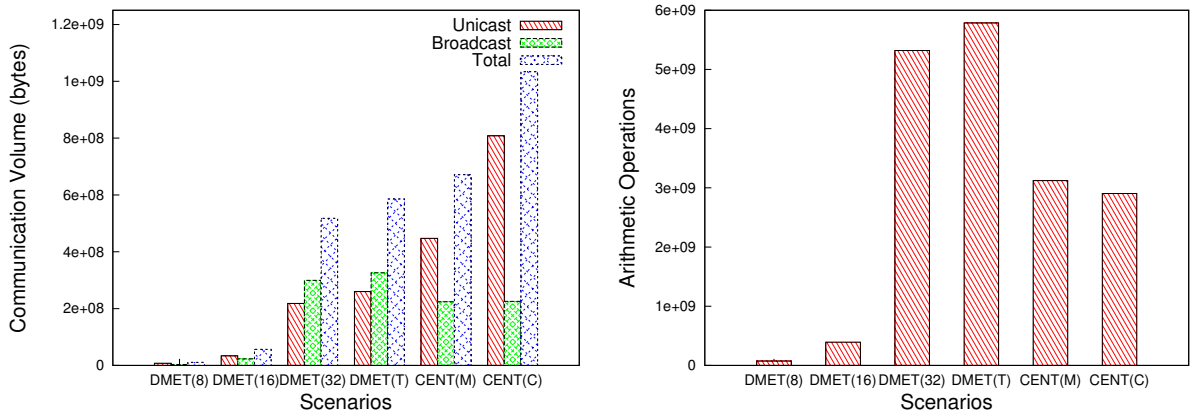
the synthetic model provided in [26] to evaluate the estimation quality,

$$
\begin{aligned}
e_1 &= (\sum_{i=1}^{n}(\tilde{\mathbf{s}}_i - \mathbf{s}_i^*)^2 / \sum_{i=1}^{n}(\mathbf{s}_i^* - \bar{\mathbf{s}})^2)^{\frac{1}{2}} \\
e_2 &= \sum_{i=1}^{n} |\tilde{\mathbf{s}}_i - \mathbf{s}_i^*| / \sum_{i=1}^{n} |\mathbf{s}^*| \\
e_3 &= \max |\tilde{\mathbf{s}}_i - \mathbf{s}_i^*|
\end{aligned}
$$

These represent the normalized root mean squared distance, the average absolute value distance and the worst case distance respectively. The result is shown in Figure 3.6, DMET(8) means that the distance analysis of DMET algorithm with resolution 8^3 and CENT(32) indicates the result of centralized algorithm. First we observe that in DMET algorithm, the distances are decreasing along with the increase of the resolution. The distances in DMET(32) are even smaller than CENT(32), this is because that we use the relative update as the stop criteria in Bayesian ART method and the centralized algorithm may stop before the distance is small enough. This analysis can imply that the multi-resolution evolving scheme can give a good approximation on each resolution level for estimating partial travel times and the computation can approximate the centralized solution while not diverging in the local computation in each partition.

3.3.3 Communication and Computation

From the experiment result and analysis, it is validated that DMET gives a good result to approximate the tomography compared with the centralized algorithm. Next, we will compare the communication and computation cost between DMET and the centralized algorithms. Since the ray tracing algorithm is very expensive to perform, we assume that each node does ray tracing itself and sends the indexed ray path to a base station (centralized) or to the corresponding landlords. Also, after centralized computation done on the base station, it will broadcast the model to the network for future ray tracing since the whole tomography process is iterative.

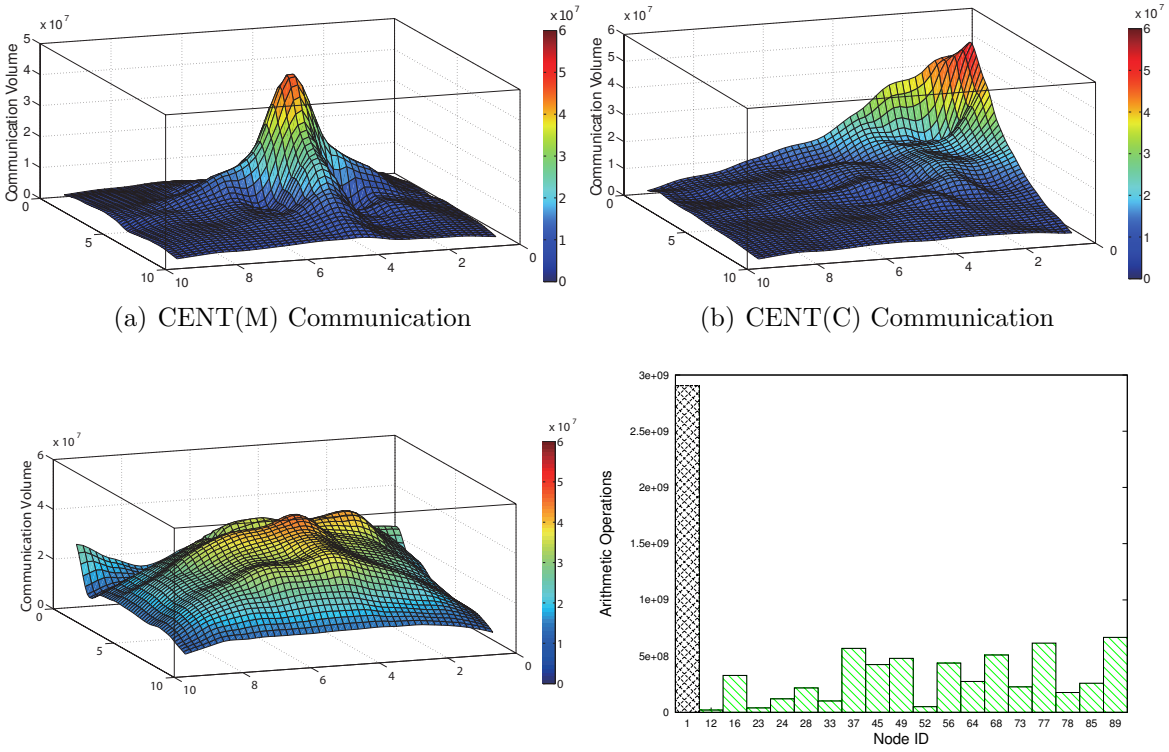


(a) Communication (b) Computation

Figure 3.7 Communication and Computation Cost Analysis.

Figure 3.7 gives the communication and computation cost analysis where DMET(T) is the total cost for DMET algorithm, CENT(M) and CENT(C) are the cost of centralized algorithm when the base station is in the center and the corner of the network respectively. In Figure 3.7(a), we can see that the total unicast cost of DMET is much less than centralized algorithm (about one third of CENT(C)) since the centralized data collection will cause more interference and congestion when the data rate is high. At the same time, the broadcast cost of DMET is more than centralized algorithm because multiple nodes need to broadcast in DMET. By counting the arithmetic operations in the computation of the base station and

landlords, Figure 3.7(c) gives the computation cost for centralized algorithm and DMET. The total cost of DMET is higher than centralized since we partition and distribute the system to landlords, and the computation on landlords are entirely local and it is lack of the global information to constraint the problem for fast convergence.



(a) CENT(M) Communication

(b) CENT(C) Communication

(c) DMET Communication

(d) Computation

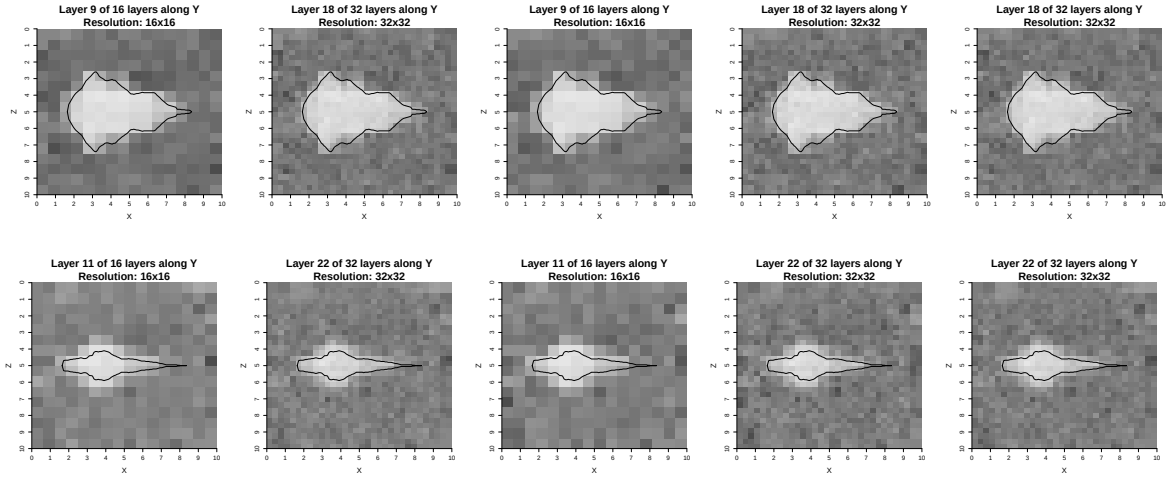
Figure 3.8 Communication and Computation Load Balance.

The experiment results above have shown that the total communication cost of DMET is less than the centralized algorithm while the total computation cost is higher. Next, we count the communication and computation cost on each node in the network to show that DMET balances the communication and distribute the computation load in the network. Figure 3.8(a), (b) and (c) visualize the communication cost on each node in the network for 3 different scenarios with heat maps. From Figure 3.8(a) and (b), we can see that the communication cost in centralized scenario for the nodes near the base station (either in the center or corner) are much higher than other nodes since all the messages need to be delivered

through them. The communication cost on each node in our algorithm shown in Figure 3.8(c) is more balanced. In Figure 3.8(d), the first column is the computation load on node 1 for CENT(C) algorithm, all other columns are the computation load on corresponding landlords of DMET algorithm. We can see that although the total computation cost of DMET is higher but the computation load is much more balanced.

3.3.4 Data Loss Tolerance and Robustness

We did an evaluation on the robustness of DMET. The algorithm runs with the same data set and two different packet loss ratios of 10% and 40% set in the emulator. Figure 3.9 gives part of the 2D slice rendering along Y axes with packet loss. From Figure 3.9, we can see that with 10% or even 40% packet loss, compared to the result without packet loss there is not much difference in terms of the magma area outline. This validate the robustness of DMET algorithm which can be tolerant to a severe packet loss.



(a) 10% packet loss Res: 16×16×16 (b) 10% packet loss Res: 32×32×32 (c) 40% packet loss Res: 16×16×16 (d) 40% packet loss Res: 32×32×32 (e) No packet loss Res: 32×32×32

Figure 3.9 2D Tomography Rendering with Data Loss.

PART 4

END-TO-END SYSTEM DESIGN AND EMULATION

In traditional seismology, the raw seismic data is collected for manual analysis including P-wave arrival time picking on the seismograms. Then centralized methods will process the data and compute seismic tomography. Keep in mind that our goal is to design a system which can deliver 3D tomography in real-time over a large-scale sensor network by utilizing the limited communication ability in the network and the computation power on the sensor node. To reach this goal, first no raw seismic data should be transmitted over the network, which requires a light weighted algorithm that can accurately pick the P-wave arrival time on the sensor nodes locally inside the network. Second, an efficient distributed tomography computation method is needed for processing data and inverting volcano tomography in the network while avoiding both costly data collections and centralized computations.

In this chapter we present InsightTomo - an in-situ seismic tomographic imaging system framework in sensor network. The design of InsightTomo consists of a series of algorithms and network design to automatically process the seismic data, pick the P-wave arrival time, identify seismic events and compute the seismic tomography in-situ in real-time. The system design is evaluated with real data from the San Andreas Fault (SAF) on Parkfield.

4.1 System Overview

4.1.1 System Model

The mesh network architecture is employed in the design of InsightTomo. Each sensor node in the network is equipped with a seismic sensor (e.g., single or three component geophone) that continuously samples and records the signal in an external storage (e.g., SD memory card). Also, the sensor node has a low power MCU (e.g., MSP-430 or Imote series) that has limited computation resources but keeps a very low power profile. Besides, all

the sensor nodes have GPS modules on board [1] such that the clocks on sensor nodes are time-synchronized so are the time pickings. A powerful computation unit (e.g., BeagleBone Black board, cellphone or tablet) is also installed on each node; the unit can complete the computation-intensive tasks including the event location and tomography inversion.

4.1.2 System Architecture

In this section, we will give an overview of the system architecture and the data flow of InsightTomo respect to the design requirements mentioned above. InsightTomo consists of several algorithms running on sensor nodes and coordinator nodes, and a bundle layer protocol is proposed to improve the performance of communication. Figure 4.1 illustrates the architecture and dataflow of InsightTomo system. There are three steps in the architecture corresponding to the workflow of seismic tomography.

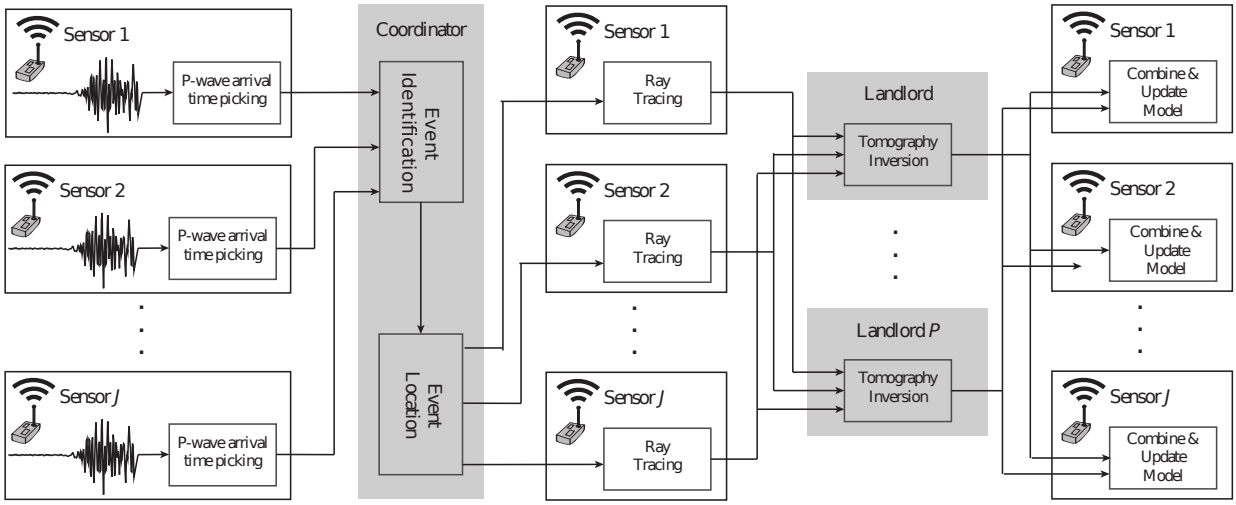


Figure 4.1 The Architecture of InsightTomo System.

(1) Sensor node receives the signal from the sensor and analyze the signal with P-wave arrival time picking algorithm. The algorithm will continuously monitor the samplings and alarm if an event is detected, then the picking algorithm will pick the P-wave arrival time. After the arrival time picking done, the sensor node only needs to send the arrival time to coordinator node. Refer to section 4.2.1 for more details.

