Real-time In-situ Ambient Noise Seismic Imaging in Sensor Networks

by

Maria Valero

(Under the Direction of WenZhan Song)

Abstract

The use of seismic arrays as a tool for imaging subsurface infrastructures and monitoring the corresponding underground activities enables real-time subsurface security and surveillance applications. However, the existing approaches rely on manual data collection and/or centralized computing, which is not scalable as network size grows, and bottleneck problems can occurr when all data is sending to a central point. These approaches may also take a long time to get useful results. In this dissertation, we investigate and propose an in-situ and cooperative subsurface imaging system for a variety of subsurface infrastructure imaging applications. The proposed approach integrates in-situ signal processing techniques as well as inter-nodes communication and cooperation to obtain reliable velocity maps for subsurface characterization and monitoring. It generates real-time subsurface images by taking advantage of collective computation power in sensor networks while avoiding transferring all raw data to a central place or server. Eikonal tomography and Spatial Autocorrelation methodologies are studied to solve imaging challenges. The system is autonomous, self-healing, scalable and almost independent of external interventions. A communication-reduced method is investigated to meet bandwidth, communication, and energy constraints of the sensors during the imaging process. Experiments are conducted in both, emulator and field scenarios.

An exhaustive evaluation regarding bandwidth utilization and communication cost were conducted to highlight the benefits of the proposed approach. The uses can be extended to other applications like border security, building monitoring, underground water detection for agriculture, and more.

INDEX WORDS: Subsurface imaging system, in-situ signal processing, cooperative communication, eikonal tomography, spatial autocorrelation, communication-reduced method.

Real-time In-situ Ambient Noise Seismic Imaging in Sensor Networks

by

Maria Valero

B.S. Computer Engineering, University of Tachira (Venezuela), 2003

M.S. Computer Science, University of Los Andes (Venezuela), 2009

A Dissertation Submitted to the Graduate Faculty

of The University of Georgia in Partial Fulfillment

of the

Requirements for the Degree

Doctor of Philosophy

Athens, Georgia

2019

Real-time In-situ Ambient Noise Seismic Imaging in Sensor Networks

by

Maria Valero

Approved:

Major Professors: WenZhan Song

Committee: Xianqiao Wang
Mable Fok
Fan-Chi Lin

Electronic Version Approved:

Ron Walcott
Interim Dean of the Graduate School
The University of Georgia
December 2019

Real-time In-situ Ambient Noise Seismic Imaging in Sensor Networks

Maria Valero

October 3, 2019

Dedication

To my children *Santiago* and *Samuel*. Thank you, boys, for growing with me during this process and always cheering me up. Thank you for your understanding, support and love. I love you both. To my beloved husband *Jose*. You have been my number one guide. Thank you for helping me so much, being with me all the time, and supporting me when I needed the most. To my wonderful Mom *Sonia* for her prayers and support during all my life. I love you, mommy. To my awesome brother *Marco*, thank you brother for your orientation and support.

Daddy, this is for you... I love you... rest in peace.

"For whatever is born of God overcomes the world;
and this is the victory that has overcome the world
our faith." 1 John 5:4

Acknowledgments

I would like to express my special appreciation and thanks to my advisor Professor Dr. WenZhan Song; you have been a tremendous mentor for me. Thank you for encouraging my research and allowing me to grow as a research scientist. Your advice on both research as well as on my career have been invaluable.

I would like to thank Dr. Fangyu Li, for the patient guidance, encouragement, and advice he provided throughout this journey. I have been extremely lucky to have a person who cared so much about my work, and who responded to my questions and queries so promptly.

I must express my gratitude to all the members of my examination committee, Dr. Mable Fok, Dr. Fan-Chi Lin, and Dr. Xianqiao Wang for their invaluable help and the constructive comments.

I would like to express my sincere gratitude to all the faculty and staff of the College of Engineering, University of Georgia for fulfilling my dream of being a graduate student here and provide support and advice all the time. I am particularly grateful for the assistance given by Dr. Lawrence Hornak, Mrs. Margaret Sapp, and Ms. Angie Malone.

Thank you to my fellow lab mates, I will always remember our talks and discussions.

Thank you my good friend Maria Alejandra (Malala) for all the help during this time. You have been a rock for me.

Finally I thank my God, my good Father, for guiding me day by day. You are the one who let me finish this journey. I will keep on trusting You for my future. Thank you, Lord.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Knowing what is happening inside the subsurface, in deep and shallow scale, has been always a topic of interest for engineers, seismologists, geophysics, petroleum geologists, space scientists, and more. One of the most effective ways to understand underground dynamics is generating subsurface images. Many methods have been used for subsurface imaging generation, and most of them use the information of passive energy sources, like earthquakes [2, 3]. In the past years, researchers have made efforts to study passive imaging that comes from ambient noise information [4, 5], which can be applied to regions with non-existing seismicity and can be an attractive cheap scenario since there is no need of producing passive energy sources (like explosions) for generating reliable subsurface images.

Seismic sensors are currently used for gathering seismic data that is later processed to obtain subsurface images using ambient noise methods [6–9]. Current solutions employ image reconstruction methods that rely on a centralized approach, which means sending the raw data to a central station for further processing. Current sensor networks do have a high capacity of data processing, then thinking on leverage these capacities introduce a new aspect

on subsurface imaging generation. Can we generate subsurface images without transmitting raw data to a central place?, can we generate images in real-time? can we do the process of imaging autonomous?, can the sensors cooperate with each other for transmitting only meaningful data?

In-situ processing, also knows as near-data processing, is the handle and management of data that is collected in place, which means processing data inside the same sensor. The in-situ feature enables real-time estimations and results because the data do not need to be transmitted to other places for processing. Also, in-situ processing facilitates autonomous sensor behavior as there is no need for human intervention during the time the process is conducted. Furthermore, sensors can autonomously cooperate with each other to transmit only reduced and pre-processed data to enable distributed computation of the results, which means that optimization can be done to generate reliable images with minimal data transmission. Because sensor networks have also communication limitations due to bandwidth and energy constrains, such kind of system needs to include optimal approaches for meeting these constraints.

In this work, three main solutions to accomplish in-situ and cooperative subsurface imaging system in sensor networks are presented.

The first one is a sensor system that uses an ambient noise imaging methodology called Distributed Eikonal Tomography. This is an innovative distributed cooperative system for real-time seismic imaging generation through ambient noise data. We integrate in-situ processing techniques to correlate the noise signals between sensors and derive the phase velocity under the limited network resource constraints. The methodology can be implemented using aggregation and consensus algorithms for distributed computing exchanging only information with immediate neighbors. The system was tested using three applications, USArray ambient noise data, Sweetwater TX ambient noise data, and a field deployment at the University of Georgia campus. We compared the results with the centralized approach for validation

purposes. The system was also evaluated under field conditions of sensor networks, such as loss of packets to demonstrate its robustness in terms of loss tolerance. We envision this system can be used for many exploration applications. Methodology, results and evaluations are presented in Chapter 2: Distributed Eikonal Tomography in Sensor Networks.

Then, we evolve the system to make it able to image shallow subsurface infrastructures. Because Eikonal Tomography is not suitable for small areas, we adopt another technique that can be used with ambient noise called Distributed Spatial Autocorrelation (dSPAC). This innovative in-situ spatial autocorrelation method is designed to image different layers of underground infrastructures at different frequency bands and finally result in a 3D image. We propose a methodology that produces in-situ and real-time results. Similar to Distributed Eikonal Tomography, in dSPAC, sensor nodes cooperate with each other in a suitable topology to produce shallow underground images. The system was tested using two applications: pipeline detection and water leakage monitoring. We evaluated the results against ground truth, and we ensured to meet network constraints like bandwidth and communication. Methodology, results and evaluations are presented in Chapter 3: Distributed Spatial Autocorrelation in Sensor Networks.

Finally, thinking on the major problem of current sensor networks - the bandwidth and energy limitations, we design an approach to optimize communication in our system. In distributed ambient noise imaging the most intensive communication process is the transfer of data for cross-correlation [10]. To cross-correlate data, every node has to send its own data to the neighbor nodes. Even though some communication-reduction techniques were applied (reduction of data using a pre-specific narrow band of frequencies, compression techniques, etc.), sending data to all the neighbors can be inefficient. Furthermore, some nodes waste computational performance doing cross-correlations that can be done in other nodes. We propose a new communication-reduced method that can be applied in ambient noise imaging. The method is designed as a combinatorial optimization problem that first transforms the

topology of the mesh network in a suitable graph for a transportation problem [11]. Then, the optimization problem is solved to get the best nodes for computing cross-correlation to meet network limitations. By applying the proposed communication-reduced method, we show that energy and computational cost of the nodes is also preserved. We performed multiple experiments with bandwidth and energy limitations and shows how this optimization helps to improve the subsurface in-situ and cooperative imaging technique. Methodology, results and evaluations are presented in Chapter 4: Subsurface imaging in sensors: Optimized Communication for In-situ Ambient Noise Imaging.

The new approach taken in this dissertation is general, and it can be implemented as a new field network paradigm for real-time imaging of highly dynamic and complex environments, including both natural and man-made structures. We believe the system can be applied to a wide range of sustainability-related topics. The potential scientific and social impact is significantly and broadly widespread.

1.2 Background

Ambient noise seismic imaging (ANSI) has been widely used for extracting surface wave velocity maps in geophysical fields. The method has been applied worldwide (e.g US [7], Asia [8], Europe [12], New Zealand and Australia [9].) A considerable part of these approaches was developed to treat the travel time (time that it takes for seismic waves to travel from one node to another) between every station pair independently. Some approaches, like the one presented by *Lin et al.* [4], have utilized an array of seismic stations to gather information and treat all travel time measurements together to improve the resolution of the tomographic result (velocity map). Even though these approaches have been successfully applied, they lack real-time results. The cross-correlation process needs at least several days for collecting data, and then manual extraction is needed to gather the information to a central server.

Thus velocity maps may take days or months in being generated.

Real-time seismic imaging generation is possible across an array of sensors due to the capabilities of current sensor networks. Furthermore, distributed cooperation between nodes for generating seismic imaging has proven to be a milestone in-network computation. Examples of the development of such computing and network methodology for seismic imaging applications can be found in [13–17]. All of them have been successful in generating 2D and 3D seismic tomography by applying travel time tomography techniques and using earthquake information to illuminate the subsurface of the earth.

In ambient noise seismic imaging, the data are recovered from ambient seismic noise, which implies no need for passive energy sources like earthquakes. Ambient noise imaging can be applied to regions with non-existent seismicity, and it produces reliable measurements at frequencies that are particularly difficult using earthquakes or explosions due to scattering and attenuation. This advantage represents an attractive cheap scenario since producing passive energy sources (explosions) in non-seismic areas is very costly.

The first attempt to compute Ambient Noise Seismic Imaging (ANSI) in distributed sensor networks was made in our previous work [1]. We proposed the use of a method called *distributed eikonal tomography* for generating velocity maps. However, in Valero et al. [1], we assume the travel times have been already calculated to perform eikonal method, and this implies only the last step of the ANSI process. Later, we carefully incorporate and improve all previous steps of the ANSI process to get a complete system [10], namely recording raw data, performing distributed cross-correlation, calculating in-situ frequency-time analysis, deriving travel time measurement, performing distributed eikonal tomography and allowing velocity maps visualization.

To the best of our knowledge, this was the first comprehensive end-to-end system to compute ANSI under distributed constraints through sensor network computing capabilities, cooperation between nodes, and in-situ distributed seismic imaging algorithms. The ANSI

prototype system has implemented all steps of ambient noise tomography, from raw data to velocity maps, and it can be extended as a general field instrumentation platform for ambient noise seismic data.

Nowadays, there is an increasing interest in structure monitoring and anomaly detection using ambient noise techniques. Besides eikonal tomography, another method called spatial auto-correlation (SPAC) can be used in small spaces that are not suitable for eikonal tomography. Passive seismic monitoring based on ambient noise assessment has been used for risk management and reduction in many engineering applications. Stork et al. [18] investigated the potential to monitor seismic velocity changes following a hypothetical leak of CO_2 from Aquistore storage site in Saskatchewan, Canada using passive monitoring methods. They showed the map of near-surface velocities obtained with ambient noise techniques could be useful for near-surface static corrections when using passive-source seismic reflection surveys to image and monitor the reservoir. However, further similar studies are required to assess leak detection at other CO_2 storage sites.

Liu et al. [19] conducted an experiment to test the idea of using ambient noise to characterize building vulnerability for strong ground motions. By placing one geophone on each floor in a building (from the basement to the seventh floor), they were able to determine shear wave velocities, which implies that the ambient noise approach appears sensitive enough to reveal some wave propagation difference and variations in a civil infrastructure. Other studies has also used ambient seismic noise [20] to study building infrastructures for detecting and locating structural damages [21–23]. Mordret et al. [24] also measured the shear velocity and the apparent attenuation factor of a building using ambient noise techniques. They linked the velocity variations with weather parameters. They determined that the variation in velocities in the building is intrinsically related to air humidity. Olivier et al. [25] also took advantage of ambient noise techniques to investigate an active underground mine (Garpenberg, Sweden) by cross-correlating seismic noise generated by mining activities. Their

study revealed the existence of a high-velocity zone and a low-velocity zone that corresponded with known ore bodies inside the mine.

Other approaches, different from ambient noise, has been used to detect buried pipelines and potentially their leakages. De Paola et al. [26] performed an experimental installation of a pipeline to investigate the behavior of a leaking pipe in a realistic condition. In particular, the interaction of a leak with the surrounding soil. However, no subsurface velocity studies were made. In 2017, Amran et al. [27] presents a study on the possibility of using ground penetrating radar (GPR) with a frequency of 1GHz to detect pipes and leakages in underground water distribution piping system; and in 2019, De Coster et al. [28] introduce a method that includes a physically-based near-field antenna effects removal approach and reflection detection and segmentation algorithms with GPR. However, there are no approaches using ambient noise for pipeline and water leakage detection.

To the best of our knowledge, this is the first attempt to use distributed sensor networks running ambient noise techniques algorithms to imaging buried pipeline structures and its possible water leakages based on a proposed technique called distributed spatial auto-correlation (dSPAC).

The proposed distributed eikonal tomography and distributed spatial correlation techniques are performed on sensor networks, which means that the network limitations has to be addressed to avoid bottlenecks and other network problems. In general, in a network the communication subsystem has much higher energy consumption than the computation subsystem. It has shown that transmitting one bit may consume as much as executing a few thousand instructions [29]. Therefore, reducing communication has a significant impact on saving sensor's energy.

The study of network communication efficiency in seismic area is relatively new, and it has been focus on how the nodes sense and transmit the data to a central station. For example, the literature on channel model for Electromagnetic (EM) based in sensors is

rich and has a lot of application on seismic exploration. However, few works exists on the routing protocols. Due to the limited transmission range of EM waves in the underground environment, multihop communication has been investigated. For example, the idea of dense sensor networks with multi-hop communication for oil and gas exploration was presented in [30]. Moreover, Conceiccao et al. [31] developed a TCP/IP based simulator for underground sensor networks. Furthermore, they evaluated the performance of various multiple access scheme in [32]. The path connectivity problem for EM-based underground sensor networks was investigated in Trang et al. [33] which showed that low volumetric water content and low operating frequency lead to a higher probability of connectivity. In Liu et al. [34], the throughput of EM-based underground sensor networks was optimized to achieve the QoS (Quality of Service) requirement.

In the field of ambient noise imaging, mostly of the work are based on improving imaging methodology instead to network communication. However, Xie et al. [35] present a signal detection scheme with low-communication overhead for distributed ambient noise imaging. The method in that paper only requires each sensor to exchange a limited number of frequency samples to with each other. They also were able to detect whether a pair of sensors contains coherent signal for cross-correlate. A large scale deployment or survey was not presented and the approach is only based on local temporal energy.

To the best of our knowledge, this is the first optimization methodology for improving communication and energy during data cross-correlation process for ambient noise seismic imaging. Also, we believe that it is the first attempt to transform a wireless seismic network in a transport network to select the best nodes to compute and avoid communication overhead.

1.3 Dissertation overview

To have a better understanding of this dissertation path, an overview is presented to guide the reader into the main motivation and contributions presented in each chapter as shown in Fig. 1.1 .

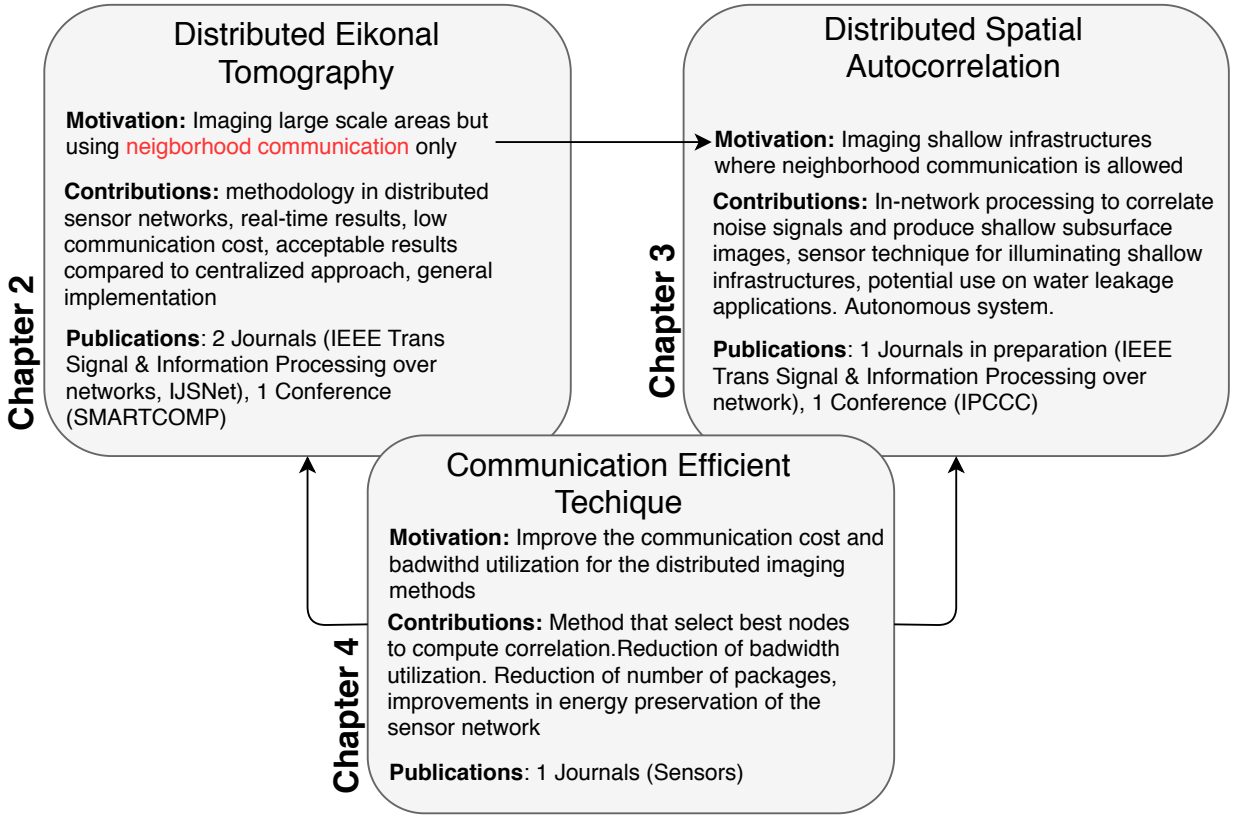


Figure 1.1: Dissertation Overview.

Chapter 2 introduces the distributed version of a well-known tomography method called Eikonal Tomography. This method has been successfully applied in geophysics area to image extensive subsurface areas; for example, the west part of the United States with a seismic array called USArray. In this chapter, a new real-time and in-situ method is processed to allow neighbor nodes talk with each other and exchange meaningful data instead of transmitting all the data to a central place or post-processing after manual retrieval. In traditional eikonal tomography, every node correlates its data with the data of all other nodes; this is called all-

to-all cross-correlation. All-to-all cross-correlation has a tremendous impact on a distributed network solution as the transmission of high volumes of data can cause bottlenecks and communication problems. We propose to use neighborhood communication only, and we show that even though the final tomographic result can be a degraded version of the traditional one, it is a good approximation with less communication cost and real-time calculations. We designed and implemented an end-to-end system that is reliable, autonomous, and provide results in real time. This chapter has led to two journal publications and one conference publication.

Chapter 3 introduces the distributed version of a method called spatial auto-correlation. This method complements the eikonal tomography method and is a reliable solution as the need for all-to-all correlations is not imperative in this method. In other words, the neighborhood distributed approach is perfectly applicable to this method. We show dSPAC can be applied to small deployments too, and it seems to have a great potential to illuminate shallow infrastructures. We design again an end-to-end system that can illuminate shallowly buried pipelines, and we realized that it is possible to note a difference of the underground infrastructures when a water leakage occurs. We envision that the method can be applied to a large variety of application, including non-invasive water leakage detection. This chapter has led to one conference publication and one journal preparation paper.

Chapter 4 introduces a novel method to reduce the communication cost of distributed tomography. The method selects the best nodes for computing correlation between all possible neighbors, ensuring that the result is the most reliable with the minimum cost. By reducing the cost, the use of the bandwidth is improved, the number of transmitted packets is reduced, and the energy of the sensors is preserved for a longer time. This chapter has led to one journal publication.

Finally, chapter 5 concludes the dissertation and present the future work path for the research.

Chapter 2

Distributed Eikonal Tomography in Sensor Networks

2.1 Motivation

Over the last years, ambient noise tomography has become one of the fastest growing research areas in seismology and exploration geophysics. Compared to earthquake-based seismic tomography methods, ambient noise tomography is particularly useful in imaging shallow earth structures [4, 5]. Moreover, because of the persistent nature of the seismic background noise, temporal variation of the earth structure can be analyzed and monitored by studying the variation in the noise cross-correlation function [36–38]. Ambient noise methods have the advantage of being low cost and having resistant repeating sources with a minimal environmental disturbance.

The problem is that the existing ambient noise tomography methods use post-processing approaches to recover subsurface structures, and they do not have the capability of obtaining information in real time. Current approaches involve manual collection of raw seismic data from the sensors to a central server for post-processing and analysis. Sensor network

technology has matured to the point where it is now possible to deploy and maintain large networks for earth structures monitoring [17, 39, 40]. Also, the computing power of every sensor can be one of the most exciting opportunities for in-situ computing due to the ability to generate real-time imaging of the earth's interior and study the complex dynamic processes occurring within. However, it is virtually impossible to collect all raw data to a central place through wireless sensor networks due to the severe energy and bandwidth constraints[1] and disruptions caused by harsh environmental factors. Even though system-level challenges of deploying wireless sensor networks are significant, focusing on distributed in-network signal processing and computing can help to support real-time tomographic imaging.

In this chapter, we present a novel real-time ambient noise imaging system through in-situ computing in sensor networks, and we illustrate the process from signal processing challenges to end-to-end system design. The Ambient Noise Seismic Imaging (ANSI) system is a sensor network of nodes that can efficiently perform seismic ambient noise cross-correlations and compute real-time tomography by continuously monitoring detailed structures within the top few kilometers underground. This system is particularly cost attractive because the ambient noise used for tomographic imaging does not rely on any passive sources or earthquakes, and it is autonomous and self-sustainable with all processing and computing in the network. To achieve the goal, we integrate the cutting edge seismic noise analysis, tomography, sensor communication, and large data computation methods. Specifically, we integrate communication and computation devices with sensors such that data recorded by every sensor can be cross-correlated on site with the data recorded at other neighboring sensors, and tomography can be achieved without transmitting the raw data back to a data center. The seismic raw data are also stored at each device database for future analysis; since the sensors form a mesh network, the raw data can be accessed by any device within the

[1]According to [29], the energy of transmitting 1KB a distance of 100m is approximately the same that executing 3 million of instructions in one processor. Hence, local data processing is crucial.

network if needed.

2.2 Methodology

Ambient noise seismic imaging is a kind of passive imaging where vibrations of the ambient noise recorded by passive sensor arrays can be used to image the medium through which waves travel. To perform tomography with ambient noise, many methods can be used, for example eikonal tomography [6], straight-ray tomography [41], seismic interferometry [42]. All of them have its own properties and mathematical formulation. For straight-ray and interferometry, an inversion problem needs to be settled. The eikonal method is a surface wave tomography that complements the traditional methods. The main advantages of this method are: (i) there is no explicit regularization; this makes the method largely free from ad hoc choices; (ii) the method account for bent rays, and ray tracing is not needed. The gradient of the phase front provides information about the local direction of wave travel. In traditional tomography, the use of bent rays would need iteration with ray tracing performed on each iteration; and (iii) **the ray tracing, matrix construction and inversion of the traditional methods are not needed. Those have been replaced by surface fitting, computation of gradients and average.** Therefore, the method is very fast in terms of computational cost and suitable for distributed approaches.

The ANSI methodology, that we introduce in this chapter, involves the steps shown in Fig. 2.1. Those include: i) using seismic sensors (green circles) to measure the vibration of the ambient noise; ii) calculating the cross-correlation of the signal waves with neighbors and performing a frequency-time analysis to obtain travel time measurements of the ambient noise signal; and iii) using eikonal tomography [6] to build velocity maps.

We briefly summarize the steps described in [1, 4, 43, 44] as follows:

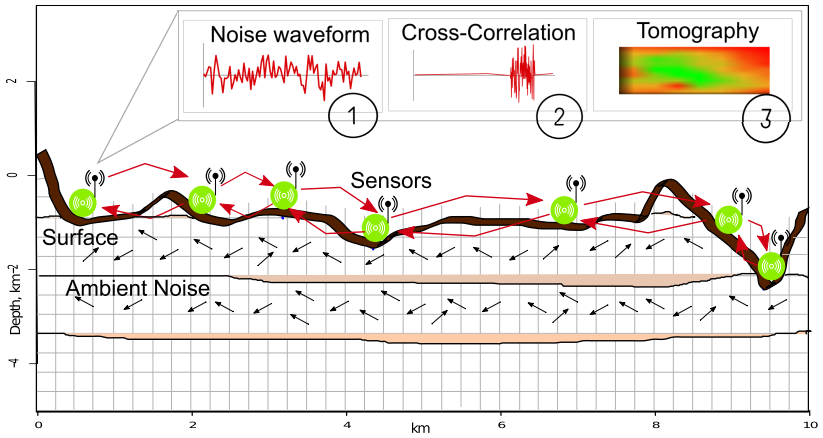


Figure 2.1: Ambient noise seismic imaging process.

2.2.1 Signal pre-processing

The ambient noise raw data gathered from each individual sensor need to be prepared to get a suitable individual waveform for future cross-correlation. As explained in [43], the purpose of this preparation is to accentuate ambient noise by attempting to remove earthquake signals and instrumental irregularities that tend to hide ambient noise. The signal preparation has three important steps: (i) removing instrumental error response and cutting data; (ii) time-domain normalization and (iii) spectral whitening.

To remove instrumental irregularities, the first step is to remove the mean and the trend of the signal. Then a taper is applied to improve signal properties in the frequency domain [45]. A simple cosine taper filter that applies cosine-shaped attenuation function to specified frequencies at low and high frequencies is applied to remove instrument irregularities. Additionally, the data should be cut into a specific time-window to be analyzed in a window fashion. Data can be cut on one day, some hours, o few minutes. This window of time λ will be used for posterior steps (cross-correlation) and stacked together until complete the total time T of the signal.