(2) The coordinator (could be any sensor node) receives the P-wave arrival times from sensor nodes that detected events. The only information received by the coordinator node is a series of arrival time pickings. Based on these pickings, the coordinator first needs to identify which arrival times are corresponding to the same event. The reason is that not all the sensor nodes can detect one specific event due to the event intensity, event position, sensor instability and so on. After the event is identified, the coordinator node can compute event location and origin time, then send the event location and origin back to the corresponding nodes that detected it. Refer to section 4.2.2 for more details.

(3) This step consists of several in-network computation and communication tasks. We put them together since they are all parts of the distributed tomography computation mechanism. Once the sensor node receives one event location from the coordinator node, it can trace the ray path from the event location to itself. Based on our distributed tomography algorithm, the sensor node sends ray paths to corresponding landlord node that will be in charge of the tomography computation. After the computation done, each landlord will broadcast the partial tomography model to the network so that each sensor node can update its model for future ray tracing and computation. Refer to section 4.2.3 for more details.

4.2 System Design

In this section, we study the design of InsightTomo step by step following the discussion above. Our design is based on careful analysis of the real data from previous Parkfield deployment.

4.2.1 P-wave arrival time picking

Primary waves (P-waves) are the seismic waves that travel faster than any other waves through the earth. P-waves arrive at the seismic sensors first and the arrival time of P-waves are essential to the first-arrival traveltime tomography. Figure 4.2 shows the seismograms from four seismometers deployed in Parkfield when an event happens. The vertical lines represent manual pickings of the P-wave arrival times. Due to the different wave propagation

delays, the P-wave arrival times on sensors are different. In local seismic tomography, the scale of the field is up to tens of kilometres and the maximum difference of the P-wave arrival times among sensors is about several seconds, so that the accuracy of the picking is significant. Besides, manual analysis of seismograms and picking of arrival times require post processing of the data and are very time consuming, especially in a large sensor network. To avoid raw seismic data transmission and meet the real-time requirements, InsightTomo demands an on-line automatic event detection and P-wave arrival time picking method that runs on each sensor node.

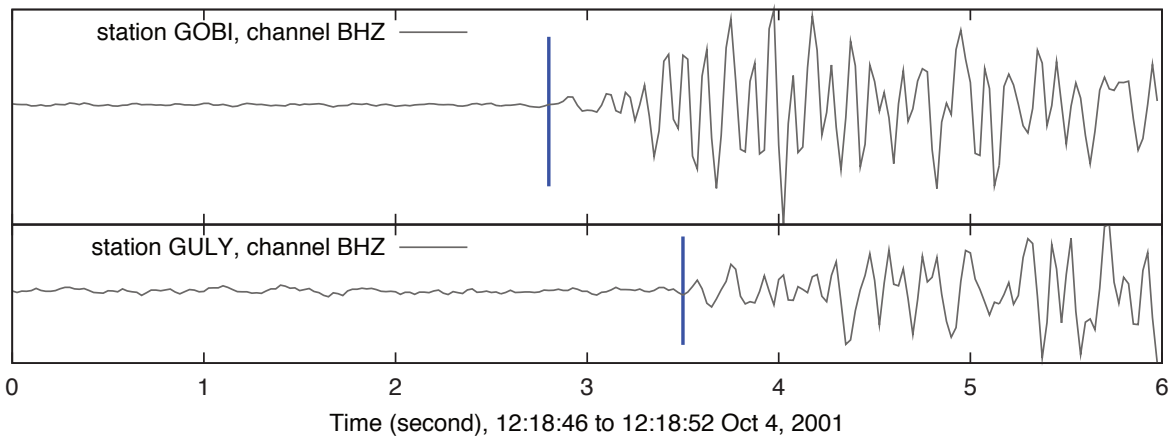


Figure 4.2 The seismogram from BHZ channel of four seismometers in Parkfield when an event happens. The vertical lines indicate the manual pickings of P-wave arrival times.

From the seismograms in Figure 4.2, one can see that there is a big difference on the amplitude of the signal before and after the arrival of P-waves, the P-wave arrival time is a change point of the variance of the signal amplitude. Based on this observation, a method is proposed here by utilizing the maximum-likelihood (ML) estimation to estimate the variance of the signal amplitude following a statistical model. In the following discussion, we use pre- and post-change to describe the signals before and after the P-wave arrival.

Without loss of generality, we assume that both the pre- and post-change signals follow a normal distribution but with different variances. Let $\{x_i\}_{i=1}^{t}$ be the continuous sequence of samples from 1 to t, then the pre- and post-change sample has a normal distribution with zero mean respectively,

$$\text{pre-change sample,} \quad x_i \sim \mathcal{N}(0, \sigma_1^2) \tag{4.1}$$

$$\text{post-change sample,} \quad x_i \sim \mathcal{N}(0, \sigma_2^2) \tag{4.2}$$

The logarithm of the likelihood function at time k is,

$$\mathcal{L} = \sum_{i=k+1}^{t} \ln \frac{f_2(x_i)}{f_1(x_i)} \tag{4.3}$$

where $f_1(x_i)$ and $f_2(x_i)$ are the probability density functions (pdf) of the pre- and post-change signals, the likelihood function then can be rewritten as,

$$\mathcal{L} = \sum_{i=k+1}^{t} \ln \frac{\frac{1}{\sqrt{2\pi\sigma_2^2}} e^{-\frac{x_i^2}{2\sigma_2^2}}}{\frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{x_i^2}{2\sigma_1^2}}} \tag{4.4}$$

$$= \sum_{i=k+1}^{t} \left[\frac{1}{2} \ln \frac{\sigma_1^2}{\sigma_2^2} - \frac{x_i^2}{2} \left(\frac{1}{\sigma_2^2} - \frac{1}{\sigma_1^2} \right) \right] \tag{4.5}$$

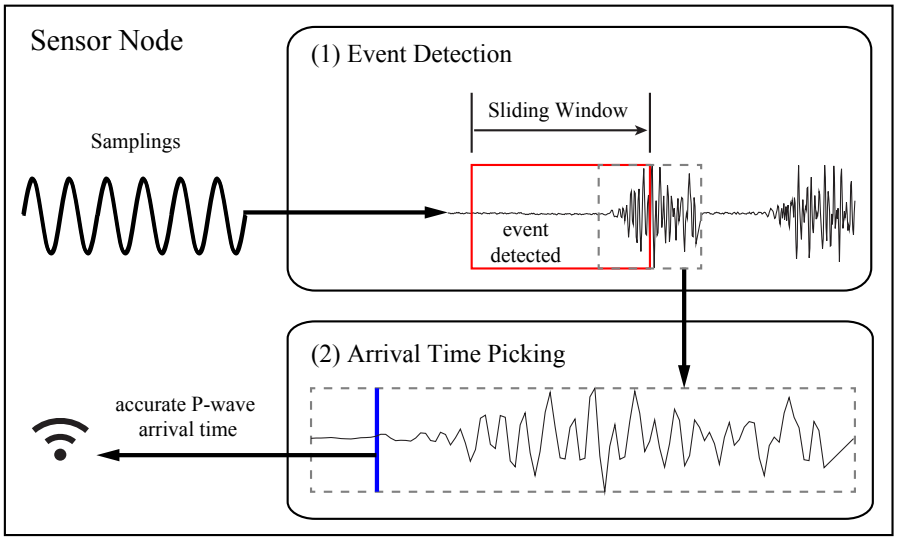


Figure 4.3 Two Step P-wave Arrival Time Picking.

Figure 4.3 illustrates how the proposed method works. The method consists of two steps, (1) Event Detection which continuously scanning the samplings from the sensor with a sliding window, claims if there is an event (change point) happens and extract a segment of signals around the change point; (2) Arrival Time Picking which takes the segment of signals from step (1), picks the exact change point (arrival time) from it and send the P-wave arrival time to coordinator node.

Event Detection The goal of event detection is to continuously check weather there is a change point in the signal stream that is probably an event. The STA/LTA (short-term average over long-term average) algorithm [22] [1] is employed for event detection here because it is fast to monitor the signal and roughly find where an event happens. To describe STA/LTA algorithm, we need to first introduce the concept of RSAM (Realtime Seismic Amplitude Measurement), which is widely used in seismology. The RSAM is calculated on raw seismic data samples per second. Assume the sampling rate of the signal is m (samples per second), let $\{x_t, \cdots, x_{t+m-1}\}$ and $\{x_{t-m}, \cdots, x_{t-1}\}$ be the samples in the i-th and $(i-1)$-th second respectively, then $e_{i-1} = \frac{\sum_{j=t-m}^{t-1} x_j}{m}$ is the average of the $(i-1)$-th second. The i-th second RSAM r_i is calculated as,

$$r_i = \frac{\sum_{j=t}^{t+m-1}(x_j - e_{i-1})}{m} \tag{4.6}$$

In our system, the STA or LTA is continuously updated based on,

$$X_i = \frac{\sum_{j=0}^{n-1} r_{i-j}}{n} \tag{4.7}$$

where r_i is i-th second RSAM; n is the STA or LTA time window size (in seconds).

The ratio of STA over LTA is continuously monitored. Once the ratio exceeds the threshold, an event is detected. Algorithm 2 gives the description of the event detection method. A sliding LTA window with a STA window keep moving second by second and calculating the STA/LTA ratio. If the threshold b is reached, a change point is detected

Algorithm 2 Event detection algorithm

1: $detected \leftarrow$ false.
2: Set LTA window size $ltaw$.
3: Set STA window size $staw$.
4: Set STA/LTA ratio threshold b.
5: **for** $t \leftarrow w, \ldots$ **do**
6: Calculate RSAM value r_{t-1} of second $t-1$.
7: Calculate $sta \leftarrow \frac{\sum_{i=t-1-staw}^{t-1} r_i}{staw}$.
8: Calculate $lta \leftarrow \frac{\sum_{i=t-1-ltaw}^{t-1} r_i}{ltaw}$.
9: Calculate STA/LTA ratio $ratio \leftarrow \frac{sta}{lta}$.
10: **if** $ratio > b$ **then**
11: **if** *detected* is false **then**
12: $detected \leftarrow$ true
13: $T \leftarrow t$
14: Run algorithm 3 to pick arrival in $[T-a, T+b]$
15: **else**
16: $detected \leftarrow$ false
17: **end if**
18: **else**
19: $detected \leftarrow$ false
20: **end if**
21: **end for**

at T, the signal in the window of $[T-a, T+b]$ are extracted and the arrival time picking algorithm will pick the accurate arrival time from it. Then the sliding window continue moving from T and calculating the STA/LTA ratio, since the event usually lasts for a period of time, the STA/LTA ratio will be over the threshold for a while until time T'. In our implementation, the STA and LTA window are 1 and 4 seconds respectively; the signal window for picking is 3 seconds where $a = 1$ and $b = 2$. This setting with threshold 2 can perform event detection very well for picking as shown in section 4.4.1. Figure 4.4 gives an example of event detection result and the detection length of a real earthquake event.

Note that the T claimed here is only a change point but not necessarily an earthquake event detection. Sometimes there might be a voltage spike from the sensor that can cause a change point detection, see Figure 4.5. Since the spikes have much larger amplitudes than events, to avoid the false alarm on spikes an upper threshold is set for $\mathcal{S}$. If the value of $\mathcal{S}$

is over the threshold the event detection will skip this change point.

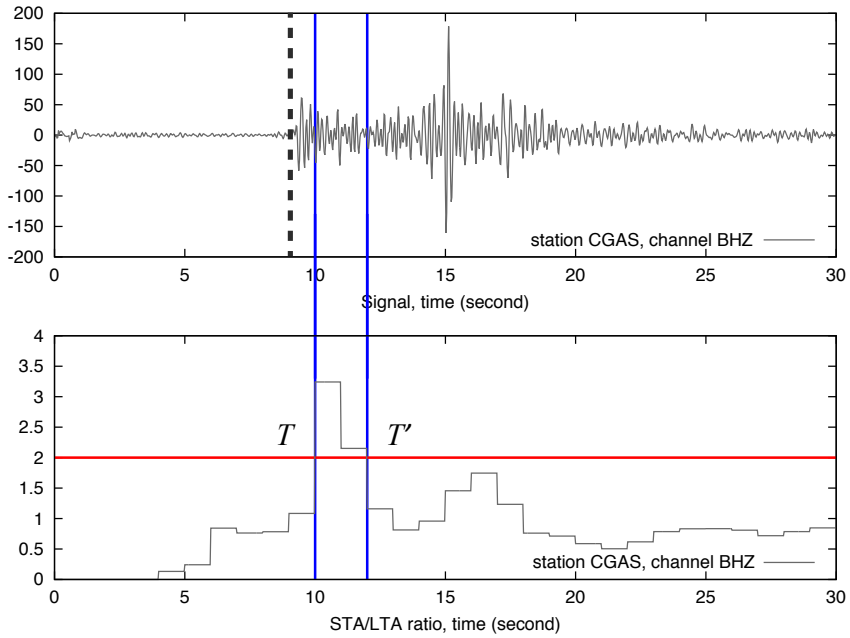


Figure 4.4 Event detection example on an Earthquake Event during 17:39:20 to 17:39:50 Feb 7, 2002.

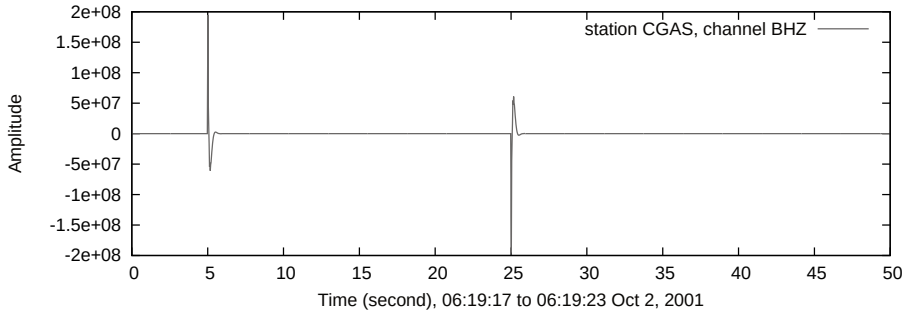


Figure 4.5 Spikes that causes change point detection.

Arrival Time Picking After the event detection done, the arrival time picking algorithm will find the exact change point k^* in $[T-a, T+b]$ which maximize the function value of $\mathcal{L}$ as discussed above. The problem can be simply formulated as,

$$k^* = \arg\max_k \mathcal{L} \tag{4.8}$$

From the definition of statistics for ML change point detection, we can see that σ_1 and σ_2 are required to calculate the value of $\mathcal{L}$. Recall that σ_1^2 is the variance of the pre-change samples in the signal, which can be considered as the noise level of the signal. This noise level can be different for each sensor due to manufacturing, temperature and so on. So every sensor needs to calculate its own σ_1^2 using some samples generated from it while no event happens. Let z_i be such a sample at time i, σ_1^2 can be calculated as,

$$\sigma_1^2 = \frac{1}{n-1}\sum_{i=1}^{n}(z_i - \bar{z})^2 \tag{4.9}$$

where n is the number of samples and $\bar{z}$ is the mean value of these n samples. Since the noise level of one sensor usually does not change much, the proposed method uses the σ_1^2 for a fixed period of time, e.g., one day and then update once.