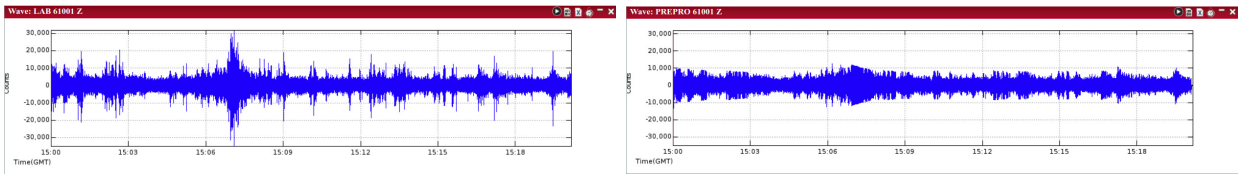
The next step is time-domain normalization, also called temporal normalization [43]. The

time-domain normalization we use is running-absolute-mean normalization [43]. This method computes the running average of the absolute value of the waveform in a normalization time window of fixed length, and it weights the waveform at the center of the window by the inverse of this average. Given a discrete time-series f, the normalization weight is

$$w_n = \frac{1}{2N+1} \sum_{i=n-N}^{n+N} |f_i|, \tag{2.1}$$

and the normalized datum is $\tilde{f}_n = f_n / w_n$. The width of the normalization window is $2N+1$, and it is used to determine how much amplitude information is retained. The size of N depends on the half of the maximum period of a bandpass filter. Fig. 2.2 shows an example of how the preprocessing methodology works.

Finally, a spectral normalization is applied. Spectral normalization seeks to reduce broad imbalances in single-station spectra to aid in the production of a broad-band dispersion measurement [43]. Inversely weighting the complex spectrum by a smoothed version of the amplitude spectrum produces the normalized or whitened spectrum. This process is similar to the temporal normalization but using frequency domain spectrum.



(a) (b)

Figure 2.2: (a) Raw data (b) Data after temporal normalization.

2.2.2 Signal Cross-Correlation and Green's functions

After pre-processing of the raw seismic data, the next step to get travel time between two nodes is to apply cross-correlation and stacking processes. Cross-correlation is a common

method to process ambient noise data. The cross-correlation should be symmetric as the positive and negative lag signals are averaged. The result of cross-correlation has a positive correlation of the Green's function, and this contains the information of group velocity and phase velocity at different frequencies [46].

Theoretical work by [4] describes how to estimate the Green's function $G_{AB}(t)$ between nodes A and B using the ambient noise cross-correlation $C_{AB}(t)$ between them:

$$G_{AB} = -\frac{d}{dt}\left[\frac{C_{AB}(t) + C_{AB}(-t)}{2}\right] \qquad 0 \leq t < \infty \tag{2.2}$$

To obtain the unbiased phase and group velocity measures, the cross-correlation should be transformed to the Green's function using equation 2.2.

2.2.3 Stacking

The stacking process is usually employed to increase the signal-to-noise ratio (SNR) of the signal [47]. In this case, we stack the cross-correlation results every time it is performed (every λ minutes). The stacking means superposition and summation. Before stacking, the cross-correlation is normalized by the maximum amplitude. As a side note, we want to mention that there are usually three channels in seismic sensors that measure vibration on X, Y, Z direction respectively. In this work, we focused on the vertical component Z because our main interest is the Rayleigh waves. In literature, it is well-known that long-range coherent noise can be found on the vertical component [48–50]. However, this work can be easily extended to be used with horizontal components too. The important part of working with the horizontal component is the rotation of the data in the radial/transverse coordinates. To make the process almost the same, we can borrow the idea of Lin et al. [4] and postpone the component rotation until after the cross-correlation by allowing east and north components temporally normalized together. The distributed sensor network would be the same, only

adding an extra-step in the preparation of the data is needed.

2.2.4 Frequency-time analysis

Frequency-time analysis (FTAN) generates the dispersion curve of the Rayleigh wave phase velocity [43]. A whole FTAN process includes: a series of Gaussian band-pass filters to get Green's function with different central frequencies and transformation processes to get the envelope function and phase function of time series data. With the envelope function and phase function, we can generate a figure called FTAN map with the x-axis as apparent period and the y-axis as group velocity. The local maximum point of this map represents the travel time t_{max} between two nodes [4]. The value of the phase function at t_{max} can be used to determine phase velocity.

The summary of the signal processing analysis to obtain the travel time measurements can be seen in this way: suppose two nodes (A and B) need to correlate their ambient noise signals to obtain the travel time measurement between them. The pre-processing process includes performing a uniform down-sampling of the signal (DS), applying data preparation (Pre) as explained in section 2.2.1 and compressing the signal (Cmp) using a compression library. We use *zlib* data compression algorithm [51] and we achieve a compression rate of $\sim$ 50%. If an initial bandpass filter is applied (BP) the compression rate is higher. The BP application is configurable is a configuration file. After the communication of this pre-processed data, we perform the cross-correlation ($\bigotimes$). Every λ minutes (for our test we select 5 minutes due to experts' recommendations[2]) the process is repeated and the cross-correlations are stacked (S). Then a narrow band-pass filter (NBP) is applied at different frequencies.

[2]In literature, different cutting window sizes for cross-correlation has been used; for example, 1-minute window [52] or 30-minute window [53]. We chose 5-min after consulting with Dr. Fan-chi Lin, one expert on ambient noise tomography, and after doing empirical tests of suitable package size for network transmission. However, this size is configurable in the system.

Frequency-time analysis (FTAN) is then applied and we obtain the travel time measurements at different frequencies.

2.2.5 Eikonal Tomography

The method of eikonal tomography does not need an initial model of the medium for computing. It only needs the travel times between each pair of stations. The gradient of the travel times provides information about local direction and travel of the wave, hence, deriving phase velocity maps is possible.

Eikonal Equation

Once the travel time $\tau(r_i, r)$ are known for positions r (arbitrary point in the medium) relative to a node r_i, the eikonal tomography is performed. The eikonal equation[6] is based on the solution of Helmholtz equation:

$$\frac{1}{c_i(r)^2} = |\nabla\tau(r_i, r)|^2 - \frac{\nabla^2\Lambda_i(r)}{\Lambda_i(r)\omega^2}. \tag{2.3}$$

At high frequencies, when the second right-hand term is small enough, it can be dropped as:

$$\frac{\hat{k}_i}{c_i(r)} \cong \nabla\tau(r_i, r), \tag{2.4}$$

where, c_i is the phase velocity for event i at position r. $\hat{k}_i$ is the unit wave direction vector for the event i at position r. ω is the frequency, and Λ is the amplitude of an elastic wave at position r. The gradient is computed relative to the field vector r. Equation 2.4 is derived from equation 2.3 by ignoring the second term from the right-hand side. These conclude that the gradient of the travel time is related to the local slowness (1/velocity) at r position, and the direction of propagation of the wave (azimuth) denotes the local direction of the wave. Dropping the second term on the right-hand side of equation 2.4 is justified when either the

frequency is high or the amplitude variation is small[6]. When eikonal tomography is used, there is no need for a tomographic inversion because taking the gradient of the phase travel time surface gives the local phase speed as a function of the direction of propagation of the wave.

Isotropic Wave Speeds

Applying eikonal equation 2.4 can introduce some errors and usually the phase velocity map is noisy due to imperfections in travel time surface calculation. To overcome this issue, a mean slowness and its standard deviation are calculated in order to obtain the isotropic phase speed.

Traditionally, to compute phase velocity maps through eikonal tomography we need the following: (i) to generate a grid of arbitrary points (r) in the field through interpolation of travel times; (ii) to construct a phase travel time surface for obtaining slowness and azimuth vectors in every effective source relative to each arbitrary point in the grid; (iii) to calculate the mean slowness and standard deviation of the phase travel time surface to overcome errors; and (iv) to invert the final slowness vector to obtain the velocity map.

In the next section, we explain how we design a distributed system for obtaining velocity maps from a series of raw data recording from ambient seismic noise. The centralized approach is also explained to further comparison.

2.3 Distributed System Design

During the ANSI computing, two phases need message exchange between nodes. In the first phase, called the **correlation phase**, nodes communicate every λ minutes to cross-correlate its pre-processed data with those from its neighbors; here, there is no need of distributed computation because nodes compute locally their results and only talk to neighbors for

sending pre-processed information. In the second phase called the **imaging phase**, nodes calculate its partial maps locally, and then communicate these results to produce the final velocity map; here, a distributed approach to implement in sensor networks is required.

We provide a detail description of each phase; and particularly for **imaging phase**, and we formulate the distributed problem to aggregate the final velocity map. We also compare this distributed approach with a standard centralized solution.

2.3.1 Correlation Phase

The overview of the correlation phase is described in Fig. 2.3. In this phase, every node reads raw data from a medium; for example, a seismic sensor reads seismic waveforms in a field. Once the node has completed λ minutes of the reading process, it activates the next steps: preprocessing, communication, cross-correlation and stacking process. Note that the reading process is continuous and the other processes are done in parallel when they are activated. The pre-processing of the data is made in-situ, and it consists of preparing waveform data from each node individually.

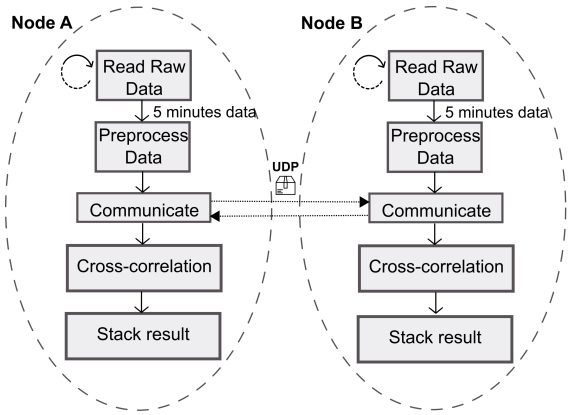


Figure 2.3: System design of Phase 1 (Correlation phase).

After the preprocessing, the node compresses the data into an UDP (User Datagram Protocol) package[3] and broadcasts the package to its neighbors. The node is also receiving

[3]The maximum size of the UDP package is 65KB. However, with the compression technique (using *zlib*)

pre-processed data from its neighbors. Notice that this communication process may be asynchronous, and the system is able to handle this situation by using a time-stamp inside the UPD package to let the nodes know which data to correlate. Every node cross-correlates its data with each one of its neighbors and stacks it. The stacking process is referred to add the results up for each λ minutes already processed. Algorithm 1 presents the detail process for correlation phase.

Algorithm 1 Correlation phase algorithm

```
Define T (Total time for stacking cross-correlation)
Define window size λ
Activate thread reading and thread correlate
Begin thread reading
while T − λ ⩾ 0 do
    Read data D from medium;
    if size of D is corresponding to λ then
        Activate thread prepare
    end
End thread reading
Begin thread prepare
    Apply down-sampling to D
    Remove instrument noise in D
    Apply Taper process in D
    Apply Frequency whitening process in D
    Compress data D
    Add time-stamp to the compress data D
    Broadcast D
End thread prepare
Begin thread correlate
    Receive data from neighbor i (D_i) and verify time-stamp
    if time stamp in D_i is equal to time-stamp in D then
        Decompress D_i
        Cross-correlate D_i and D
        Stack cross-correlation D_i and D
    end
End thread correlate
```

In the algorithm 1, T is the total time we need for stacking results of the cross-correlation. Usually, for getting meaningful velocity maps, we need to stack hours to weeks, depending

and depending the band-pass filter is applied in the data preparation stage, we achieve a compression rate between 50% and 70% which is significant, and it helps to meet bandwidth constraints.

on the node spacing and noise condition, of cross-correlated data [43]; hence, this parameter is configurable in the system. For our experiments, we stacked one week of cross-correlated data and we got the velocity maps; however, the system correlate in real time, hence, we can generate the velocity map at any moment. λ is the windows size for cross-correlation. We used a window size of five (5) minutes, but this parameter is also configurable in the system. For performing cross-correlation between data of different nodes, we use three different threads. The first thread is called *reading*, and it is responsible for reading data continuously and every λ (5) minutes activating the thread for pre-processing and broadcasting data. The thread for pre-processing and broadcasting data is called *prepare*, and it applies pre-processing techniques to the signal, compresses the data (through compression libraries), creates the package to send, and broadcasts that package to its neighbors. At the same time, the thread *correlate* is listening for receiving packages from neighbor nodes. Once it receives a package, it decompresses the package and verifies the time-stamp to correlate the package data with its own data. Finally, the thread stacks the correlated results for each neighbor. The output of this algorithm is a set of cross-correlated signals between a node and its neighbors.

2.3.2 Imaging phase

The overview of the imaging phase is described in Fig. 2.4. After completing the correlation phase, every node has a set of correlated signals between its neighbors and itself. The next step is to apply FTAN techniques to obtain travel times (τ) from the cross-correlation results. Notice that every node calculates individually the travel time between itself and each one of its neighbors.

After travel time calculation, every node constructs its own phase travel-time surface based on its travel time measurements. To construct the phase travel-time surface, the node needs to interpolate its travel time data onto a finer and regular grid. Algorithm 2 describes the process for calculating the phase travel-time surface in each one of the nodes.

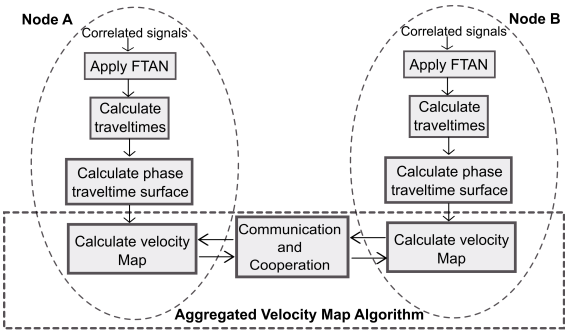


Figure 2.4: System design of Phase 2 (Imaging phase).