Since σ_2^2 represents the variance of the post-change samples, the value of σ_2^2 depends highly on the property of event. This imposes that for each k, a σ_2^2 needs to be derived to maximize $\mathcal{L}$. This follows that $\partial\mathcal{L}/\partial\sigma_2^2$,

$$\frac{\partial \sum_{i=k+1}^{t}\left[\frac{1}{2}\ln\sigma_1^2 - \frac{1}{2}\ln\sigma_2^2 - \frac{x_i^2}{2}\left(\frac{1}{\sigma_2^2} - \frac{1}{\sigma_1^2}\right)\right]}{\partial\sigma_2^2} = 0 \tag{4.10}$$

and the σ_2^2 can be expressed as,

$$\sigma_2^2 = \frac{\sum_{i=k+1}^{t} x_i^2}{t-k} \tag{4.11}$$

The corresponding algorithm is described in algorithm 3. The vertical dashed line in Figure 4.4 indicate the arrival time picking of the algorithm for that detected event.

4.2.2 Event Location

Based on the architecture of InsightTomo, if a sensor node detects an event and picks the P-wave arrival time, it will send the time picking to the sensor node that acts as the coordinator for event location. The coordinator node only receives the time pickings from the sensor nodes and has the knowledge that which picking is from which node. In all of

Algorithm 3 Arrival timing picking algorithm

1: Calculate σ_1^2
2: $l^* \leftarrow 0, k^* \leftarrow 0$.
3: **for** $k = T - a + 1, \ldots, T + b - 1$ **do**
4: Calculate σ_2^2 and $\mathcal{L}$, $l = \mathcal{L}$.
5: **if** $l > l^*$ **then**
6: $l^* \leftarrow l, k^* \leftarrow k$
7: **end if**
8: **end for**
9: **return** k^*

these pickings, there might be false alarms or some small and remote event is only detected by few stations. As we known, to estimate an event location, at least three pickings from different sensor nodes are required, the event detected only by one or two nodes is impossible to be located. Also, more pickings from different sensor nodes for one event usually lead to a better estimation. Thus there are two steps in event location, (1) Event Identification where the coordinator node identifies how many events existing in a series of arrival time pickings received and which pickings belong to the same event; (2) Location Estimation which uses Geiger's method to estimate the event location from the pickings of that event.

Event Identification Since the InsightTomo system focuses on local seismic tomography (contrast to global tomography which focuses on spherical earth), the maximum arrival time difference among the sensor nodes is about several seconds. The event identification is based on two rules, (1) The maximum difference of the time pickings from the *same* event should be less than a threshold β; (2) The number of time pickings from the *same* event should be over a threshold θ.

Suppose that the coordinator node receives a series of time pickings and puts them in a list C. Each item in C is a pair $\langle$picking,node$\rangle$ which represent the arrival picking and the node picked it. The coordinator sorts list C increasingly according to pickings and gets C'. Then the coordinator will find the continuous subsequences in C' where the maximum difference among the items in the same subsequence is less than β, if the length of some subsequence is greater than θ the items in this subsequence are from the same event. Event

identification will feed all the satisfied subsequences to step (2) that will estimate the event locations. Figure 4.6 gives an example of the event identification based on the 629 pickings of the P-wave arrival time picking method in section 4.2.1. In this example, the x-axis is the arrival time picking index for sorted list C' and the y-axis is the time difference between two adjacent pickings. Four events are identified (circles in the figure, $\beta = 4.0$ and $\theta = 14$) from this data while three of them are identified in the manual pickings.

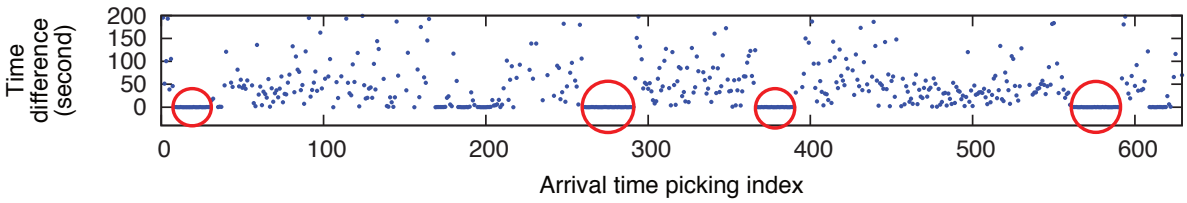


Figure 4.6 Sorted arrival time pickings from 30 stations with the entire day samplings on Feb 7, 2002 in Parkfield.

Location Estimation The arrival time pickings in seismology are also called observed arrival times. To estimate the event location, an initial reference model of the subsurface P-wave velocities is required which can be used to calculate arrival time based on the guess of event location, this arrival time is referred to predicted arrival time. In this work, we adopt 1D velocity model which treats the subsurface as a layered model and is commonly used in routine earthquake location.

For sensor node v, we designate the observed arrival times by τ_v and the predicted arrival time by t_v which is a function of the estimated location x, y, z and the origin time q for the event. The residual, or the difference between the observed arrivals and the predicted arrivals is $r_v = \tau_v - \chi(q, x, y, z)$. Then an approximate location is estimated by seeking a small perturbation in the location that makes the residual smaller.

The guess, or estimation, is designated as, $\mathbf{x}_0 = \{q_0, x_0, y_0, z_0\}$ and let $\Delta\mathbf{x} = \{\delta q, \delta x, \delta y, \delta z\}$ be the small adjustments to the guess that bring X_0 closer to the correct location. The new location is represented as $\mathbf{x} = \mathbf{x}_0 + \Delta\mathbf{x}$ and the travel time associated

with this location is $\chi = \chi_0 + \Delta\chi$. We can use the total derivative of χ to get the effect of perturbations,

$$\Delta\chi = \Delta q + \frac{\partial\chi}{\partial x}\Delta x + \frac{\partial\chi}{\partial y}\Delta y + \frac{\partial\chi}{\partial z}\Delta z \tag{4.12}$$

The residual calculated from the initial guess is $r_{0v} = \tau_v - \chi(q_0, x_0, y_0, z_0)$. The v-th residual is thus $r_v = \tau_v - (\chi_{0v} + \Delta\chi) = r_{0v} - \Delta\chi_v$. To estimate the location, the goal is to minimize the residual, i.e. find the perturbation that gives the least squared residual.

$$r_v = r_{0v} - \left(\Delta q + \frac{\partial\chi_v}{\partial x}\Delta x + \frac{\partial\chi_v}{\partial y}\Delta y + \frac{\partial\chi_v}{\partial z}\Delta z\right) \tag{4.13}$$

where the unknowns are $\Delta q, \Delta x, \Delta y, \Delta z$. Then the goal is to minimize the sum of the squares of the residuals with respect to these variables, i.e., the objective is,

$$\min \sum_v r_v^2 \tag{4.14}$$

Suppose there are n pickings in one event, the matrix of residuals can be written as,

$$\begin{bmatrix} 1 & \frac{\partial\chi_1}{\partial x} & \frac{\partial\chi_1}{\partial y} & \frac{\partial\chi_1}{\partial z} \\ 1 & \frac{\partial\chi_2}{\partial x} & \frac{\partial\chi_2}{\partial y} & \frac{\partial\chi_2}{\partial z} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \frac{\partial\chi_n}{\partial x} & \frac{\partial\chi_n}{\partial y} & \frac{\partial\chi_n}{\partial z} \end{bmatrix} \begin{bmatrix} \Delta q \\ \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} r_{01} \\ r_{02} \\ \vdots \\ r_{0n} \end{bmatrix} \tag{4.15}$$

this is the standard inversion of matrix equations $\mathbf{M}\Delta\mathbf{x} = \mathbf{r}$. We use Bayesian ART method (refer to section 4.2.3) to solve this equation system and get the perturbation solution $\Delta\mathbf{x}$. The event location is obtained by adding $\Delta\mathbf{x}$ to initial guess. This solution may not be close to the real location, then we can use this as a new guess and solve the system again until the solution is good enough.

After the event location done, the coordinator node needs to send the event location to corresponding sensor node that has a arrival time picking on this event. InsightTomo then can proceed to the ray tracing and tomography computation.

4.2.3 Tomography Inversion

On the reception of event location information, the sensor nodes will trace the ray paths and send them to the nodes that perform landlord for distributed tomography computation. Since the ray tracing method used in this work is standard bending based method based on 1D velocity model and the distributed tomography computation method used here need to partition the ray paths after ray tracing is done, we will focus on tomography inversion in this section and ignore the details on ray tracing algorithms.

As discussed in chapter 3, the seismic tomography inversion problem is to solve the system,

$$\mathbf{As} = \mathbf{t} \tag{4.16}$$

where $\mathbf{A} = [\mathbf{A}_1^T, \mathbf{A}_2^T, \ldots, \mathbf{A}_N^T]^T$ and $\mathbf{t} = [\mathbf{t}_1^T, \mathbf{t}_2^T, \ldots, \mathbf{t}_N^T]^T$. This system is usually overdetermined and the inversion aims to find the least-squares solution $\mathbf{s}$ such that,

$$\mathbf{s} = \arg\min_{\mathbf{s}} \parallel \mathbf{t} - \mathbf{As} \parallel^2 \tag{4.17}$$

To solve the equation system in tomography inversion problem, there are many methods can be used as discussed in chapter 2. In InsightTomo, we employ Bayesian ART method which has been proved to be a good smoother for tomography inversion.

Algebraic Reconstruction Technique (ART) is a row action method to solve equation system. As an iterative method, ART produce a sequence of estimated vectors which converge to the required solution. Consider the system in tomography inversion $\mathbf{As} = \mathbf{t}$ where $\mathbf{A} \in \mathbb{R}^{L \times M}$ and $\mathbf{s}, \mathbf{t} \in \mathbb{R}^{M \times 1}$. The basic ART method can compute the approximation of the solution of the system with the following iterative formula,

$$\mathbf{s}^{(k+1)} = \mathbf{s}^{(k)} + \rho^{(k)} \frac{t_l - a_l^T \cdot \mathbf{s}^{(k)}}{\|a_l\|^2} a_l \tag{4.18}$$

where a_l is the $l-$th row (e.g., the $l-$th ray path traced) of $\mathbf{A}$, a_l^T is the transpose of a_l, t_l is the $l-$th component of vector $\mathbf{t}$ and $\rho^{(k)}$ is a relaxation parameter. $\|a_l\|^2 = a_l^T \cdot a_l$ and (k)

denotes the iteration number, the procedure can repeatedly operate on the equations with $l = (k) \mod (L+1)$. The formula above provides a simple iteration routine; if the system is consistent basic ART is proved to converge to the minimum-norm solution.

For the inconsistent system, a Bayesian version of the basic ART is proposed by G.T. Herman [58] for the image reconstruction in medical tomography. Suppose the system $\mathbf{As} = \mathbf{t}$ is inconsistent, then we consider the system $\mathbf{As} + \mathbf{u} = \mathbf{t}$ where $\mathbf{u}$ is chosen from given any $\mathbf{s}$. Then the system is transformed to a well-proposed problem, $\mathbf{s}$ and $\mathbf{u}$ can be solved simultaneously. Bayesian ART method has the following iterative formulas,

$$d^{(k)} = \rho^{(k)} \frac{\lambda t_i - (u_i^{(k)} + \lambda a_i^T \cdot \mathbf{s}^{(k)})}{1 + \lambda^2 \|a_i\|^2} \tag{4.19}$$

$$\mathbf{s}^{(k+1)} = \mathbf{s}^{(k)} + \lambda d^{(k)} a_i \tag{4.20}$$

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + d^{(k)} \mathbf{e}_i \tag{4.21}$$

where $\mathbf{e}_i$ is a unit vector with the $i-$th component equal to one, λ is the regularization parameter. Bayesian ART method find the solution $\mathbf{s}$ such that,

$$\mathbf{s} = \arg\min_{\mathbf{s}} \| \mathbf{t} - \mathbf{As} \|^2 + \lambda^2 \| \mathbf{s} \|^2 \tag{4.22}$$

Note that in Bayesian ART method, we need an additional vector $\mathbf{u}$ of length L, but in $(k)-$th step only one component of $u^{(k)}$ needs to update locally without communications.

To compute the tomography inversion in the network, the distributed algorithm in chapter 3 is employed to distribute the computation load, reduce the communication cost and approximate the least-squares solution of the seismic tomography inversion problem in the network.

4.3 System Implementation

To conduct the InsightTomo emulator for tomography computation, the CORE[1] and EMANE[2] network emulators [59] are employed to emulate the sensor network with sensor nodes. The advantage of emulation is that the code developed over the emulator can be transplanted to a Linux-based device, e.g., BeagleBone Black board, virtually without any modifications.

InsightTomo 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 create 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. On one hand, in a multi hop transmission the source node need to retransmit the packet through all hops once the packet lost on the path. On the other hand, the data rate can be very low 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 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 4.7 shows the application interfaces on each node for the integration

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

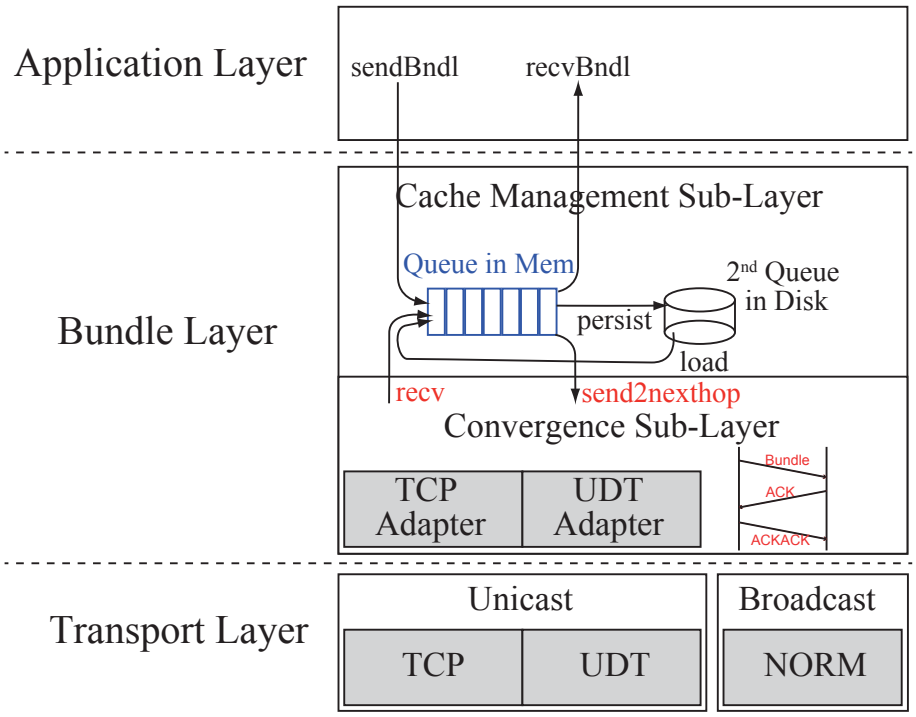


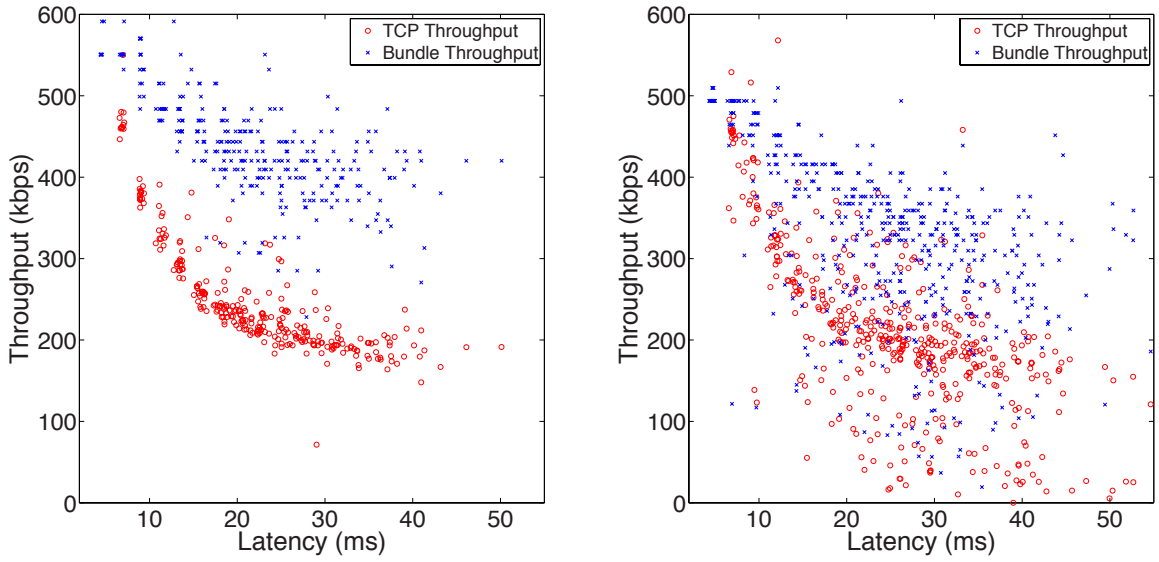
Figure 4.7 Bundle Layer Architecture.

of DTN and routing protocol. Figure 4.8 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 implement a delay-tolerant broadcasting service based on the NACK-Oriented Reliable Multicast (NORM) protocol[3]. Using NORM interface, one node can push a bundle reliably to its one-hop neighbours. 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 guarantees each node receives the same bundle at most once.

The implementation of all the algorithms in InsightTomo 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.

[3] `http://cs.itd.nrl.navy.mil/work/norm/`



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

(b) OLSR routing

Figure 4.8 Performance of Bundle layer vs TCP.

4.4 Evaluation

In this section, we evaluate InsightTomo performance with extensive experiments using real data set from the deployment on SAF at Parkfield is used in the evaluations of P-wave arrival time piking and event location. Both synthetic and real data are used to evaluate the DMET algorithm. The system implementation and experiments validate the correctness and accuracy for the proposed algorithms and the feasibility of InsightTomo system design.

4.4.1 P-wave Arrival Time Picking Accuracy

The raw seismic data set from the deployment on SAF at Parkfield is archived by IRIS[4]. The deployment is from Jan 1, 2000 to Dec 31, 2002 with 61 stations but the archived data we can download consists of 42 stations from Oct 2, 2001 to Oct 10, 2002. An extensive data set is obtained from Dr. Haijiang Zhang. This data set has more stations and longer period of data but not in seismic waves. The arrival timings in the extensive data set are

[4] `http://www.iris.edu/data/`

from manual analysis, and the event locations and velocity model are from the previous computation with double-difference tomography method [60]. We use the extensive data set for comparison in this evaluation.

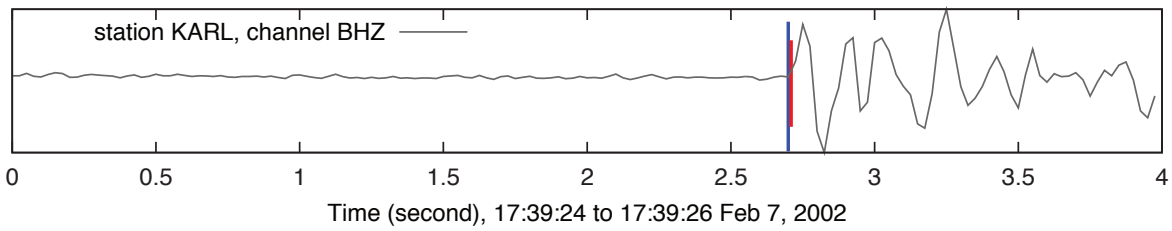


Figure 4.9 Manual pickings vs algorithm pickings on 6 stations in one event.

Figure 4.9 gives an example that compares the pickings of P-wave arrival time picking algorithm with manual pickings, the long and short vertical lines indicate algorithm picking and manual picking respectively. We can see that two pickings are close and the picking is accurate.

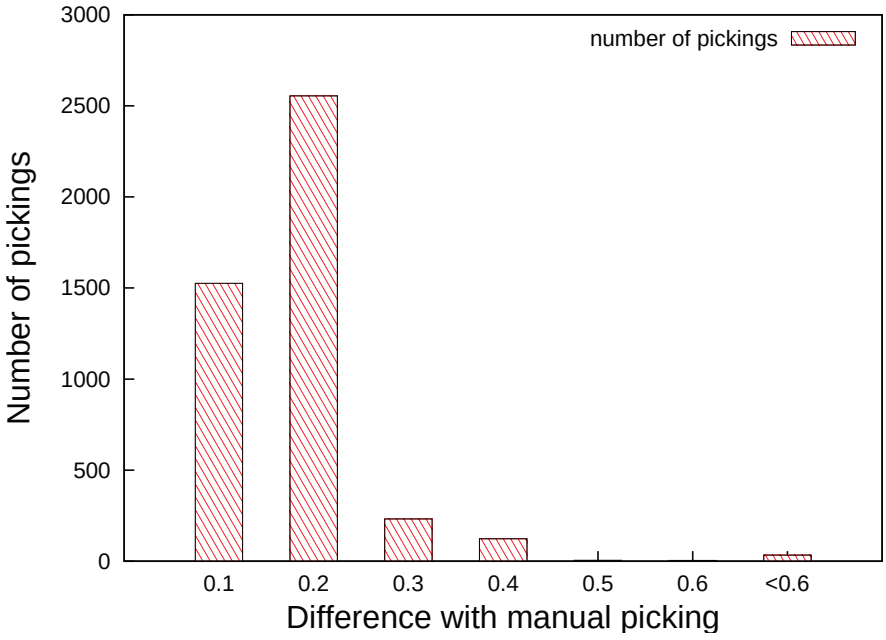


Figure 4.10 Picking Errors.

Figure 4.10 shows the algorithm picking errors in comparison with manual picks from the

extensive data set where x-axis is the picking difference between two methods and the y-axis is the number of pickings stand in that difference range. There are total 4478 pickings generated by the algorithm, which has a matching in the manual picking data for the Parkfield data. About 91% pickings from our algorithm are within 0.2 seconds of manual pickings. The mean value and the standard deviation of the difference between our pickings and manual pickings are 0.043 and 0.23.

4.4.2 Event Location Accuracy

As we discussed in the system design, the 1D reference velocity model is used for event location in InsightTomo. Figure 4.11 gives the reference model used in our evaluation. Note that this model is also used in the ray tracing and as the initial model for tomography inversion.

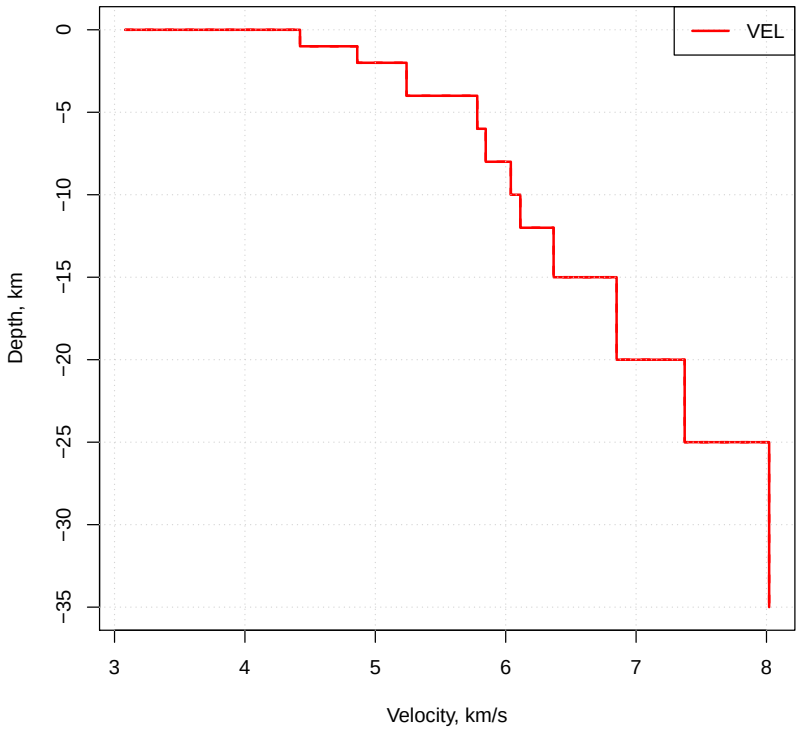
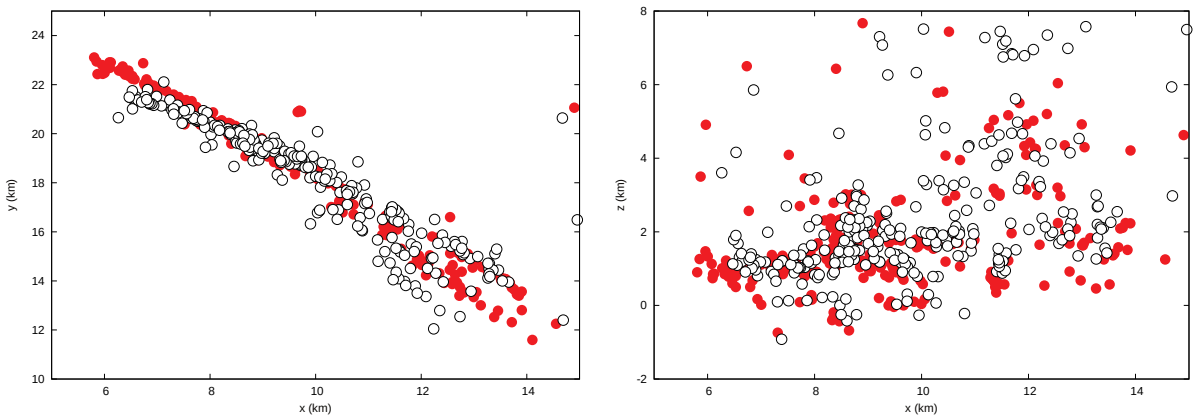


Figure 4.11 1D P-wave Velocity Reference Model.

The P-wave arrival time picking and event identification algorithm generate 433 events and 290 of them have corresponding event data in the extensive data set. The mean value and standard deviation of the difference between the positions of our algorithm and extensive data set are (0.33, 0.26) km respect to X, (0.46, 0.37) km respect to Y and (0.39, 0.40) km respect to Z. Figure 4.12 shows the event location result compared with the extensive data set.



(a) Event locations on XY

(b) Event locations on XZ

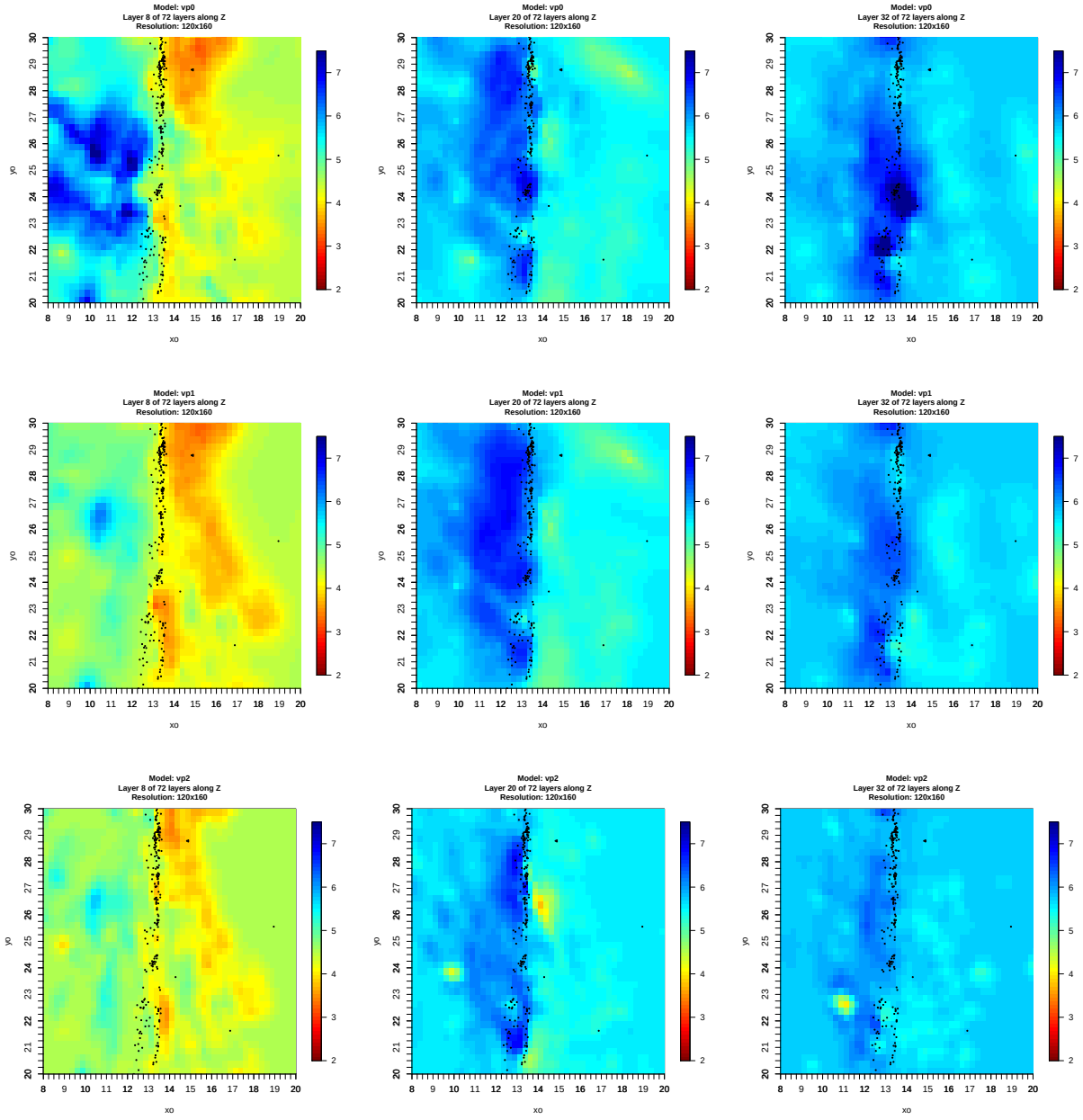
Figure 4.12 Event location result comparison, the empty circles and solid disc indicate the event location from InsightTomo and the extensive data set respectively.