Algorithm 2 Phase-velocity Travel-time Algorithm (PTT)

1: Input: travel-time measurements τ of node i
2: Interpolate all τ of i onto a $Gx^\circ \times Gy^\circ$ grid size $x \times y$
3: Perform second interpolation of τ with extra tension
4: **for** each point k in the interpolated grid **do**
5: Calculate $\nabla\tau_k$
6: Calculate Slowness S_k
7: Calculate Azimuth A_k
8: **end**
9: Output: S and A vectors for node i

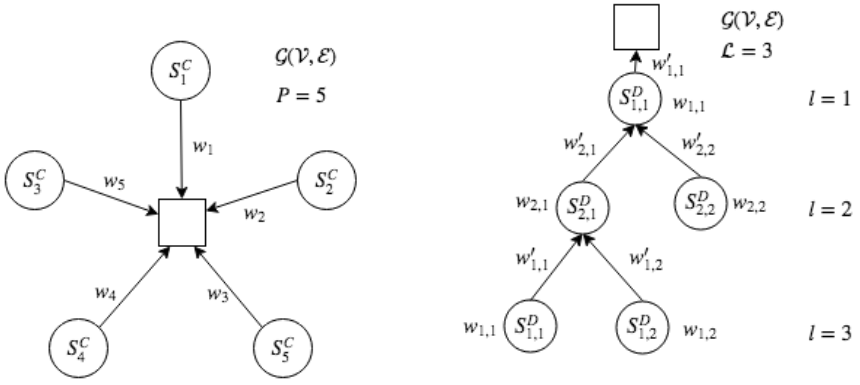
Here, every node i executes a interpolation of its travel time measurements in a grid $Gx^\circ \times Gy^\circ$ of size $x \times y$ to get a phase travel-time surface. This grid depends directly on the location of the sensors (nodes) in real field. For example, our simulated study, we used a grid of $1e^{-6^\circ}$ by $1e^{-6^\circ}$ because our real experiments are located in Sweetwater, Texas. For our real deployment, we used a grid of $2e^{-5^\circ}$ by $2e^{-5^\circ}$ because the deployment location only uses ten sensors in a smaller area. However, these parameters are fully configurable in the system. Details of how to fit this grid can be found in [4, 54]. In general, we need to choose an adequate finer, regular grid. The degrees depend on the distance between stations. Larger distances will have higher degrees. The grid also needs the minimum and maximum latitude and longitude to calculate the square regular grid to interpolate.

In the next step (line 4), the gradient of each travel-time surface is computed at each

spatial node. Using the eikonal equation (equation 2.4), the magnitude of the gradient allows to calculate the local phase slowness (S), and the direction of the gradient can be used to estimate the azimuth (A).

Once every node completes the calculation of the local phase slowness and azimuth vectors, the second round of communication between nodes begins to calculate a velocity map. Therefore, we need a technique for aggregating partial information of slowness and azimuth inside each node into a final phase velocity map.

There is exists different approaches to aggregate information on sensor networks. The common one is the centralized approach, where all nodes send its data to a central server or SINK. An example is shown in Fig. 2.5(a). However, the centralized approach introduces a high communication cost in the network, and it is unsuitable for real-time systems. In the distributed approach, Fig. 2.5(b), an aggregation tree is constructed for aggregating the partial maps into the final tomography. We are aware of consensus techniques for reaching tomography consensus on sensor networks [1,55,56]; however, we chose tree-based aggregation because is faster on real-time systems.



(a) (b)

Figure 2.5: Topology comparison from centralized imaging and tree-based distributing imaging.

We formally introduce the centralized and distributed approaches in the following way: Let

$\mathcal{G}(\mathcal{V}, \mathcal{E})$ denote an undirected connected graph (network) with node (sensor) set $\mathcal{V} = (1, ..., P)$ and edge set $\mathcal{E}$, where each edge set $\{i, j\} \in \mathcal{E}$ is unordered pair of distinct node.

Centralized Imaging

In the centralized approach, the velocity map $\mathfrak{V}^c$ is calculated based on the slowness observations of $\mathcal{V}$ in the whole network $\mathcal{G}$ as:

$$\frac{\mathbf{1}}{\mathfrak{V}^C} = \tilde{S^C} = \text{argmin} \sum_{i=1}^{P} ||\tilde{S^C} - w_i S_i^C||, \tag{2.5}$$

where S_i^C is the slowness calculation of node i in a centralized fashion, which means the calculations are already collected in a server. We want to minimize the difference between the estimated final slowness $\tilde{S^C}$ and the aggregated slowness of all nodes $\mathcal{V}$. Note that P is the total number of nodes in the network $\mathcal{G}$. A vector of all ones is defined as $\mathbf{1}$. Because all the slowness measurements are located in a central node, the velocity map $\mathfrak{V}^C$ is reliable and has good resolution. However, the cost of transmitting all measurements to a central place can be significantly high (see Section 2.5.3.)

Distributed Imaging

In the distributed approach, node i carries out communication only with its neighbors $\mathcal{N}_i = (j|\{i, j\}) \in \mathcal{E}$. A randomly formed tree is created to aggregate the velocity map in a bottom top fashion. In this tree-based approach, the aggregation is performed by constructing an aggregation tree, which in this case is formed randomly by pooling neighbor nodes. The flow of data starts from leaves nodes up to the sink and therein the aggregation done by parent nodes.

Let $\mathcal{L}$ be the number of levels of the aggregation tree. Let $S_{l,i}^D$ be the slowness vector or "partial map" of node i in the level l, where $l \in L$ and $i \in \mathcal{V}$ to be aggregated in a distributed

fashion. $|S^D_{l,i}|$ is the number of children nodes of node i in the level l.

The final velocity map $\mathfrak{V}^D$ is estimated as:

$$\frac{1}{\mathfrak{V}^D} = \tilde{S}^D = \text{argmin}\,||\tilde{S}^D - U_{\text{root}}||, \tag{2.6}$$

where $U_{root} = U_{1,1}$ is the final slowness vector after the aggregation process at the root node. The process starts from leaves in a bottom-top fashion, and $U_{l,i}$ is calculated for each l level and i node in the level as follows:

$$w'_{l,i} = \begin{cases} w_{l,i} & \text{if } \left|S^D_{l,i}\right| = 0 \\ \sum_{j=1}^{|S^D_{l,i}|} w_{l+1,j} + w_{l,i} & \text{otherwise} \end{cases} \tag{2.7}$$

$$U_{l,i} = \begin{cases} S^D_{l,i} & \text{if } \left|S^D_{l,i}\right| = 0 \\ \mathcal{Z}_{l,i} & \text{otherwise} \end{cases} \tag{2.8}$$

$$\mathcal{Z}_{l,i} = \frac{\left(w_{l,i} * S^D_{l,i}\right) + \sum_{j=1}^{|S^D_{l,i}|} w'_{l+1,j} U_{l+1,j}}{w'_{l,i}}, \tag{2.9}$$

where $w_{l,i}$ and $w'_{l,1}$ are the original weight assigned to the partial map and the weight after aggregation respectively. The original weight is assigned in eikonal tomography, and it is related to the azimuth vector [6]. After the aggregation process, the root node contains U_{root} that we can consider the final velocity map.

We design an algorithm to aggregate the information, manage cooperation between nodes, and communicate in a tree structure through broadcasting.

In this approach, we can view the broadcast as a Breadth-First Search (BFS) in the network. Every node i is associated with its level $l(i)$. This level is the length of i shortest path to the root, and it is computed during BFS in the spanning tree. At the beginning, the level of every node i is $l(i) = \infty$, and the level of the root r is $l(r) = 1$. For creating the

spanning tree, the r node makes a broadcast. This message contains the root level $l(r) = 0$. When a node i receives a message from a node j contained $l(j)$, i checks its $l(i)$ value to see if $l(i) = \infty$. If this happens, i sets its level to $l(i) = l(j) + 1$ and forward the query to its neighbors; otherwise, i stores the level of j as $l(j)$. The tree has been formed when all l-values are less than ∞. An example of tree is shown if Fig. 2.6.

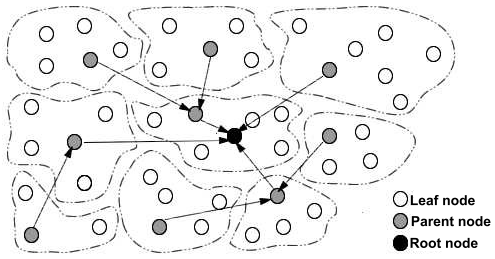


Figure 2.6: Example of tree in the aggregation process [1].

The algorithm then computes the aggregation function in a bottom-up fashion in the spanning tree. Every node has to wait until it receives the information from its children, or reaches maximum time. Then, the node aggregates the information and sends it to all of its parents in the tree. In the end, the final velocity map is displayed in the root node.

Algorithm 3 is called *aggregated velocity map algorithm (AVMA)*, and it illustrates the process from a single node point of view. First, each node sets its own level in ∞ to start forming the tree. Then, if the node is selected as root, it sets its $l(i)$ in one. In the emulation scenario, the root node is selected randomly. In a real scenario, the root node is selected during the deployment process. If the node is a root, it waits until receiving the aggregated information from its children (or, alternatively, reaches maximum time) and creates the final phase velocity part by inverting the final slowness vector (line 10). If the node is not a root, it can be either a leaf or a parent node. We know a node is a leaf when it does not have any child. At the same time, each node registers its father in the tree. If the node is a leaf (lines 15-16), it just needs to send its information to its father. If the node is a parent node (lines 17-25), it broadcasts its level $l(i)$ and waits for aggregating the information of its children

Algorithm 3 Aggregated velocity map algorithm (AVMA)

1: Define i as node number in the network.
2: Initialize node level $l(i) = \infty$
3: **if** i is selected as root **then**
4: Set $l(i) = 0$
5: Broadcast $l(i)$ to its neighbors $|N_i|$
6: **While** until receive from all children or reach MaxTime
7: Receive $(S_j(t),A_j(t),w_j(t))$ from j where $j \in |N_i|$
8: Update $S_i(t+1),A_i(t+1),w_i(t+1) =$ AGG$(S_j(t),A_j(t),w_j(t)$ and $S_i(t),A_i(t),w_i(t))$
9: **end while**
10: Calculate final phase velocity map $\mathfrak{V}^D = \mathbf{1}/S_i(t)$
11: Output $\mathfrak{V}^D$
12: **else**
13: Receive $l(j)$ where $j \in |N_i|$
14: Set own level $l(i) = l(j) + 1$
15: **if** $i = leaf$ **then**
16: Send $(S_i(t),A_i(t),w_i(t))$ to its father
17: **else**
18: Broadcast $l(i)$ to neighbors $|N_i|$
19: **While** until receive from all children or reach MaxTime
20: Receive $(S_i(t),A_i(t),w_i(t))$ from j where $j \in |N_i|$
21: Update $S_i(t+1),A_i(t+1),w_i(t+1) =$ AGG$(S_j(t),A_j(t),w_j(t)$ and $S_i(t),A_i(t),w_i(t))$
22: Send $(S_i(t),A_i(t),w_i(t))$ to its father
23: **end while**
24: **end if**
25: **end if**

with its own information. Once the parent node finishes the aggregation process, it sends the results to its father (line 22).

The aggregation process (AGG$(S_j(t),A_j(t),w_j(t)$ and $S_i(t),A_i(t),w_i(t))$ defined in lines 8 and 21 is not a trivial addition of values. For performing the aggregation of slowness and azimuth vectors, we use statistical averaging. The statistical average used is a weighted average where the weight is calculated based on azimuth vector collected from node i and j (See Section 2.3.2).

2.3.3 Aggregation vs. Consensus for Imaging Phase

For computing distributed eikonal tomography, we proposed two distributed approaches: aggregation, which aggregates information in a bottom-up fashion (Section 2.3.2), and consensus, which uses information of neighbors for computing a consensual average. In this section, instead to use Algorithm 3 to aggregate the final velocity map in a distributed fashion, we use a consensus algorithm to estimate the final velocity map by using only information exchange between neighbors.

In consensus algorithms, every node over the network has a initial value or group of values (measurements), and they aim to calculate the average of all these values through a distributed linear iteration method. A major advantage of these algorithms is they can calculate an average of the network measurements iteratively and distributively using local information exchange among neighbors and calculation of weighted sums at every node. Some papers have shown that consensus algorithms are effective for obtaining consensus values over a network[55–57].

The main idea is described as follows. Consider a graph $G = (V, E)$ where $V = \{1...n\}$ represents the vertices or nodes in the network, and $(i, j) \in E$ represents the edges or path of communication between nodes i and j. Let N_i be the set of neighbors of node i and $N_{[i]}$ be the neighborhood of node i. Every node i initially has a local value $S_i \in \mathbb{R}$. The goal is to compute $\bar{S} = \frac{1}{n}(\sum_{i=1}^{n} S_i)$ in a distributed fashion. Let $x_i(0) = S_i$. At time $t + 1$ every node transmits its current estimate to its neighbors and then updates its information $x_i(t + 1) = \sum_{j \in N_{[i]}} w_{ij} x_i(t)$ where w is the weighted sum at every node.

Algorithm 4 is called *Consensus velocity map algorithm (CVMA)*. The algorithm starts by setting all nodes of the network with an ID and a initial weight of one. Then, all nodes execute in parallel the *Phase TravelTime Algorithm*. Until they converge or reach the maximum number of iterations, all nodes execute in parallel steps 7 to 10. First, a node select randomly a neighbor and send its slowness, azimuth and weight values (S_k^t, A_k^t, w_k^t) in time t. The

Algorithm 4 Consensus velocity map algorithm (CVMA)

1: Initialize nodesID for all nodes n.
2: Set $w_k = 1$ for all nodes n
3: **for** all nodes n with points k in the grid **do in parallel**
4: Execute PHASE TRAVELTIME ALGORITHM PTT
5: **end**
6: **for** $t \leftarrow 0$ until convergence or maximum number of iteration **do**
7: Select randomly a neighbor and send (S_k^t, A_k^t, w_k^t)
8: Receive $(S_{n_k}^t, A_{n_k}^t, w_{n_k}^t)$ of its neighbor $\in N_{[k]}$
9: Update $(S_k^{t+1}, A_k^{t+1}, w_k^{t+1}) = \text{AGG}(S_{n_k}^t, A_{n_k}^t, w_{n_k}^t)$
10: **end**
11: Calculate final map $(1/S_k)$
12: Output phase velocity map

node then receives the values of its neighbor $(S_{n_k}^t, A_{n_k}^t, w_{n_k}^t)$ in time t. Using the *Isotropic Algorithm*, which has been designed to compute a weighted aggregation of information, every node obtains new averaged values for its slowness, azimuth, and weight $(S_{n_k}^{t+1}, A_{n_k}^{t+1}, w_{n_k}^{t+1})$ at time $t+1$. For verifying convergence, every node measures the relative error between S_k^t and S_k^{t+1}. When the consensus or the maximum number of iterations has been reached, in every node the final map is computed using the inverse of the final slowness vector S_k.