4.4.3 Tomography Result

Since the real structure of the Earth subsurface is unavailable, we can not directly compare the tomography image from the real data with ground truth. To verify the correctness, accuracy and the performance for DMET algorithm, here we use the real data set from Parkfield to construct the image of seismic tomography using InsightTomo and compare with previous research result.

The tomography results of Parkfield delivered by InsightTomo System are all from the raw seismic data. Followed [60], the y-axis of the tomography rotated 40 degrees counterclockwise from north so that it is parallel to the local strike of the SAF.

Figure 4.13 gives the P-wave velocity model around SAF at Parkfield. The tomography



(a) Vp at depth = 1km (b) Vp at depth = 4km (c) Vp at depth = 7km

Figure 4.13 Horizontal slices of the P-wave velocity at depths of 1, 4, and 7 km. The fault is located around X=13.5km.

in first row is centralized calculated from the extensive data with event locations (1538 events involved). It is close to the result in [60], it is easy to see that the velocity model is different on different side of SAF. The dots in the tomography indicate the event locations estimated in InsightTomo system. A scientific fact is that the events often happen around the fault which is verified by our result. The tomography in second row is centralized calculated from the extensive data, but not all the data is used. Only 438 events happened between Oct 2, 2001 and Oct 10, 2002 are used. This is because InsightTomo only used the raw seismic data inside that time period. It makes more sense to compare the InsightTomo result with the centralized result on similar data set. We can see that the fault feature is easy to get from the second row of tomography but the velocity contrast at different sides of SAF is reduced.

The third row is the result from InsightTomo based on 433 events. A two-level DMET algorithm is applied where one landlord will start the tomography inversion with low resolution and four landlords compute partial model in high resolution. From the tomography image, except the Vp at depth 7km, the main feature of SAF can still be recognized and comparable with the centralized result. This verifies the feasibility of InsightTomo system.

PART 5

HARDWARE PROTOTYPE: DESIGN AND OUTDOOR EVALUATIONS

In this chapter, we present a sensor network system prototype for real-time in-situ seismic tomography computation. The design of the sensor network consists of hardware, sensing and data processing. This system design is evaluated both in lab environment for 3D tomography with real seismic data set (previous deployment on San Andreas Fault (SAF) in Parkfield) and in outdoor field test for 2D surface tomography.

5.1 System Design

In this section, we give the overview of our system architecture and the details of both hardware and software design. According to the motivation and requirement of the system design discussed above, the specific goals of the sensor network system design is as following:

Synchronized Sampling:

The event location and travel-time tomography requires the P-wave arrival time of earthquake events. The P-wave arrival time analysis is based on the temporal and spacial correlation of the recorded signals on stations. So all stations need to perform synchronized sampling and timestamp the record with precise UTC time. The synchronization accuracy should be less than the time interval of sampling (e.g. 20 millisecond for 50Hz sampling rate).

Long-term Robust Deployment:

To get accurate event location and high resolution seismic tomography, the more data recorded the better result can be potentially delivered. Since the earthquake activities are unpredictable, the long-term robust deployment is necessary to get enough data. Also, due to the harsh weather conditions for remote deployment, a low-cost energy efficient station with renewable energy and weatherproof capacity is required.

P-wave Arrival Time Picking:

As we discussed in the beginning of this chapter, this sensor network system will send the arrival time back instead of all raw seismic data. The system must be able to continuously monitor the signal, detect and pick arrival time in real-time.

Online Monitoring and Configuration:

To monitor the status of the network, perform the real-time signal processing and in-situ computation, the sensor network should be able to respond to external control from the base station for status report or node configuration. The command and control needs to be delivered reliably in real-time.

Distributed Computation Extension:

This system is not limited to be a in-situ signal processing and data collection framework. In the future, the system can be used for more complicated seismic analysis that may include cross correlation of signals between stations, distributed computation and so on. Those tasks will require more computation power on each sensor node. An extension for adding a computation unit is required to make this system more extensive and general.

5.1.1 System Architecture

Our system consists of several components. Figure 5.1 shows the architecture of the sensor network system design. First, the sensor nodes with seismometers and RF modules form a mesh network. Each sensor continuously records the signal, once an event happened, the sensor will detect it and pick the P-wave arrival time from the signals. Then the arrival time along with the station coordinates is delivered to the base station. The base station is a computer that equipped with RF module, it runs various tools to process the received data, compute the tomography, visualize the result, monitor the network status and configure the sensor nodes. This system can deliver either 3D tomography through event location, ray tracing and inversion, or 2D surface tomography with Eikonal tomography method [61].

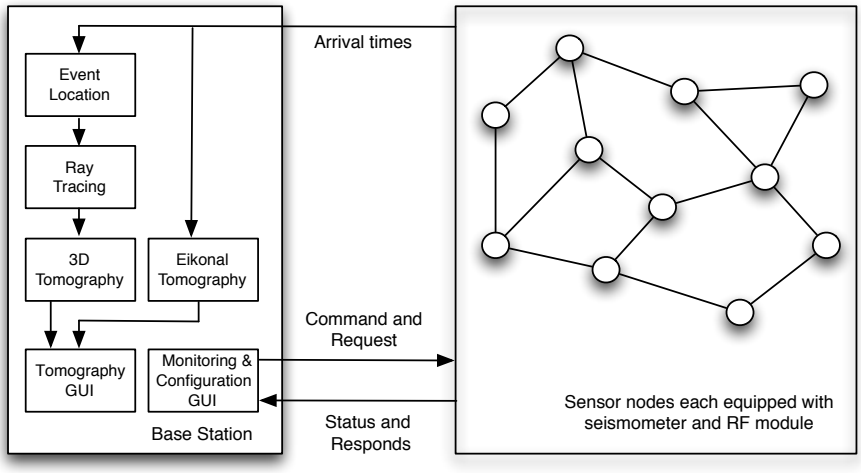


Figure 5.1 Sensor Network System Architecture.

5.1.2 Hardware Design

Considering the system design goals, our sensor node design encapsulates all the hardware components in a weatherproof plastic box. Figure 5.2 shows the configuration of the sensor node in the field. The sensor node box connected with a 10 Watt solar panel to get renewable energy. A single-axis 4.5Hz seismometer, GeoSpace geophone is connected as the sensor component. We mounted a 9 dBi omnidirectional antenna on the box for the communication of a 900MHz RF module to get a reasonable line-of-sight range. All the connections are also sealed by weatherproof connectors for the harsh environment. The total weight of each sensor node station is about 10 pounds which can be carried by a person for remote deployment.

Figure 5.3 shows the hardware components inside the sensor node box, with a dimension of $0.82 \times 0.55 \times 0.31$ (inch). All our components in the system are mounted on a single-layer PCB board. The core of the system is a TI MSP430F6779 processor, 25MHz, 512KB of program ROM and 32KB of SRAM. This processor also provides seven independent 24-bit Sigma-Delta ADCs with different inputs and variable gain. This is one of the most powerful low-cost and ultra-low-power micro controller from MSP430 family. Besides, there are rich I/O interfaces to support flexible extensions on the processor, including 6 SPI, 4 UART, 2

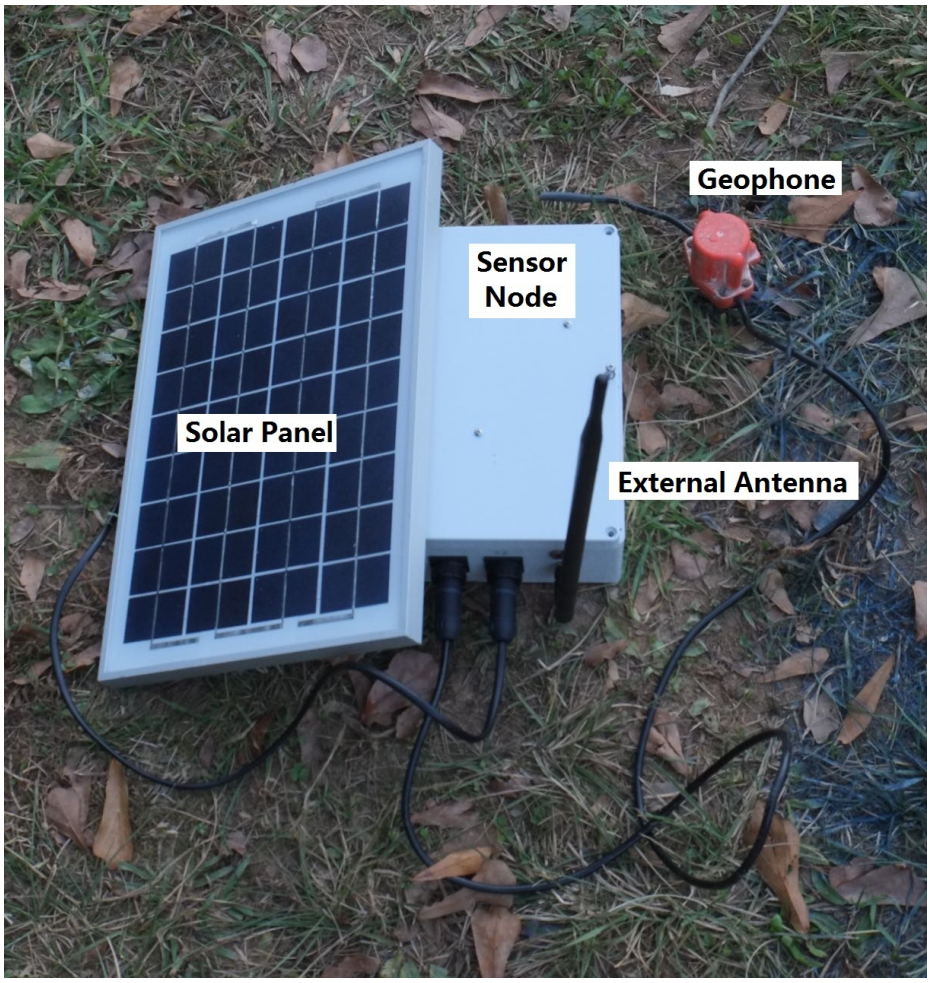


Figure 5.2 Sensor node in the field.

I2C and 90 GPIO pins. A JTAG interface is connected to program the board.

The low-power radionova M10478-A2 GPS interface is connected to the processor through UART1 to provide raw GPS data, and through GPIO 40 to provide PPS (pulse-per-second) signal capturing. The GPS interface is used to provide the coordinates information (latitude, longitude and altitude) of the sensor node, and the timestamps for recorded data with accurate UTC time. The accuracy of its time pulse is up to 50 ns.

For wireless communication, we employed the XBee-PRO 900HP module to provide a low-power, low maintenance, long outdoor range and self organized wireless network. As an commercial industry product, XBee module is easy to configure and use. It provides best-in-class range wireless connectivity to devices. They take advantage of the DigiMesh networking

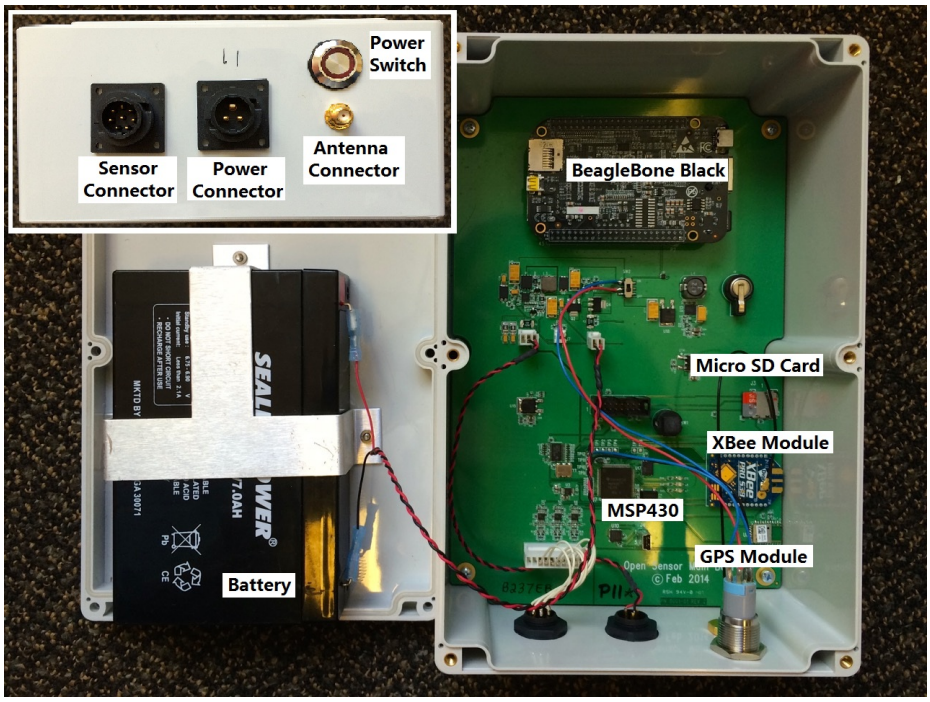


Figure 5.3 Hardware components in the box.

protocol from Digi company. They can run on dense network operation and support for sleeping routers for energy efficient. Besides, various point-to-multipoint configurations are available for the network. The MSP430 is connected to XBee using UART2 with 9600 baud that provides 960 Kbps data rate. The XBee module is connected to a external 9dBi omnidirectional antenna with a SMA connector.

Since the sensor network is designed to sense the signal, pick and send the P-wave arrival timestamp back without transmitting the raw seismic data. All the raw data is stored in a micro SD card for other post analysis required by seismologists. We use the DM3D-SF connector and connect the processor with memory card through SPI0 for SPI communication and clock, and through GPIO 68 for SPI card select pin.

The node sensor connector is designed to connect up to three channels of seismometer. The node can connect either to a single-axis or a tri-axis geophone. Both geophones are passive instruments and the ground motion can generate voltage which is digitized by the ADC module in MSP430. The ADC channel 0, 1 and 3 on MSP430 are connected to channel

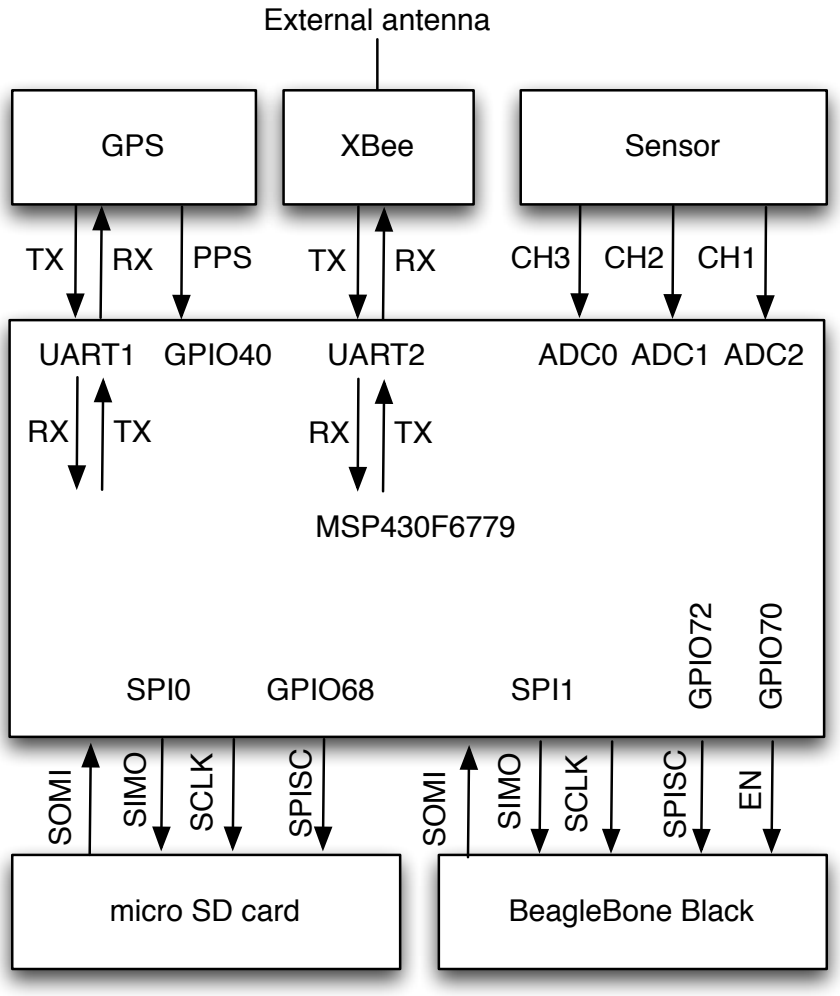


Figure 5.4 Main Hardware Components Connection.

3, 2 and 1 on a tri-axis geophone or only channel 1 on a single-axis geophone.

For the distributed computation extension requirement, one BeagleBone Black (BBB) module is connected with the expansion connector to the board through its SPI0 interface. We use the SPI1 on MSP430 for SPI communication and clock, the GPIO 72 for SPI card select pin and the GPIO 70 as the power switch for BBB. The main hardware components connection relationships are shown in Figure 5.4.

5.1.3 Sensing and Data Processing

Aim to achieve the system design goals, based on our hardware design, we give the description of the software design for sensing and data processing on the sensor node in this section. Figure 5.5 illustrates the framework of the sensing and data processing components on the sensor node.

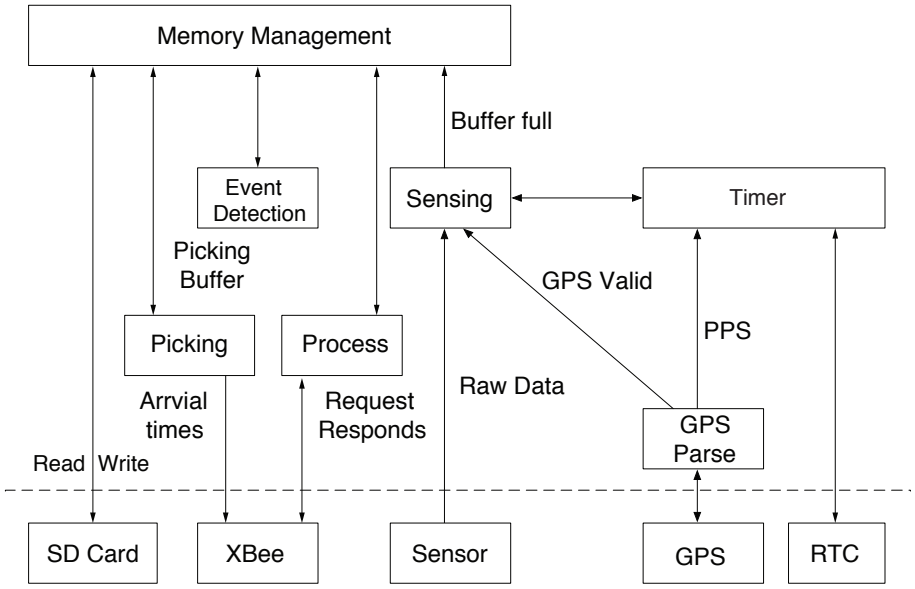


Figure 5.5 Sensing and Data Processing Framework.

Since the accurate event location and high-resolution tomography are depend on precise timing by utilizing the temporal and spacial correlation of recorded signals across stations. The first goal of our system design is synchronized sampling and precise UTC time timestamp for the recorded data. Our collaborators from seismology requires 100Hz sampling rate on our open nodes, thus, the synchronization error should be no more than 10 milliseconds. Notice that, the synchronized sampling is based on time synchronization but not the same. Synchronized sampling does not only means that all sensor nodes in the network has the same sample interval but also sample at the same time point.

In the hardware design, each sensor node employs a low-cost GPS receiver that provides UTC time information and PPS signal. The GPS system time from GPS signal has an

accuracy within 50ns referenced to UTC time, it can be used for time synchronization. The problem is that decoding and processing of the GPS message can generate delays and degrade the synchronization accuracy. Instead, we use PPS signals to synchronize the RTC. To achieve the synchronized sampling, we designed a Timer component to maintain RTC with millisecond resolution. Then when the system catches the first valid PPS interrupt, the timer is reset and keep counting on milliseconds. When next valid PPS interrupt is captured, if the timer is not in exact thousand milliseconds, the timer will be reset and the RTC will be synchronized properly.

Notice that, the GPS signal can disappear or the PPS signal can not fire properly. If the time period without GPS signal or valid PPS signals is long, the sampling across sensor nodes might not be synchronized. The sensor node tags every second of data with a timestamp and a flag. The timestamp represents the time point corresponding to the first sample in this second. The flag indicates whether the samples in this second is under valid time synchronization or not. The system will tag the second of data invalid synchronization if: (1) there is no GPS signal for 60 seconds; (2) there is no valid PPS interrupts for 20 seconds.

The *Sensing* component samples the sensor with 10 milliseconds sample interval according to *Timer* service. There is a small circular buffer to sample one hundred samples (one second data under 100Hz) from sensor. Once this buffer is full. The *Sensing* component will send it to memory management to write the buffer into micro SD card with proper timestamp and flag. Also, the *Event Detection* component takes this buffer and perform the event detection processing, if one event is detected, the related buffered data is processed by *Picking* component to get the arrival time and send it through XBee module. There is another module *Process* that processes the requests from base station and send responds back for status monitoring and network configuration. More details about this can be found in section 5.1.4.

5.1.4 Online Monitoring and Configuration

Considering the complexity and remoteness of environment monitoring, online status monitoring and sensor node configuration is highly desired.

With online monitoring, users can easily get the status of the sensor nodes in the network. This is very helpful when the deployment is initiating, one can monitor all the sensor nodes in the network remotely without actually visiting them remotely. There are two modes in the system, test and deploy. When the deployment starts, the node will start with test mode, and it will report the status periodically. After a while, if the node status is normal, users can switch the nodes to deploy mode where the sensor node only report the status if requested. The status report consists of the GPS status (satellite numbers, latitude, longitude, altitude), the sensing status (number of events detected, number of seconds recorded), the power status (solar panel input voltage and battery voltage).

In the sensor node, many parameters need to be configurable. For example, the window size and threshold in the event detection and picking algorithms. Since different kind of events have different properties, different parameters could identify various classes of events according the interesting of seismologists. Also, the sensing parameters such as channel, data resolution, sensor status and reference voltage gain can also be configured.

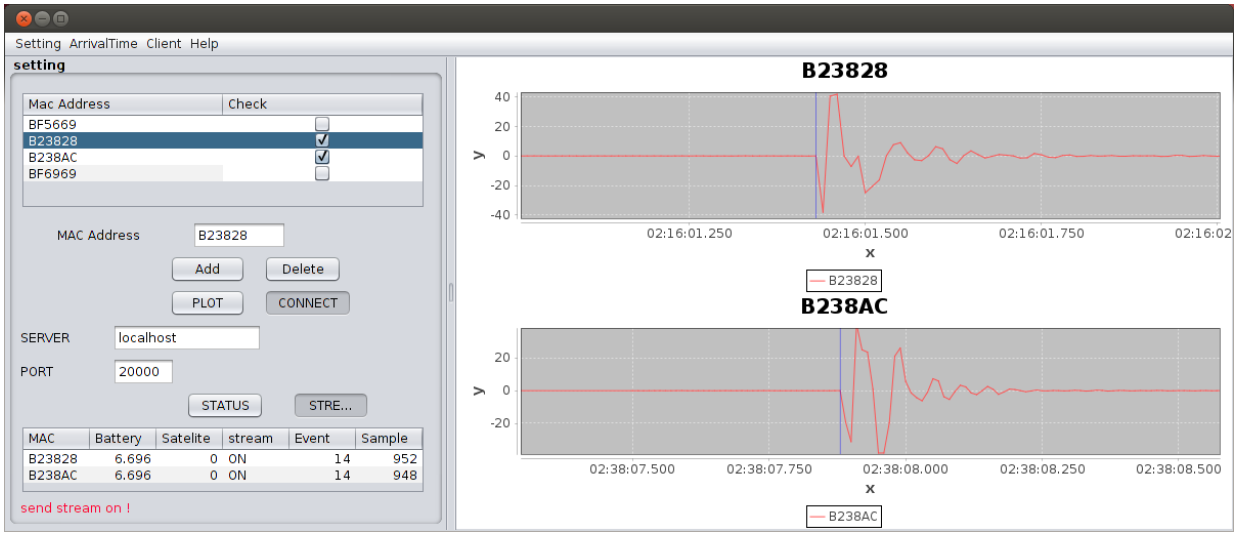


Figure 5.6 Stream data with arrival time picking on the monitoring and configuration tool.

In our system design, the sensor node only sends the arrival time with station information back by default. One problem is that in different field or with different interests for events, the parameters for event detection and picking can be varied. After deployment, users need to know whether the arrival time picking is accurate or not. Thus, a stream option is added into the system. When the stream option is on, the sensor node will send the raw stream data in the picking buffer with the arrival time. Users can visualize it on base station to check the picking accuracy on base station in real-time. Figure 5.6 shows the stream data and arrival time picking from the monitoring and configuration tool on the base station. Besides, seismologists might be interesting in the raw data for other analysis in the deployment period. But they can not afford to visit the sensor node remotely all the time. Another feature in this system is that users can download the data from any node by specifying the start and end time point.

5.2 Data Quality and Picking Accuracy

Before the field deployment and end-to-end tomography delivery of the system. We conducted several tests to verify the quality of recorded data with the sensor node and the accuracy of the arrival time picking mechanism.

5.2.1 Data Quality

The scientific value of the data is the final and most important measurement of the sensor network system. The first test here is to see weather this system can provide scientifically meaningful data to seismologists. Since it is not easy to find a place to record earthquake events and it might take long time to validate. With the suggestion from seismologists, we conduct a hammer shock test that is commonly used by the experts for preliminary. This test is to use a hammer to hit on the ground to generate seismic wave propagation. The signal from a hammer shock is not so different from an earthquake except the energy is smaller.

To validate the data quality, the test is conducted to compare the data recorded by

Figure 5.7 SigmaBox Configuration.

our sensor node and a current state of the art commercial seismic acquisition system called SigmaBox. The SigmaBox is designed by iSeis Corporation, shown in Figure 5.7. In the test, 7 sensor nodes and 4 SigmaBox are deployed. Four sensor nodes are placed with SigmaBoxes side by side to compare the recorded data quality, see Figure 5.8. The distance between each pair of nodes is 10 meters. We used the hammer to hit the ground near the SigmaBox 70 and sensor node 18 for 20 times. In Figure 5.9, we can see the recorded data for a hammer shock event by our sensor node and SigmaBox. The SNR is similar between two data record

Figure 5.8 SigmaBox and sensor node deployment.

and the seismologist were satisfied with the data quality overall.

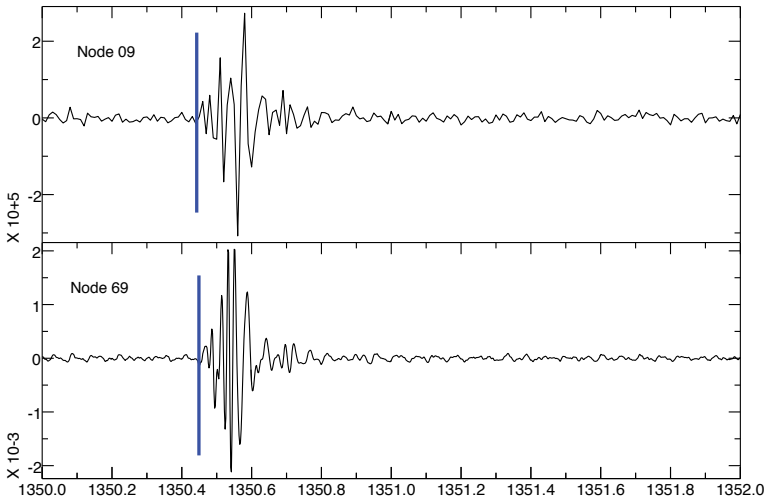


Figure 5.9 Waveform of a hammer shock event on sensor node 09 and SigmaBox 69.

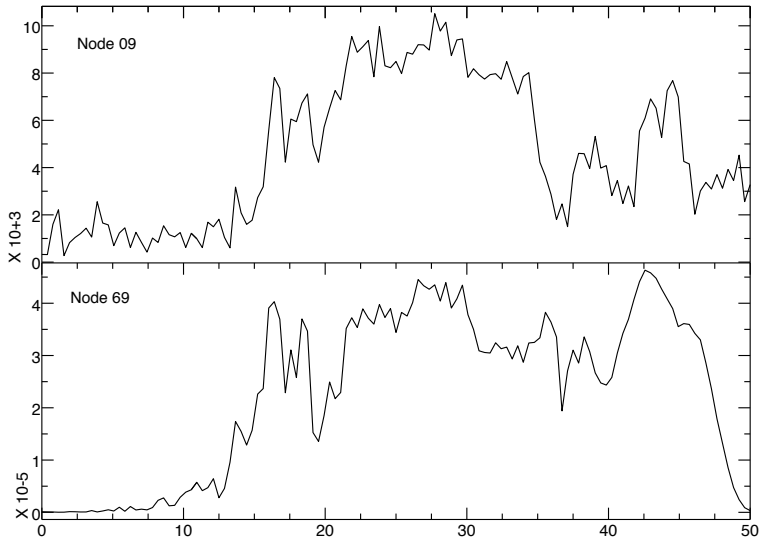


Figure 5.10 Spectrum of the hammer shock event on sensor node 09 and SigmaBox 69.

The spectrum of the waveform in Figure 5.9 is shown in Figure 5.10. We can see that the spectrum distribution is similar between two signals. This further validate the data quality of our sensor node. Notice that the SigmaBox costs about $3K, while the sensor node costs less than $1K.