2.4 Deployment 1: USArray

2.4.1 Deployment Specifications

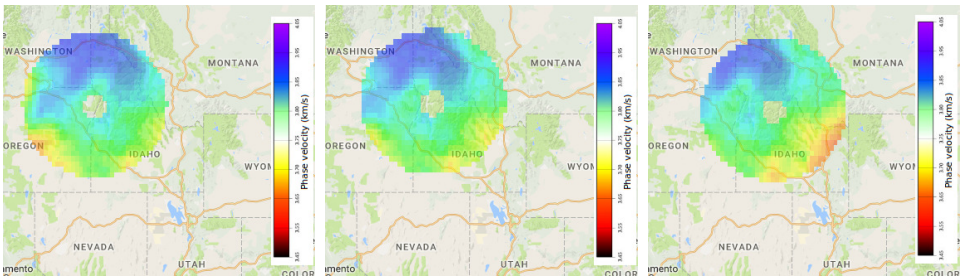
For comparing the two velocity maps algorithms AVMA (Algorithm 3) and CVMA (Algorithm 4), we used data collected from 1211 stations of the USArray Transportable array database (USArray)[4], which records seismic data across the United States. Even though spacing nodes of the USArray is based on radio of kilometers, we used this data to compare final results by assuming that the nodes are spreading in smaller distances. This assumption is

[4]http://www.earthscope.org/science/observatories/usarray

made because eikonal tomography can be very well applied for monitoring shallow subsurface spanning few kilometers and a dense array of stations can be easily deployed with space in 10s meters[58].

2.4.2 Results

In this experiment, we setted up nodes on the CORE emulator to compute eikonal tomography for ANSI. Every node can be considered as a effective central node. Results of these partial surfaces after applying algorithm 2 are shown in Figure 2.7.



(a) (b) (c)

Figure 2.7: Partial maps on every virtual receiver station. (a) Station G10A partial velocity map. (b) Station G11A partial velocity map. (c) Station G12A partial velocity map.

Observe that every effective central node computes its partial map using the traveltimes available from its neighbors.

We computed AVMA and CVMA algorithms for getting final phase velocity maps over the network. Results of applying each method are shown in Figure 2.8. Notice that Figure 2.8(a) refers to the centralized approach, Figure 2.8(b) shows the AVMA algorithm result, and Figure 2.8(c) the CVMA algorithm result.

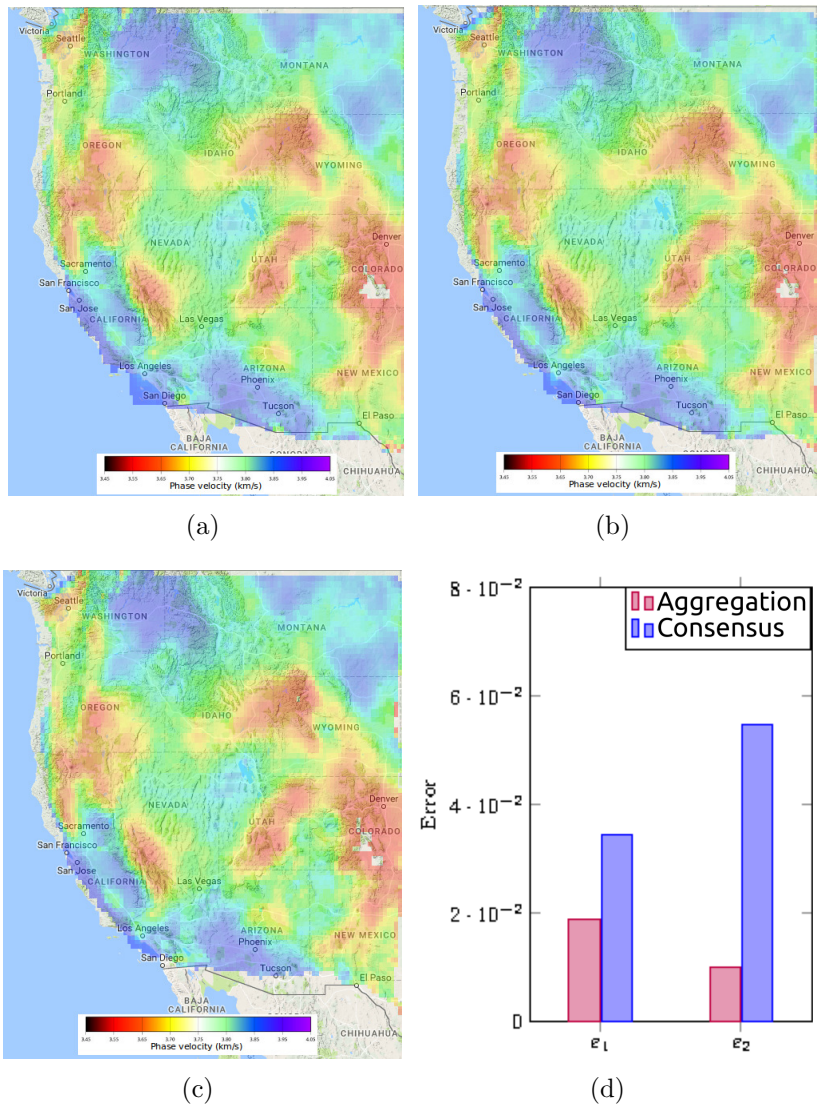


Figure 2.8: Comparison between final phase velocity maps. (a) Centralized Method. (b) AVMA Algorithm. (c) CVMA Algorithm. (d) Measures of Distance from centralized approach

2.4.3 Evaluation

Accuracy

To evaluate the aggregation algorithm (AVMA) and the consensus algorithm (CVMA), we compared them with the centralized approach in which all nodes send the information to a

central unit. Using $\tilde{m}$, m^* and $\bar{m}$ to represent the centralized model, the proposed distributed model and the mean value of m^* respectively, we used the following quantitative measures of distance from the centralized model to evaluate the estimation quality

$$e_1 = \left[\sum_{i=1}^{n} (\tilde{m}_i - m_i^*)^2 / \sum_{i=1}^{n} (m_i^* - \bar{m})^2 \right]^{1/2}. \tag{2.10}$$

$$e_2 = \sum_{i=1}^{n} |\tilde{m}_i - m_i^*| / \sum_{i=1}^{n} |m_i^*|. \tag{2.11}$$

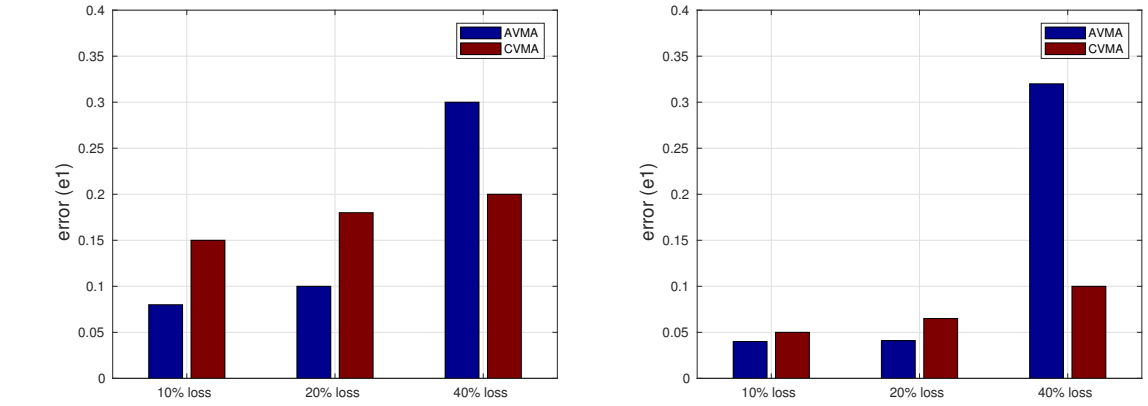
These represent the normalized root mean squared distance and the average value distance, respectively. Results are shown in Fig. 2.8(c).

We can conclude that both methods produce similar results to the centralized approach in terms of image reconstruction. Note that the distance from the centralized approach is higher in the CVMA algorithm. This can be due to the probabilistic nature of the algorithm since it selects randomly one neighbor node to exchange data in each iteration. After some iterations certain nodes may only exchange information with the same neighbor. On the other hand, distance from centralized approach is considerably less in the AVMA algorithm. Basically, this is because AVMA uses all the information of the nodes to compute an aggregated map.

Robustness and Loss Tolerance

In the next set of experiments, loss tolerance and robustness of AVMA and CVMA are evaluated. The algorithm runs with the same configuration for three different packet loss ratios of 10%, 20% and 40% in the emulator and the results are shown in Figure 2.9.

Note that even though AVMA produces images near to the centralized approach, when the packet loss increase the error increase considerably, which means that the results is less accurate. Note also that, as expected, CVMA performs well under package loss constraints. Even with 40% fewer transmitted packages, CVMA is able to recover a closer phase velocity



e1

(a) (b)

Figure 2.9: Distance from centralized approach with packet loss. (a) Error e_1. (b) Error e_2

maps compared to centralized approach. This is mainly because in CVMA algorithm, every node spreads its information to its neighbors in every iteration.

Figure 2.10 shows the 2D tomography with packet loss and we can see that with 10% or even 40% packet loss, there is no significance difference in terms of the image reconstruction when compared to the results with no packet loss. We selected Figure 2.10 as a zoom of one part of the maps in Figure 2.8(b) and 2.8(c). The black line is the ground true area of study obtained with the centralized approach. Notice that in both algorithms, this area shows small variation respecting centralized results even with severe package loss.

Communication Cost

In this section, we compared the communication cost of the centralized algorithm with proposed distributed algorithms in terms of number of messages exchanged to reach the solution. This is computed based on the number of messages that every node receives during the computation. From Figure 2.11(a) we can see that communication cost in a centralized

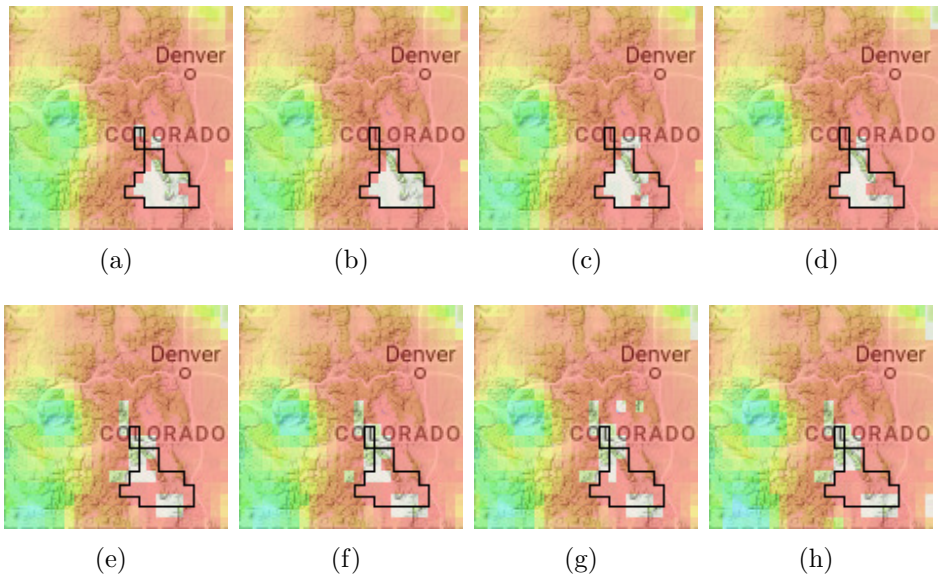


Figure 2.10: Loss tolerance and robustness of AVMA and CVMA with a map with incomplete parts. (a) AVMA no package loss. (b) AVMA 10% Loss. (c) AVMA 20% Loss. (d) AVMA 40% Loss. (e) CVMA no package loss. (f) CVMA 10% Loss. (g) CVMA 20% Loss. (h) CVMA 40% Loss.

setup is high near the "central node" as all the slowness, azimuth information is transferred over the network before computation. Figure 2.11(b) shows the communication pattern for AVMA. Notice that the communication cost is less compared centralized scheme. This is mainly because AVMA distributed the computation load in trees of nodes. In Figure 2.11(c) we present the communication pattern of CVMA algorithm. Observe that this communication pattern is almost flat which indicates CVMA is able to balance the computation load across the network.

The AVMA has acceptable communication cost, since children communicate with their parents only one time. The CISM algorithm is robust even if a node fails, its neighbors have enough information to maintain a reliable final map. However, communication cost is directly dependent on the number of iterations required for reaching consensus, since every node has to communicate many times with their neighbors in order to reach this consensus. So, if the

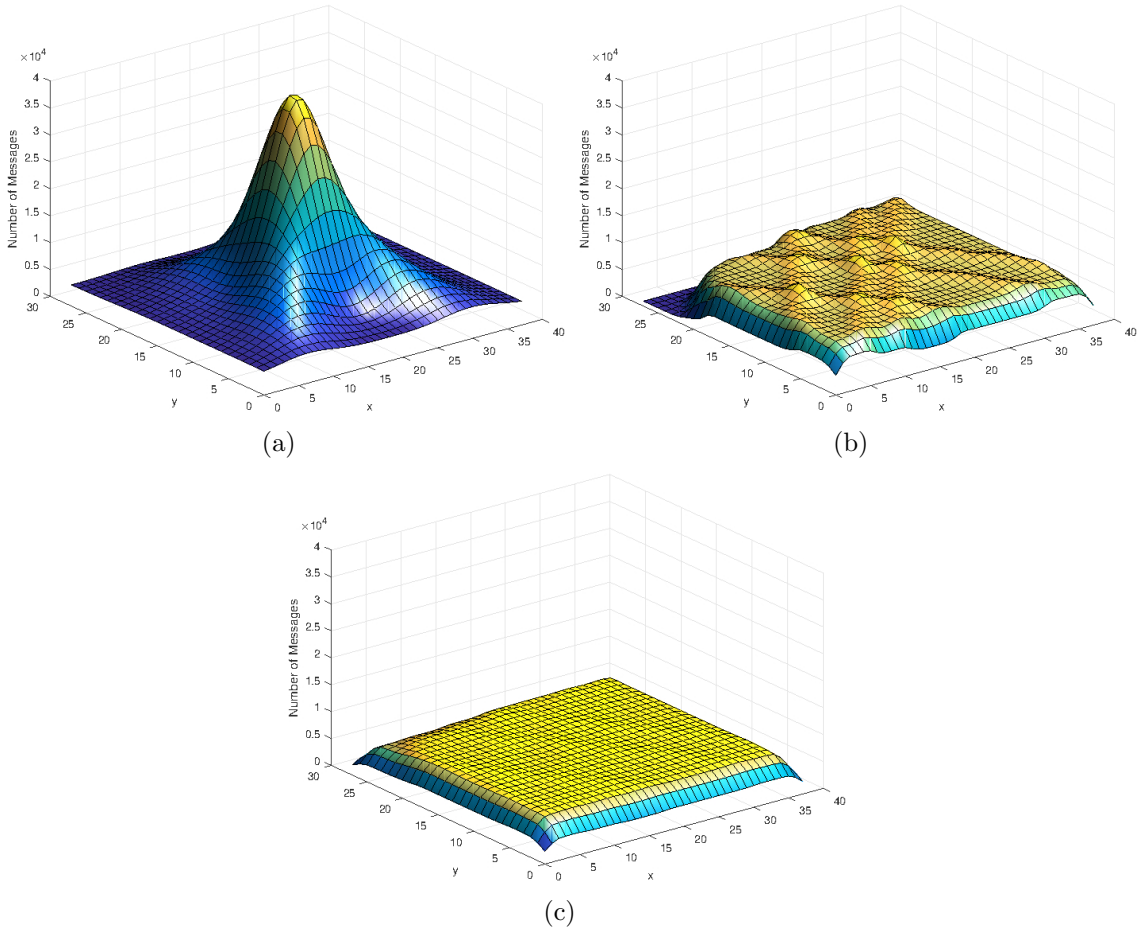


Figure 2.11: Communication cost. (a) Centralized Approach. (b) AVMA Algorithm. (c) CVMA Algorithm. (d) Communication volume

number of iterations is high, the communication cost will be high too.

2.5 Deployment 2: Sweetwater, Texas

We use the distributed eikonal tomography system with the data obtained from sensors deployed in Sweetwater, Texas [59].

2.5.1 Deployment specifications

We selected CORE[5] network emulator [60] for validating our system performance. We used CORE emulator because the code developed over it can be easily transferred to a Linux-based device virtually without any modifications. This property is due to the tool allocates for each network node a Linux virtual machine. CORE will allow us to closely emulate the future deployment because we assume the use of Linux-based, tiny but powerful computational units (e.g. Beagle-bone Black, Raspberry Pi). Once the system was successfully tested on emulation scenarios, we deployed special sensors on the field for system real-test validation.

We validated the system design through the use of a real database of ambient seismic noise. We used a time series data recorded by 75 sensors located in the area of Sweetwater, Texas. The data were recorded between March 21 and March 27, 2014.

We deployed the data of each sensor within virtual nodes in the CORE emulator. We carefully designed the deployment structure to match with the physical location of the real sensors. The deployment structure is an important step as cross-correlation of signals is needed. Fig. 2.12 shows the emulator scenario for Sweetwater database.

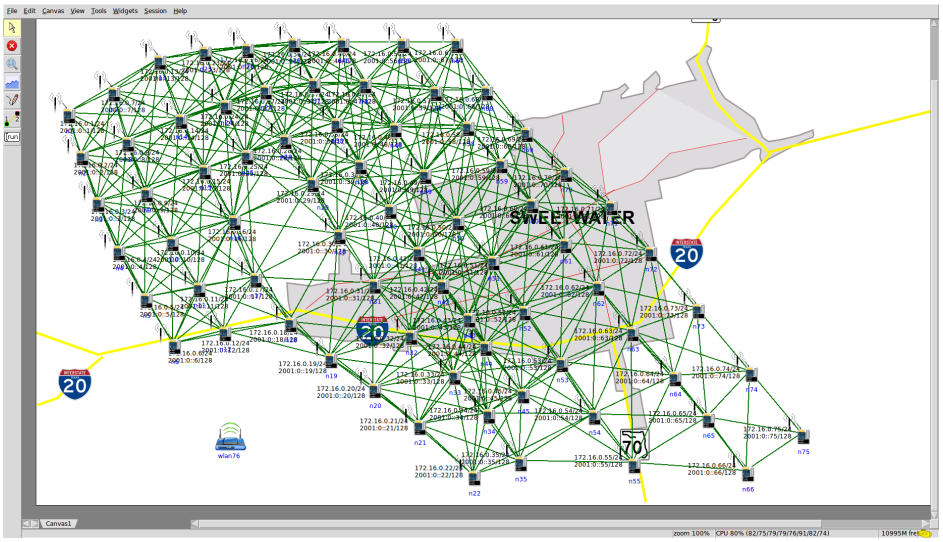
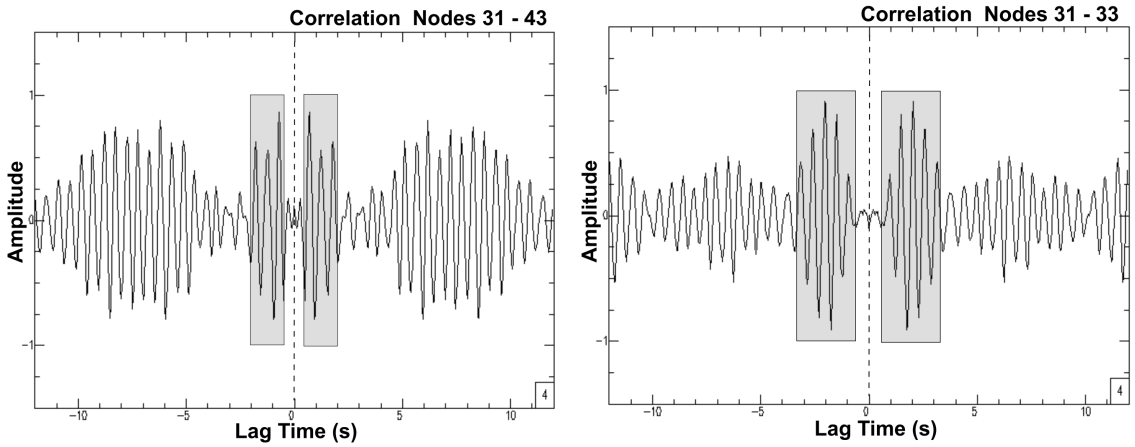


Figure 2.12: A deployment of 75 nodes over Sweetwater area in CORE emulator.

[5]http://cs.itd.nrl.navy.mil/work/core/

2.5.2 Results

As mentioned, the correlation phase is responsible for calculating signal cross-correlation between neighbors. Two examples of our results in the correlation phase are shown in Fig. 2.13. These results were obtained after exchanging pre-processed data with neighbors every five (5) minutes for seven (7) days. From Fig. 2.13(a), we can observe the cross-correlation function result between node 31 (red start on Fig. 2.15(b)) and node 43 (yellow start on Fig. 2.15(b)) calculated by node 31. These nodes represent the physical sensors 6T497 and 6X497 in the Sweetwater deployment. Fig. 2.13(b) illustrates another cross-correlation between node 31 and 33 (black start on Fig. 2.15(b)) calculated by node 31. Node 33 corresponds to the physical sensor 6T536. We configured the system to use a frequency band of 2 Hz.



(a) (b)

Figure 2.13: Cross-correlation results between stations (a) 31 and 43 (1km distance), (b) 31 and 33 (2.5km distance). Grey area represents group velocity arrival of the wave signal. The delay time is shorter in (a) as stations 31 and 43 are physically closer than stations 31 and 33.

These cross-correlation functions are used for every node to apply a frequency-time analysis and obtain the estimated travel time between them (section 2.2.4). When nodes calculate their travel times respective to their neighbors, the imaging phase begins to calculate the velocity map. Results from the velocity map are illustrated in Fig. 2.14. We plotted the velocity map over the real location through `Google Maps`. Fig. 2.15(a) shows the location of

the Sweetwater area. Fig. 2.15(b) illustrates the sensor locations and the final velocity map generated by our system.

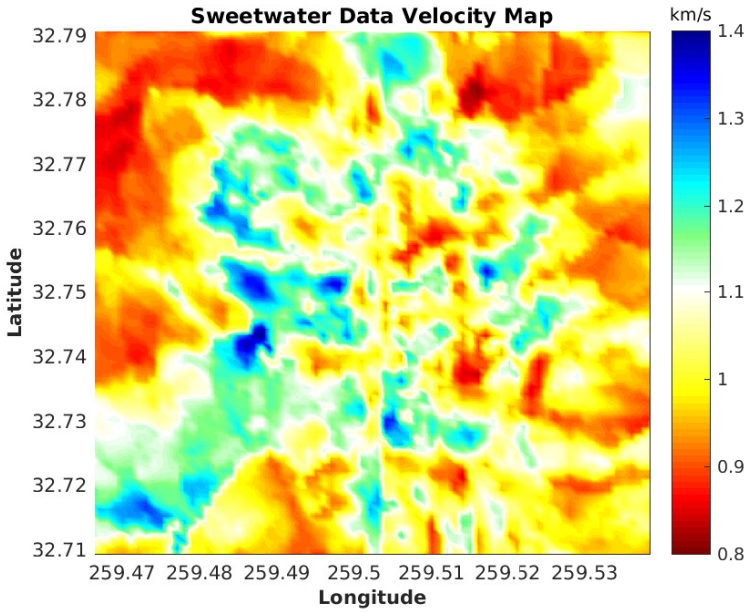
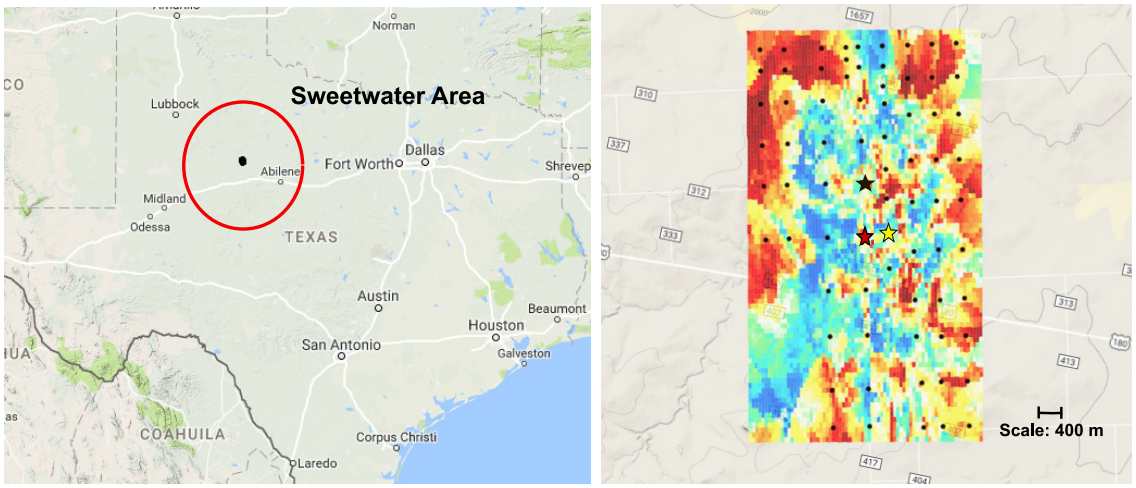


Figure 2.14: Velocity map (dominant frequency 2 Hz) for Sweetwater Data using CORE emulator.



(a) (b)

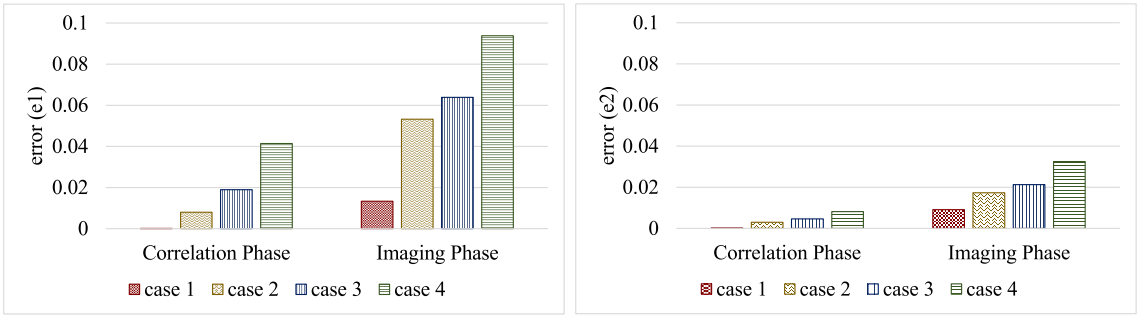
Figure 2.15: Location of the analyzed field data and velocity map obtained from the proposed approach. (a) Sweetwater area (red circle) location over Texas. (b) Zoomed in illustration of stations and the velocity map.

2.5.3 Evaluation

Robustness under unreliable links

To validate our results, we compare the cross-correlation functions and the velocity map, which was generated by our system with the centralized setup. The errors were calculated based on Eq. (2.10) and Eq. (2.11).

Additionally, one of the important characteristics of our system is the fault tolerance, and here, we validated it by simulating node and link failure. We ran both correlation and imaging phases with four different cases: case 1) No failure case; all nodes communicate correctly and generate cross-correlation functions and aggregate the velocity map; 2) 20% of the nodes fail for 20% of the time; 3) 40% of the nodes fail for 20% of the time; and 4) 60% of the nodes fail for 20% of the time.