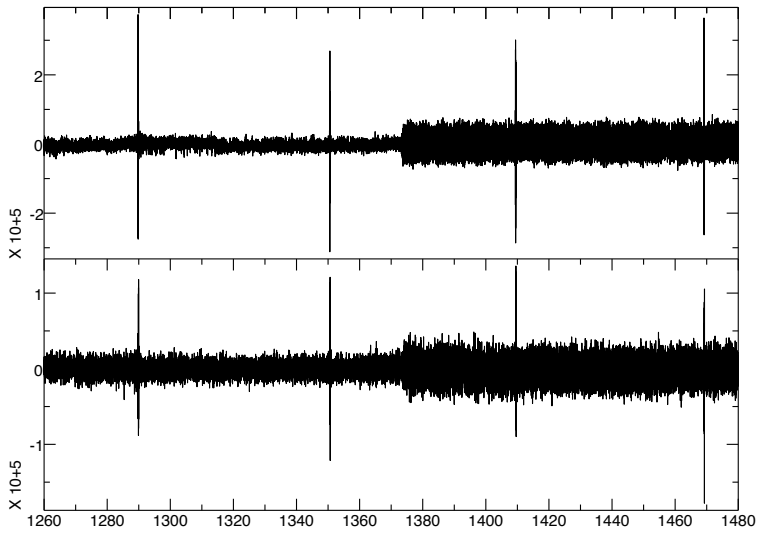


Figure 5.11 Signal Noise Level Change

One problem in the test is that the noise level of the signal can be changed due to the environmental or hardware problem. Figure 5.11 gives an example of the noise level change in one of our hammer shock tests from two nodes. This test is finished on a grass area of a community. The lower noise of the first two minutes is mostly from the air conditions and traffic nearby. A lawn mower was operating around after a while in the test and generated higher noise. In real world deployment, there are many factors which can influence the noise level such as wind, rain, traffic and so on. Since the picking algorithm depends on the noise level σ_1^2, an adaptive computation for noise is necessary. Here we add an adaptive noise level estimation base on the periodical noise level computation with a filter. Basically, the system evaluate the noise level periodically and use the weighted average of recent noise level values as σ_1^2.

5.2.2 P-wave Arrival Time Picking Accuracy

The example in Figure 5.9 shows the arrival time picking of our algorithm on sensor node and SigmaBox sensed data. Notice that the sampling rate of SigmaBox was 500Hz in the test. In the example, the time difference between two pickings is 6 millisecond, which is smaller than the sampling interval of our sensor node. The average time difference of all

pair of pickings in this test is 4.2 millisecond. This test shows that the algorithm can deliver similar arrival time result on different node. It only means that the recorded data quality from two kinds of nodes is similar to perform detection and picking algorithm. To validate the accuracy of the picking algorithm, the algorithm should perform on the real data set with manual pickings from experts as the reference.

Figure 5.12 Audio to Sensor Channel adapter.

We used a real data set obtained from seismologists. This data set was recorded from a previous deployment on San Andreas Fault (SAF) at Parkfield. The deployment is from Jan 1, 2000 to Dec 31, 2002 with 61 stations. The data set has been cut into short waveforms that contain events with the manual pickings. Then the problem is that how can we send the waveforms to sensor node for validation. We made an adapter from audio input to channel 0 as shown in Figure 5.12. Then waveforms were converted into audio wave and can be sent to the sensor node by any audio player on a computer, cellphone or tablet.

There are totally 4326 arrival times picked by the algorithm from the data set. About 90% picking errors of our algorithm are within 0.2 seconds. The mean value and the standard

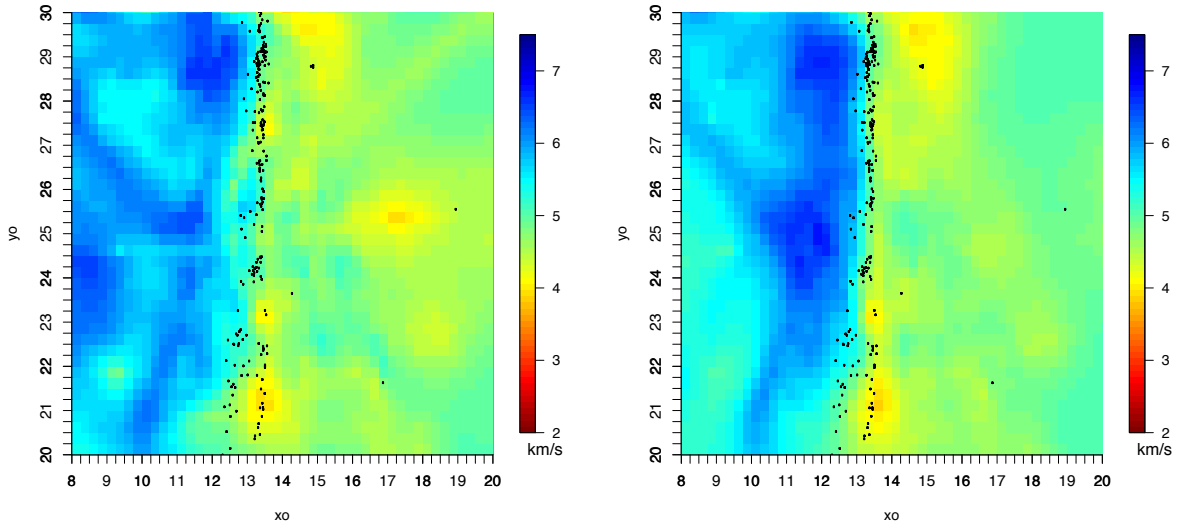
deviation of the difference between our pickings and manual pickings are 0.044 and 0.232. This is comparable with some recent method in seismology literature [62].

5.3 System Evaluation

In this section, we conduct two experiments to evaluate the sensor network system for both 3D and 2D surface tomography.

5.3.1 Parkfield 3D Tomography

From the discussion of previous section, we use an adapter to send the waveform from computer to our sensor nodes simulating the sampling process. The sensor node then process the data, picked the arrival times and send to a base station set in the lab. After the base station received some arrival times, it should compute the event location. Notice that this computation is an online process, because in real deployment, one can not predict when and how many arrival times can be received since the earthquake activity is not predictable.



(a) Vp at depth = 2km

(b) Vp at depth = 3km

Figure 5.13 Horizontal slices of the P-wave velocity at depths of 2 and 3 km. The fault is located around X=13.5km.

The base station only receives the arrival times from the sensor nodes and has the

knowledge that which picking is from which node. In all of these pickings, there might be false alarms or some small and remote event is only detected by few sensor nodes. As we known, to estimate an event location and origin time, at least four pickings from different sensor nodes are required, the event detected only by one or two nodes is impossible to be located. Also, more pickings from different sensor nodes for one event usually lead to a better estimation. Thus there are two steps in event location as we discussed, (1) Event Identification where the base station identifies how many events existing in a series of arrival times received and which pickings belong to the same event; (2) Location Estimation which uses Geiger's method to estimate the event location from the arrival times of that event.

After identified the events and computed the event locations, base station will do ray tracing based on the event information and the station coordinates received with the arrival times, followed by the 3D tomography inversion.

Figure 5.13 shows the tomography result from the base station. It is easy to see that the velocity model is different on different side of SAF. The dots in the tomography indicate the event locations estimated on base station. A scientific fact is that the events often happen around the fault which is verified by our result. We can see that the fault feature is easy to get from the tomography result. This result is comparable with the previous research on the Parkfield tomography [60].

5.3.2 Hammer Shock Field Test

To verify the sensor network system in outdoor field, we conduct a field test and created the event with hammer shock on the ground to generate the surface wave. In this case, we can control the location of the event source and it is easy to verify if the recorded data is meaningful, the arrival time pickings is correct and the 2D surface tomography result is validated.

In the previous discussion, the hammer shock test has already been used for data quality and arrival time picking validation. From that test, we found that on the soil ground, the hammer shock can generate waves propagated up to around 30 meters, depends on how hard

Figure 5.14 Hammer Shock Test Deployment.

the hammer hit the ground. Thus, 25 sensor nodes are deployed on a 20×20 meter area with 5 meter space between the adjacent sensor nodes. Figure 5.14 shows the deployment of 25 sensor nodes. In the area we deployed, the upper half of it covered by wet soil under the tree while the other half is covered by drier soil under the sunshine in day time. The reason we choose this area is that we would like to see the difference from Eikonal tomography based on the property of the different acoustic wave propagation speed in wet and dry soil.

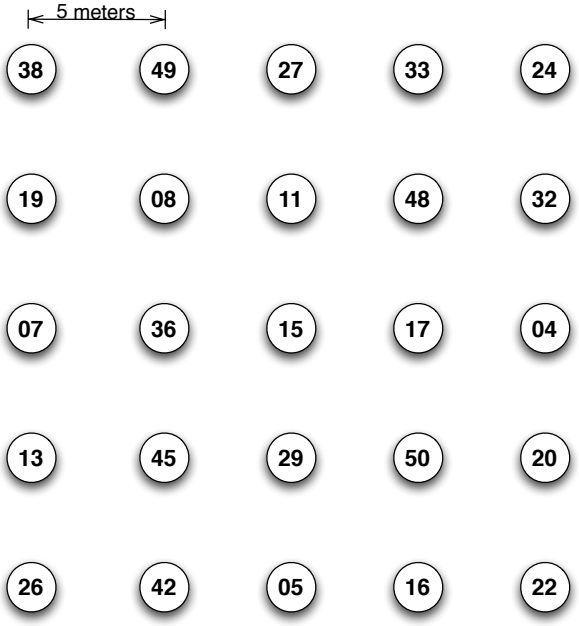


Figure 5.15 Deployment map of sensor nodes.

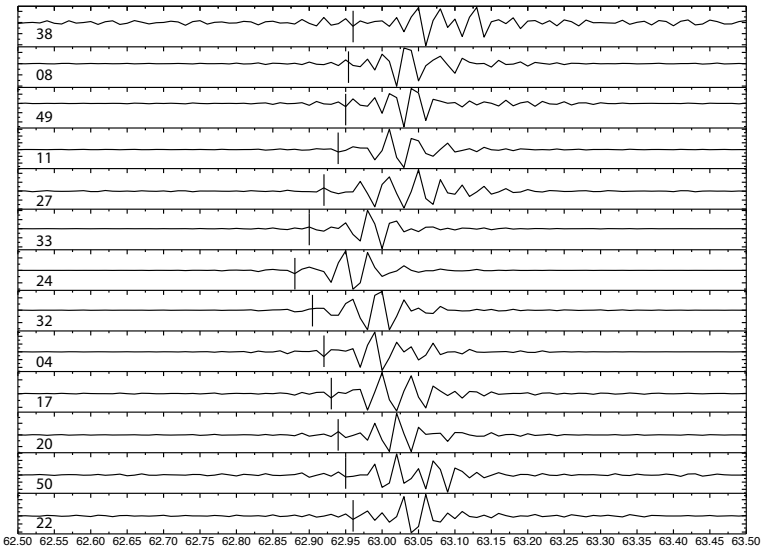


Figure 5.16 Hammer shock event captured.

After the deployment done, the base station monitored the status of all sensor nodes and told us when all nodes stared working normally. Then we created 5 hammer shocks (events) beside each station, totally 125 events were created. The deployment map of the sensor nodes is shown in Figure 5.15.

Finally, after all hammer shocks done, the base station received more than 2000 arrival times and computed the 2D surface tomography with Eikonal tomography method. Before showing the tomography result, we take a closer look at one seismic event recorded by the sensor network and the arrival times picked out of this event.

The hammer shock event captured in Figure 5.16 was generated by hit beside node 24, which located on the upper right corner in the deployment map. From the recorded signal and picked arrival times shown in Figure 5.16, node 24 got the earliest arrival time and the further nodes got the relatively delayed arrivals, which shows the wave propagation in the deployed area. Within such a small area, this further verified the synchronized sampling accuracy of our sensor network system.

In this test, the base station received totally 2012 arrival times picked on the sensor nodes. Some events are not picked on some nodes. The reason is that the sensor node can not get good event signal, it depends on how hard the hammer hit the ground and how far

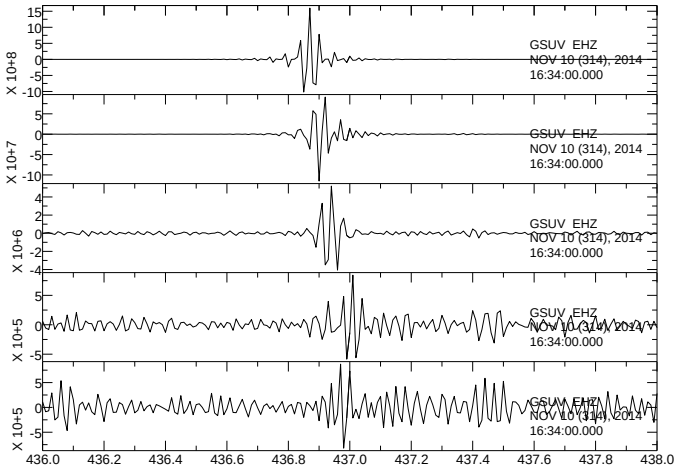


Figure 5.17 Hammer shock event captured along the diagonal.

the sensor node is from the event location. Figure 5.17 shows an event generated beside node 32 and the corresponding data record along the diagonal of the deployment area. We can see that the SNR is very low on the furthermost node 22, thus it is hard to detect this event and pick the arrival time.

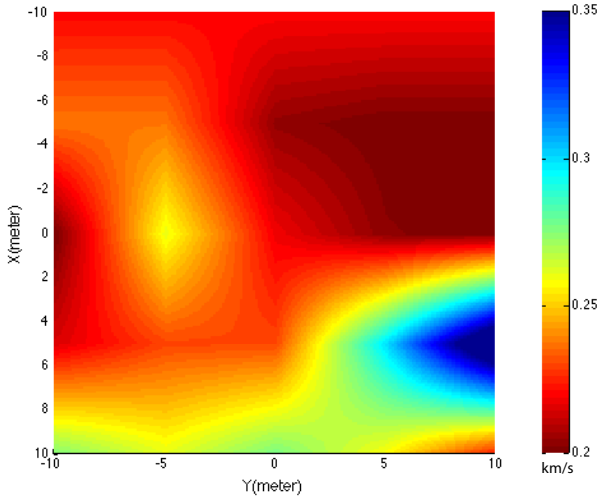


Figure 5.18 2D surface wave tomography.

Out of 2012 arrival times, the base station identified 96 events with 1905 arrival times.

Figure 5.18 shows the 2D surface wave tomography delivered by the base station. According to the research on acoustic wave propagation in soil [63], the impedance mismatch from the water to air is much greater than the water to soil frame. Thus, more saturated the soil is, slower the acoustic wave propagates in it. As we discussed above, the upper side of the deployed area contains more water in the soil. This observation is shown in the final tomography result.