(a) (b)

Figure 2.16: (a) Error e_1 and (b) Error e_2 for the different cases of fault tolerance.

First, we described the case 1. Case 1 corresponds with no failure in the system which implies every node correctly computes and communicates all the time. Notice from Fig. 2.16(a) and Fig. 2.16(b) that errors e_1 and e_2 for correlation phase are extremely low. This implies that the distributed solution is almost equal to the centralized approach. The same situation can be observed for Case 1 in Imaging phase. Both errors are low because the resultant velocity map is almost identical to the centralized map. Errors are less than 2%.

Since links between node are not always reliable, we design the aforementioned cases 2, 3

and 4. Observe that for correlation phase both errors are very low even when 40% of the nodes fail for 20% of the time. This is due to nodes continuing to correlate every λ (5) minute, and the failures could occur during the inactivity transmission time. Also, the correlation phase is more stable since the stacking process continues stacking results for a long time. However, in the imagining phase, because the aggregation process requires all information of all nodes to be sent to the root node, high failures can significantly increase the error compared to a centralized setup. We plan to overcome this issue by using another technique for combining results in the velocity map such as a consensus algorithm between nodes.

Communication Cost

In this section, we present the communication cost of the two main phases of the ANSI system: correlation and imaging. During the correlation phase, the dissemination of the pre-processed data constitutes the major part of communication. Meanwhile, during the imaging phase, the aggregation of local slowness is the process that communicates the most. We evaluated the communication cost of both phases and compared them with a centralized algorithm. In the centralized scheme for the correlation phase, every node sends its corresponding raw data to a base station, or SINK, placed at the center of the array to cross-correlate all data. For the imaging phase, every node sends its slowness calculation to the same SINK station to calculate the final velocity map.

Fig. 2.17 shows the communication cost in terms of number of received messages for each node during the correlation and imaging phases. Fig. 2.17(a) and 2.17(b) correspond to the correlation phase of (a) the centralized approach and (b) our distributed ANSI system respectively. Similarly, Fig. 2.17(c) and 2.17(b) correspond to the imaging phase of (a) the centralized approach and (b) our distributed ANSI system.

In the correlation phase with centralized setup, all nodes send the raw data to the SINK every five (5) minutes. Fig. 2.17(a) shows the number of messages after one (1) hour

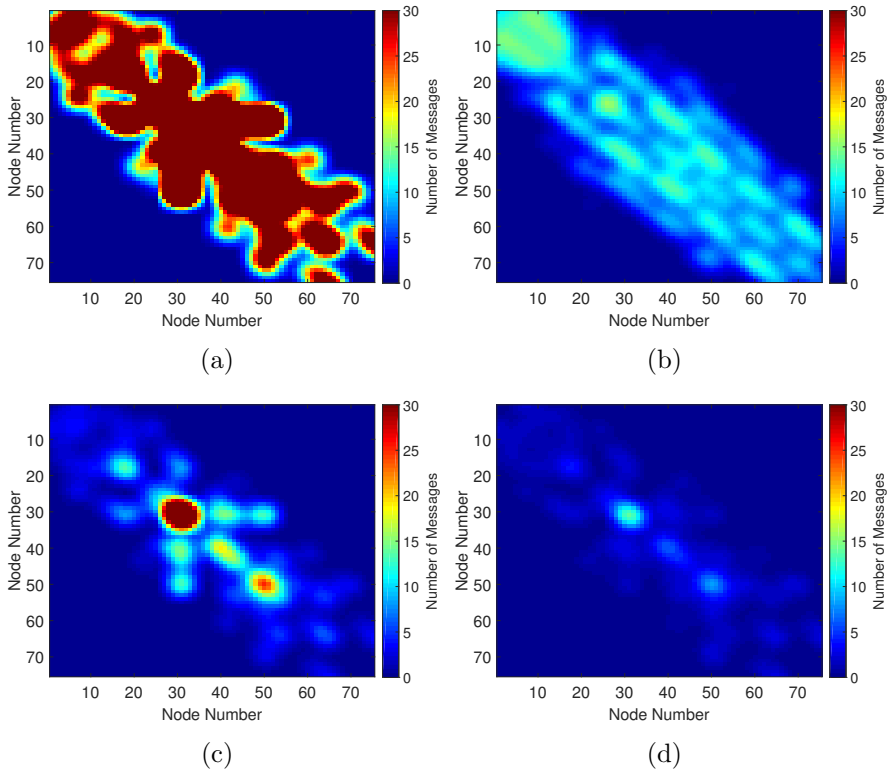


Figure 2.17: Communication cost in number of received messages as 2D heat map: (a) Centralized correlation phase, (b) Distributed correlation phase, (c) Centralized imaging phase, and (d) Distributed imaging phase.

of communication. Fig. 2.17(b) presents the communication cost after one (1) hour of communication in the distributed approach; Here, the communication cost is notably less in the whole network as nodes only share information with neighbor nodes. The number of received messages is directly proportional to the number of neighbors. From our deployment structure (Fig. 2.12), we can see the top left nodes (nodes 1, 2, 3, etc.) have more neighbors than the bottom right nodes (nodes 75, 74, 73, etc); this corresponds to the results obtained in Fig. 2.17(b) where axis `x` and `y` represent the number of nodes in consideration.

In the imaging phase, we measured the communication cost of sending the information of every local slowness to the SINK, and we compared the result with the cost of the distributed

approach. Fig. 2.17(c) illustrates centralized imaging phase. For comparison purposes, we selected as SINK the same node that was selected as root node in the distributed approach. From Fig. 2.17(d), we can observe the distributed imaging phase communicates fewer messages than the centralized. Notice that the communication cost in distributed imaging phase is higher near and around root node, as the root node has more children than parent nodes.

We also evaluate the communication volume of both phases by measuring the number of megabytes transmitted by every node in the network. Fig. 2.18 illustrates the communication volume of both phases. In the correlation phase, the communication volume represents the megabytes transmitted over the network. Observe that in the centralized setup the total volume of communication is around 146Mb for completing 1 hour of cross-correlation results. In contrast, the distributed setup transmits around 50Mb for the same hour of cross-correlation results which implies a reduction of approximately 66%. This is basically due to nodes in the distributed approach cooperating to calculate the cross-correlation with only neighbors. The same situation occurs in the imaging phase; the distributed approach reduces by more than 60% the communication volume compared to centralized setup. However, because distributed approach uses broadcasting to communicate with all neighbors, if we measure the communication volume in the network in terms of Mb received, the centralized approach may have equal or less volume than the distributed approach.

Computational Cost

We evaluated the computational cost of processing correlation and imaging phases by measuring the CPU times in seconds for each one of the nodes. Fig. 2.19 illustrates computational cost measurements for centralized and distributed setups imaging phase. As expected, our distributed system balanced the computational cost as every node calculates its own results and shares them to cooperate in the final result.

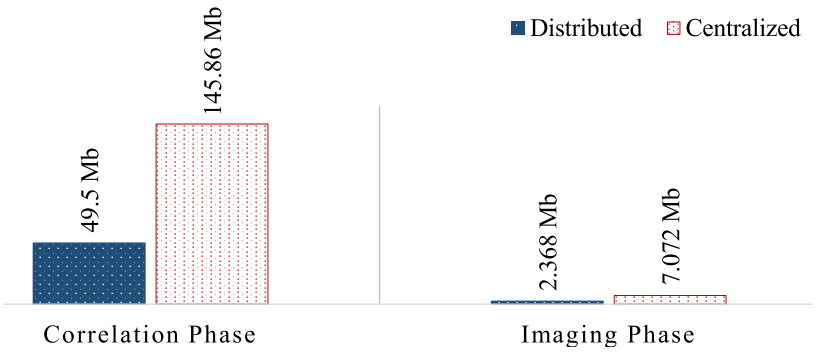


Figure 2.18: Communication volume in the network. Correlation phase (after one hour of cross-correlation) and Imaging phase.

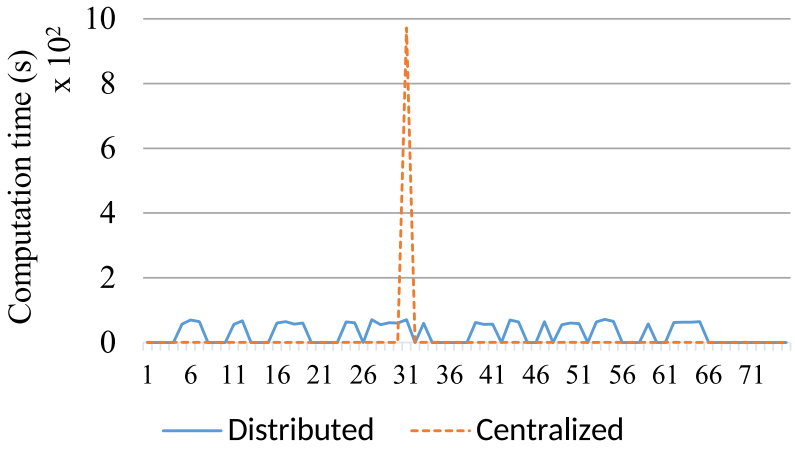


Figure 2.19: Computational Cost. X-axis represents number of nodes.

Effects of Topology

Topology plays a key role in distributed systems like the one presented in this paper. Since the method communicates only with its immediate neighbors, the topology decides how fast the information diffuses in the network. Therefore, the correlation phase on a strongly connected topology generates more information of travel times from immediate neighbors and it may impact the quality of the final velocity map. More neighbors imply more resolution in the velocity maps because there are more calculated travel times to use for interpolating the phase travel-time surface (Sections 2.3.1 and 2.3.2.).

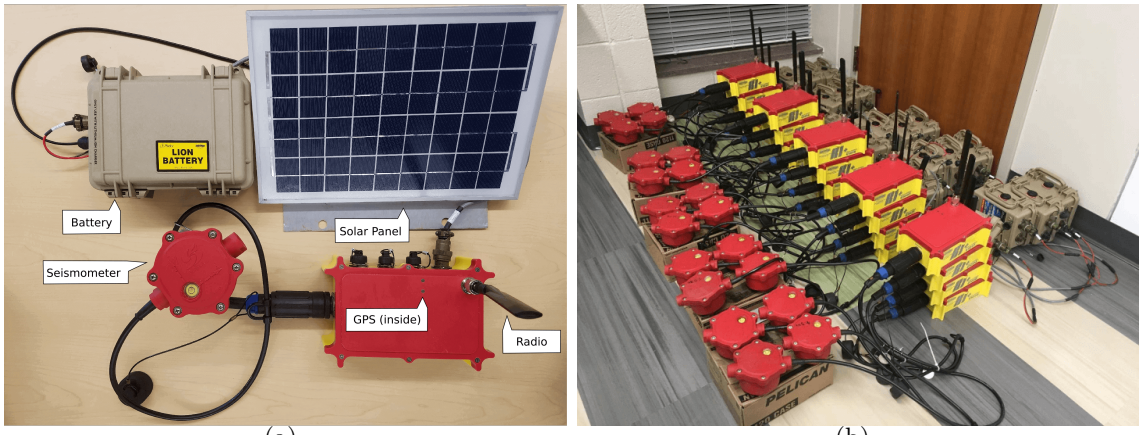
In traditional ambient noise tomography imaging, the cross-correlation of the signals is performed between all pairs of sensors, which requires $\mathcal{O}(N^2)$ computation. All-to-all cross-correlation represents a large volume of data and the use of nodes as multi-hop nodes in the network. Our system methodology aims to reduce computation and communication complexity by cross-correlating only with neighbor nodes in the mesh network (e.g. K neighboring sensors), which only requires $\mathcal{O}(KN)$ correlations. In strongly connected topologies, the velocity maps result will be almost the same as computing all-to-all cross-correlation. In sparse topologies, when we are only able to observe a subset of entries, the resolution of the recovered tomography may be reduced. However, when the underlying true map varies smoothly (which can be viewed as a low-dimensional structure), the quality of the recovered tomography using partial data will not degrade much from that recovered using the full data.

2.6 Deployment 3: University of Georgia campus

2.6.1 Deployment specifications

We use field smart seismic sensor in a outside deployment. Every sensor or field device has a global positioning system (GPS), three channel/component seismometer (geophone), a Raspberry Pi 3 board, a battery and a solar panel as shown in Fig. 2.20. Some hardware components are housed into a waterproof box called R1+ for protecting them from the harsh environment. The low-power GPS interface provides the geo-location of the sensor node and a time-stamp is used for the system to collect, synchronize and process the seismic data. The three channels geophone is incorporated into the system to detect the velocity of ground movements. Each channel records its own data respect to its axis N, E, and Z or directions North, East and Depth (vertical). The single board computer (Raspberry Pi) is the core of the system because is in charge of collecting and storing data, processing data analytics, communicating with other units and providing raw and processed information to a

visualization tool. We also integrate a waterproof battery 11V and 99.9 Wh. The battery is connected to a 10 Watt solar panel for giving to the system renewable energy.



(a) (b)

Figure 2.20: (a) R1+ hardware details. (b) R1+ seismograph nodes (for space reasons we omitted solar panels in this picture).

The detailed specifications of the main single-board computer inside R1+ are presented in Table 2.1

Table 2.1: Single-board computer specifications

Raspberry Pi 3 Model B	
CPU	1.2GHz 64-bit quad-core ARMv8
Memory	1 GB SDRAM
USB 2.0 ports	4 (via the on-board 5-port USB hub)
On-board storage	32 Gb Micro SDHC
On-board network	10/100 Mbit/s Ethernet, 802.11n wireless, Bluetooth 4.1

We deployed ten R1+ sensors on the University of Georgia (UGA) campus during January 24th, 2018. The deployment was located in an open area between three main buildings in which there are many pipes under the ground. The Google-Maps location of the devices is

shown in Fig. 2.21. The black box in Fig. 2.21 shows the ten sensors (nodes) over the field; white box illustrated the location of the nodes respecting each other. We recorded ambient noise data for 7 hours and performed cross-correlation, FTAN and Eikonal Tomography over these data using our system.