PART 6

CONCLUSIONS

In this dissertation, three research aspects are addressed for real-time in-situ seismic tomography in sensor network. First, we propose an innovative multi-resolution evolving tomography algorithm that distribute and balance the tomography inversion computation load to the network, while computing real-time high-resolution 3D tomography not only balance the computation load, but also achieves low communication cost and high data loss tolerance. Second, we present InsightTomo, an end-to-end emulation system that performed in-network data processing and obtained tomography using distributed computation. Evaluation was carried out on real data and the obtained results are comparable with previous research that used centralized method. Third, we designed a sensor network testbed that performs in-situ signal processing and obtains 3D or 2D surface tomography in real-time. The hardware and software design of the system focused on delivering a low-cost, energy efficient and reliable system to monitor and image the earthquake zone or active volcano. Several tests and experiments were conducted to show: (1) the recorded data quality is similar to current commercial product; (2) the system can deliver validated tomography result. Both the emulation and testbed platforms mark the collaboration between geophysicists and computer scientists, which provides opportunities to introduce new technology for geophysical monitoring. The approaches presented here has broader implication beyond tomography inversion and can be easily extended to oil and natural gas exploration.

REFERENCES

[1] W.-Z. Song, R. Huang, M. Xu, A. Ma, B. Shirazi, and 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)*, Jun. 2009.

[2] H. M. Iyer and P. B. Dawson, *Imaging volcanoes using teleseismic tomography*, H. M. Iyer and K. Hirahara, Eds. Chapman and Hall, 1993.

[3] A. L. Vesnaver, F. Accaino, G. Bohm, G. Madrussani, J. Pajchel, G. Rossi, and G. D. Moro, "Time-lapse tomography," *Geophysics*, vol. 68, no. 3, pp. 815–823, 2003.

[4] J. M. Lees, "Seismic tomography of magmatic systems," *Journal of Volcanology and Geothermal Research*, vol. 167, no. 1-4, pp. 37–56, 2007.

[5] ——, "The magma system of Mount St. Helens: non-linear high-resolution P-wave tomography," *Journal of Volcanology and Geothermal Research*, vol. 53, pp. 103–116, 1992.

[6] J. M. Lees and R. S. Crosson, "Tomographic Inversion for Three-Dimensional Velocity Structure at Mount St. Helens Using Earthquake Data," *Journal of Geophysical Research*, vol. 94, no. B5, pp. 5716–5728, 1989. [Online]. Available: http://dx.doi.org/10.1029/jb094ib05p05716

[7] G. P. Waite and S. C. Moranb, "VP Structure of Mount St. Helens, Washington, USA, imaged with local earthquake tomography," *Journal of Volcanology and Geothermal Research*, vol. 182, no. 1-2, pp. 113–122, 2009.

[8] S. C. Moran, J. M. Lees, and S. D. Malone, "P wave crustal velocity structure in the greater Mount Rainier area from local earthquake tomography," *Journal of*

Geophysical Research, vol. 104, no. B5, pp. 10 775–10 786, 1999. [Online]. Available: http://dx.doi.org/10.1029/1999jb900036

[9] J. M. Lees, N. Symons, O. Chubarova, V. Gorelchik, and A. Ozerov, "Tomographic images of kliuchevskoi volcano p-wave velocity," *AGU Monograph*, vol. 172, pp. 293–302, 2007.

[10] S. Ohmi and J. M. Lees, "Three-dimensional P- and S-wave velocity structure below Unzen volcano," *Journal of Volcanology and Geothermal Research*, vol. 65, no. 1-2, pp. 1–26, 1995.

[11] H. Wu and J. M. Lees, "Three-dimensional P and S wave velocity structures of the Coso Geothermal Area, California, from microseismic travel time data," *Journal of Geophysical Research*, vol. 104, no. B6, pp. 13 217–13 233, 1999. [Online]. Available: http://dx.doi.org/10.1029/1998jb900101

[12] J. M. Lees and H. Wu, "P wave anisotropy, stress, and crack distribution at Coso geothermal field, California," *Journal of Geophysical Research*, vol. 104, no. B8, pp. 17 955–17 973, 1999. [Online]. Available: http://dx.doi.org/10.1029/1999jb900158

[13] H. Wu and J. M. Lees, "Attenuation structure of Coso geothermal area, California, from wave pulse widths," *Bulletin of the Seismological Society of America*, vol. 86, no. 5, pp. 1574–1590, Oct. 1996. [Online]. Available: http://bssa.geoscienceworld.org/cgi/content/abstract/86/5/1574

[14] J. M. Lees and H. Wu, "Poisson's ratio and porosity at Coso geothermal area, California," *Journal of Volcanology and Geothermal Research*, vol. 95, no. 1-4, pp. 157–173, 2000.

[15] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat Monitoring: Application Driver for Wireless Communications Technology," in *1st ACM SIGCOMM Workshop on data communication in Latin America and the Caribbean*, Apr. 2001.

[16] R. Szewczyk, J. Polastre, A. Mainwaring, J. Anderson, and D. Culler, "Analysis of a Large Scale Habitat Monitoring Application," in *Proc. 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Nov. 2004.

[17] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein, "Energy Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet," *Proc. 10th international conference on Architectural support for programming languages and operating systems*, Oct. 2002.

[18] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Wireless sensor networks for structural health monitoring," in *Proc. 4th ACM conference on Embedded networked sensor systems (SenSys)*, Nov. 2006.

[19] K. Chebrolu, B. Raman, N. Mishra, P. K. Valiveti, and R. Kumar, "BriMon: A Sensor Network System for Railway Bridge Monitoring," in *The 6th Annual International Conference on Mobile Systems, Applications and Services (MobiSys)*, Jun. 2008.

[20] C. Hartung, R. Han, C. Seielstad, and S. Holbrook, "FireWxNet: A Multi-Tiered Portable Wireless System for Monitoring Weather Conditions in Wildland Fire Environments," in *The 4th International Conference on Mobile Systems, Applications, and Services (MobiSys 2006)*, Jun. 2006.

[21] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, "Fidelity and Yield in a Volcano Monitoring Sensor Network," in *Proc. 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, Nov. 2006.

[22] T. L. Murray and E. T. Endo, "A real-time seismic-amplitude measurement system (rsam)," ser. USGS Bulletin, 1992, vol. 1966, pp. 5–10.

[23] R. Sleeman and T. van Eck, "Robust automatic P-phase picking: an on-line implementation in the analysis of broadband seismogram recordings," *Physics of the Earth and Planetary Interiors*, vol. 113, no. 1-4, pp. 265–275, 1999.

[24] R. Tan, G. Xing, J. Chen, W. Song, and R. Huang, "Quality-driven Volcanic Earthquake Detection using Wireless Sensor Networks," in *The 31st IEEE Real-Time Systems Symposium (RTSS)*, San Diego, CA, USA, 2010.

[25] L. Geiger, "Probability method for the determination of earthquake epicenters from the arrival time only," *Bull.St.Louis.Univ*, vol. 8, pp. 60–71, 1912.

[26] J. M. Lees and R. S. Crosson, "Bayesian Art versus Conjugate Gradient Methods in Tomographic Seismic Imaging: An Application at Mount St. Helens, Washington," *Institute of Mathematical Statistics*, vol. 20, pp. 186–208, 1991.

[27] B. R. Julian and D. Gubbins, "Three-dimensional seismic ray tracing," *GEOPHYSICS*, vol. 43, pp. 95–113, 1997.

[28] R. Fischer and J. M. Lees, "Shortest path ray tracing with sparse graphs," *GEOPHYSICS*, vol. 58, no. 7, pp. 987–996, Jul. 1993.

[29] T. J. Moser, "Shortest path calculation of seismic rays," *GEOPHYSICS*, vol. 56, no. 1, pp. 59–67, Jan. 1991.

[30] J. Um and C. Thurber, "A Fast Algorithm for Two-point Seismic Ray Tracing," *Bulletin of the Seismological Society of America*, vol. 77, no. 3, pp. 972–986, Jun. 1987.

[31] A. L. Vesnaver, "Irregular grids in seismic tomography and minimum-time ray tracing," *Geophysical Journal International*, vol. 126, no. 1, pp. 147–165, Jul. 1996.

[32] D. Gordon and R. Gordon, "Component-Averaged Row Projections: A Robust, Block-Parallel Scheme for Sparse Linear Systems," *SIAM Journal on Scientific Computing*, vol. 27, no. 3, pp. 1092–1117, 2005. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/040609458

[33] H. Straková, W. N. Gansterer, and T. Zemen, "Distributed QR factorization based on randomized algorithms," in *PPAM'11 Proceedings of the 9th international conference on Parallel Processing and Applied Mathematics - Volume Part I*, 2012, pp. 235–244.

[34] R. A. Renaut, "A Parallel Multisplitting Solution of the Least Squares Problem," *Numerical Linear Algebra with Applications*, vol. 5, no. 1, pp. 11–31, 1998.

[35] L. T. Yang and R. P. Brent, "Parallel MCGLS and ICGLS Methods for Least Squares Problems on Distributed Memory Architectures," *The Journal of Supercomputing*, vol. 29, no. 2, pp. 145–156, 2004. [Online]. Available: http://www.springerlink.com/content/r1k4756082kt5382/

[36] G. Mateos, I. D. Schizas, and G. B. Giannakis, "Performance Analysis of the Consensus-Based Distributed LMS Algorithm," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, 2010.

[37] ——, "Consensus-Based Distributed Least-Mean Square Algorithm Using Wireless Ad Hoc Networks," in *Forty-Fifth Annual Allerton Conference*, 2007, pp. 568–574.

[38] I. D. Schizas, G. Mateos, and G. B. Giannakis, "Consensus-Based Distributed Recursive Least-Squares Estimation using Ad Hoc Wireless Sensor Networks," in *Signals, Systems and Computers, 2007. ACSSC 2007.*, 2007, pp. 386–390.

[39] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in Ad Hoc WSNs With Noisy LinksPart I: Distributed Estimation of Deterministic Signals," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 350–364, Jan. 2008.

[40] I. D. Schizas, G. B. Giannakis, S. I. Roumeliotis, and A. Ribeiro, "Consensus in Ad Hoc WSNs With Noisy LinksPart II: Distributed Estimation and Smoothing of Random Signals," *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1650–1666, Apr. 2008.

[41] I. D. Schizas, G. Mateos, and G. B. Giannakis, "Distributed LMS for consensus-based in-network adaptive processing," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2365–2382, 2009.

[42] S.-Y. Tu and A. H. Sayed, "Diffusion Strategies Outperform Consensus Strategies for Distributed Estimation over Adaptive Networks," *To appear in IEEE Transactions on Signal Processing*, 2012.

[43] A. H. Sayed and C. G. Lopes, "Distributed Recursive Least-Squares Strategies Over Adaptive Networks," in *Signals, Systems and Computers, 2006. ACSSC '06. Fortieth Asilomar Conference on*, Nov. 2006, pp. 233–237.

[44] S. Kaczmarz, "Angenäherte Auflösung von Systemen linearer Gleichungen," *Bulletin International de l'Académie Polonaise des Sciences et des Lettres*, vol. 35, pp. 355–357, 1937.

[45] G. T. Herman, *Reconstruction from Projections: The Fundamentals of Computerized Tomography.* Academic Press, 1980.

[46] A. Björck and T. Elfving, "Accelerated projection methods for computing pseudoinverse solutions of linear equations," *BIT*, vol. 19, pp. 145–163, 1979.

[47] A. H. Andersen and A. C. Kak, "Simultaneous Algebraic Reconstruction Technique (SART): A superior implementation of the ART algorithm," *Ultrasonic Imaging*, vol. 6, no. 1, pp. 81–94, Jan. 1984.

[48] Y. Censor, D. Gordon, and R. Gordon, "Component averaging: An efficient iterative parallel algorithm for large and sparse unstructured problems." *Parallel Computing*, vol. 27, no. 6, pp. 777–808, 2001.

[49] ——, "BICAV: a block-iterative parallel algorithm for sparse systems with pixel-related weighting," *IEEE Transaction on Medical Imaging*, vol. 20, no. 10, pp. 1050–1060, Oct. 2001‘.

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

[51] J. M. Elble, N. V. Sahinidis, and P. Vouzis, "GPU computing with Kaczmarz's and other iterative algorithms for linear systems," *Parallel Computing*, vol. 36, pp. 215–231, Jun. 2010.

[52] G. Kamath, L. Shi, and W.-Z. Song, "Component-Average Based Distributed Seismic Tomography in Sensor Networks," in *The 9th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, May 2013, pp. 88–95. [Online]. Available: http://dx.doi.org/10.1109/DCOSS.2013.17

[53] M. J. Berger and J. Oliger, "Adaptive mesh refinement for hyperbolic partial differential equations," *Journal of Computational Physics*, vol. 53, no. 3, pp. 484–512, Mar. 1984.

[54] A. Michelini, "An adaptive-grid formalism for traveltime tomography," *Geophysical Journal International*, vol. 121, no. 2, pp. 489–510, May 1995. [Online]. Available: http://dx.doi.org/10.1111/j.1365-246x.1995.tb05728.x

[55] A. Curtis and R. Snieder, "Reconditioning inverse problems using the genetic algorithm and revised parameterization," *Geophysics*, vol. 62, no. 5, pp. 1524–1532, Oct. 1997. [Online]. Available: http://dx.doi.org/10.1190/1.1444255

[56] W. Spakman and H. Bijwaard, "Optimization of Cell Parameterizations for Tomographic Inverse Problems," *Pure and Applied Geophysics*, vol. 158, no. 8, pp. 1401+, 2001.

[57] L. Shi and W.-Z. Song, "Distributed Least-Square Computing in Networks: A Survey and Comparison," http://sensorweb.cs.gsu.edu/sites/default/files/publication/PDF/report/DistributedLS.pdf.

[58] G. T. Herman, H. Hurwitz, A. Lent, and H.-P. Lung, "On the Bayesian Approach to Image Reconstruction," *Information and Control*, vol. 42, pp. 60–71, 1979.

[59] J. Ahrenholz, T. Goff, and B. Adamson, "Integration of the CORE and EMANE Net-

work Emulators," in *MILITARY COMMUNICATIONS CONFERENCE, 2011 - MILCOM 2011*, 2011, pp. 1870–1875.

[60] H. Zhang, C. Thurber, and P. Bedrosian, "Joint inversion for Vp, Vs, and Vp/Vs at SAFOD, Parkfield, California," *Geochem. Geophys. Geosyst.*, vol. 10, no. 11, pp. Q11 002+, Nov. 2009. [Online]. Available: http://dx.doi.org/10.1029/2009gc002709

[61] F.-C. Lin, M. H. Ritzwoller, and R. Snieder, "Eikonal tomography: surface wave tomography by phase front tracking across a regional broad-band seismic array," *Geophysical Journal International*, vol. 177, no. 3, pp. 1091–1110, Jun. 2009. [Online]. Available: http://dx.doi.org/10.1111/j.1365-246x.2009.04105.x

[62] H. Zhang, C. Thurber, and C. Rowe, "Automatic P-Wave Arrival Detection and Picking with Multiscale Wavelet Analysis for Single-Component Recordings," *Bulletin of the Seismological Society of America*, vol. 93, no. 5, pp. 1904–1912, Oct. 2003. [Online]. Available: http://dx.doi.org/10.1785/0120020241

[63] M. L. Oelze, W. D. O'Brien, and R. G. Darmody, "Measurement of attenuation and speed of sound in soils," *Soil Science Society of America Journal*, vol. 66, no. 3, pp. 788–796, 2002.