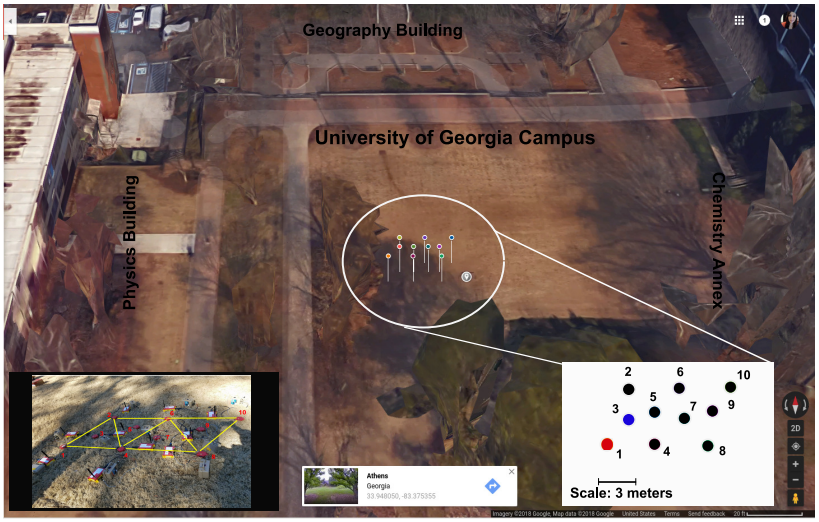


Figure 2.21: Deployment of ten R1+ over University of Georgia (UGA) Campus from Google Maps.

2.6.2 Results

From Fig. 2.22, we can observe the cross-correlation function results between between node 1 (red node on Fig. 2.21) and node 3 (blue node on Fig. 2.21). Notice that even though the correlation time was less than 1 day (7 hours), our system was able to obtain identifiable cross-correlation picks that allows FTAN to calculate the travel time between the specific two stations. The final velocity map obtained by the Eikonal method is shown in Fig. 2.23. Black diamonds represent the station locations plotted over the velocity map.

The main idea of this experiment is to test the system functionality and the ability to detect velocity variations using real devices. Because the inter-station distance is small (around 3 meters), we choose a high frequency to be analyzed. The sampling rate of our

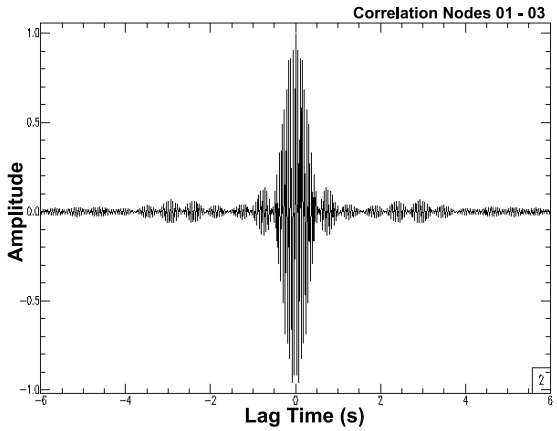


Figure 2.22: Cross-correlation result from Station 1 and Station 3 in the real deployment.

sensors is 500 Hz. Based on the Nyquist–Shannon sampling theorem, only the first 250 Hz are usable. Furthermore, to avoid aliasing effect[61], only frequencies up to 125 Hz can be adopted. From Fig. 2.23, in our application, the shallow subsurface velocity is around 1000 m/s. Considering a central frequency of 35 Hz, the wavelength Λ ($\Lambda = c/\omega$, where c is velocity and ω is frequency) will be about 28 m/s. Then, the seismic resolution is calculated by $\Lambda/4$, resulting in our vertical resolution being about 7 m. Because eikonal tomography is based on computing the spatial gradient of the travel time surface between sensors, the result of this experiment can be unstable due to the number of sensors used. However, the system functionality and the sensors' communication and computation show the possibility of computing ambient noise tomography in networks. If more sensors are added, results of eikonal tomography would be more stable.

2.6.3 Evaluation

In this section, two main aspects of our distributed approach are discussed: (i) the reliability of the eikonal tomography result compared with traditional tomography; and (ii) the trade-off between the centralized and distributed scenario.

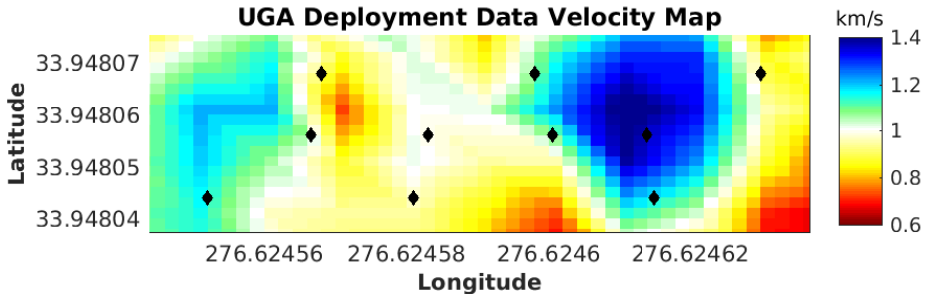
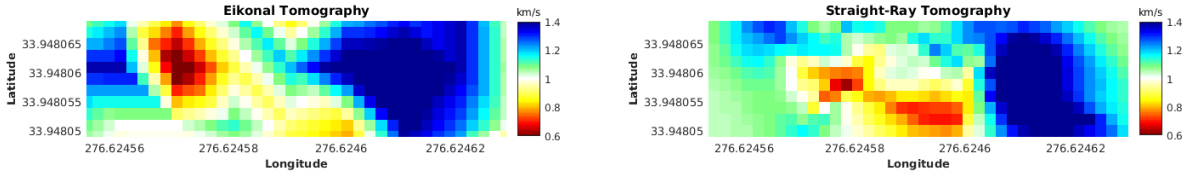


Figure 2.23: Velocity map (dominant frequency 35 Hz) obtained from the deployment area at University of Georgia (UGA).

The main purpose of our in-field experiment is to test the system ability to detect velocity variation using real devices. As mentioned, due to number of sensors limitations, the experiment was made with ten units only. The results can be unstable due to the short station-separation distance. To measure the resulting stability, we also performed a traditional tomographic method based on straight-ray approximation [41], which can be done only in a centralized fashion, and compared the results. This is a typical comparison to validate ambient noise tomography results [6]. Figs. 2.24(a) and (b) show the velocity maps obtained by eikonal tomography and straight-ray tomography respectively at a dominant frequency of 35Hz.



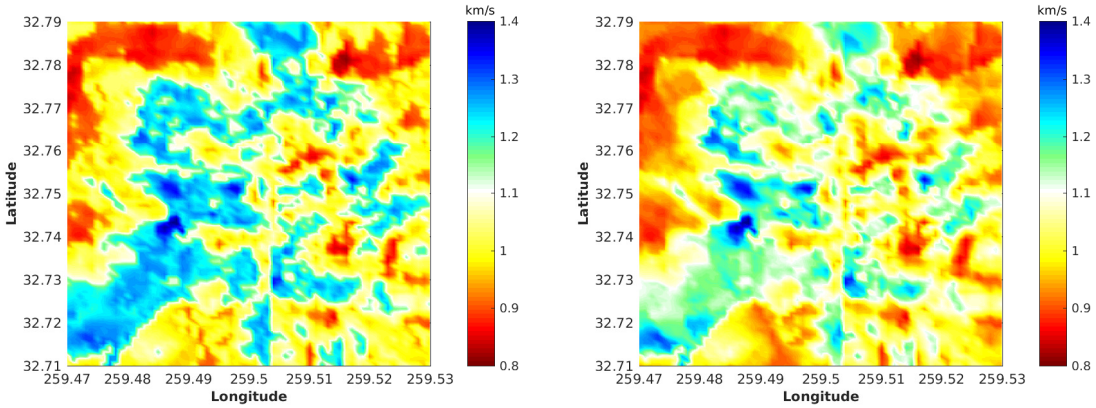
(a) (b)

Figure 2.24: (a) Eikonal Tomography from UGA deployment with dominant frequency of 35Hz (b) Traditional straight-ray tomography of the same deployment at 35Hz.

Agreement between the velocity maps produced with eikonal tomography and the traditional straight-ray tomography is generally favorable, but there are some regions with

significant disagreements. The main differences likely occur due to the regularization applied in the straight-ray inversion, which tends to distort the velocities near to the edge of the map. This was already reported in Lin et at. [6]. However, from these results, we can see our system is able to recover subsurface velocity differences in a distributed fashion. Other new distributed tomographic methods can be adapted into the system by changing the eikonal tomography module for a new tomographic technique; an adaptation of the input tomographic parameters may be needed too.

Finally, a discussion of the trade-off between the centralized and the distributed approach is presented here. For the image phase, we have made an extensive comparison with the centralized approach in Valero et al. [1]. Notice that because we have used real datasets, there is no ground truth for the velocity of Sweetwater Data and/or UGA deployment. Hence, we focus on the comparison of the proposed method with the centralized processing scheme, which can be used as a benchmark that fully utilizes the data available. Interpretation of this data requires in-depth knowledge of geophysics and is out of the scope of this paper.



(a) (b)

Figure 2.25: Velocity map of Sweetwater Database. (a) Centralized approach. (b) Distributed approach.

Figs. 2.25(a) and (b) show the final velocity map of the Sweetwater area analyzed in this paper using a centralized approach and distributed approach respectively. There are

some disagreement between both maps mainly because the centralized method utilizes more information on cross-correlations. However, the mean squared distance (e_1) and average value distance (e_2) between both approaches are less than 15%, and we can notice a similar pattern in both maps and a differentiation in the structures. This indicates that the distributed method is able to recover similar results than the centralized approach. The main advantage of the distributed method is the communication cost is significantly reduced, and the bandwidth and network constraints are met. The disadvantage relies on the fact of total failure of nodes. The system is resilient to package lost; however, if a significant portion of the nodes in the network dies, the result will be considerably affected. On the other hand, the centralized approach guarantees more accurate results, but the cost of transferring all data to a central place is very high in terms of sensor networks.

2.7 Conclusion

In this chapter, we presented an innovative distributed cooperative in-network system for real-time seismic imaging generation through ambient noise data. We integrated in-network processing techniques to correlate the noise signals between sensors and derive the phase velocity under the limited network resource constraints. We showed that computing information at the node level and cooperating with neighbors makes it possible to illuminate near-surface velocities of the earth. We showed that both system phases produce results close to the centralized approach, and they balance communication across the network. Furthermore, we also tested our algorithms under field conditions of sensor networks, such as loss of packages, and showed they are robust in terms of loss tolerance.

2.8 Publications

The work presented in this Chapter has led to the following publications

- Valero, M., Li, F., & Song, W. (2019). **Smart Seismic Network for Shallow Subsurface Imaging and Infrastructure Security.** Submitted to: International Journal of Sensor Networks (IJSNet). 31(1):10–23. DOI: 10.1504/IJSNET.2019.10022957

- Valero, M., Li, F., Wang, S., Lin, F. C., & Song, W. (2018). **Real-time cooperative analytics for ambient noise tomography in sensor networks.** IEEE Transactions on Signal and Information Processing over Networks. 5(2):375–389. DOI: 10.1109/TSIPN.2018.2876751

- Valero, M., Clemente, J., Kamath, G., Xie, Y., Lin, F. C., & Song, W. (2017, May). **Real-time ambient noise subsurface imaging in distributed sensor networks.** In 2017 IEEE International Conference on Smart Computing (SMARTCOMP) (pp. 1-8). IEEE. DOI: 10.1109/SMARTCOMP.2017.7947040

Chapter 3

Distributed Spatial Autocorrelation in Sensor Networks

3.1 Motivation

In Chapter 2, a distributed ambient noise seismic imaging technique was presented. This method aims to reduce computation and communication complexity by cross-correlating only with neighbor nodes in the mesh network (e.g. K neighboring sensors), which only requires $\mathcal{O}(KN)$ correlations. This guaranties approximated results than traditional methods. However, in traditional ambient noise tomography imaging, the cross-correlation of the signals must be performed between all pairs of sensors, which requires $\mathcal{O}(N^2)$ computation. For larger arrays of sensors, this imposition represents a diminishing of the throughput of the network. It is important to remember the performance limit of the network throughput is intrinsically related to the maximum amount of traffic per unit time [62] (usually measured in bit/sec). All-to-all cross-correlation represents a large volume of data and the use of nodes as multi-hop nodes in the network.

To overcome this problem, we introduce dSPAC (distributed Spatial Autocorrelation)

technique for computing velocity maps in a local area only with neighbors and use an interpolation technique for larger areas. The spatial autocorrelation techniques does not require all to all cross-correlations and because high frequencies can be used, shallow subsurface infrastructures can be imaged.

The use of larger arrays of sensors, separated by long distances, allows the use of low frequencies because low frequency travels further than high frequency on earth. Then, having sensors separated by larger distances can allow the study of the wavelengths in larger periods. Instead of using inversion, we use an approximation of the depth velocities using sensitivity kernels. According to depth sensitivity kernel theory [63], low frequencies allows to illuminate much depth earth structures. However, this implies the use of many multi-hop nodes in order to retransmit the signal between one station to another. If the inter-node spacing is shorter, we can only study high frequencies and in consequence more shallow structures. If traditional methods uses high frequencies instead low frequencies, the results may be approximately equal to ANSI results.

Because we are using wireless sensor networks, we are limited by the communication range between sensors. Then, we propose a system for collaborative subsurface infrastructure detection using a wireless seismic network. We use ambient noise tomography in which temporal variation of the earth structure can be studied and monitored by studying the variation in the noise cross-correlation function [36]. We use spatial auto-correlation (SPAC) [64] as the base of our ambient noise imaging.

Spatial auto-correlation is a method for subsurface exploration that is popularly used to infer phase velocities of the surface waves. Phase velocities can provide information about the subsurface structures since some structure's materials may produce different velocities at different frequencies. Even though the method is well-established and widely applied in subsurface imaging, the majority of the approaches use post-processing in a central server after gathering the data from the field. Our system is able to do in-situ signal processing and

collaborative imaging computing in the field by leveraging the potential of current sensor technology.

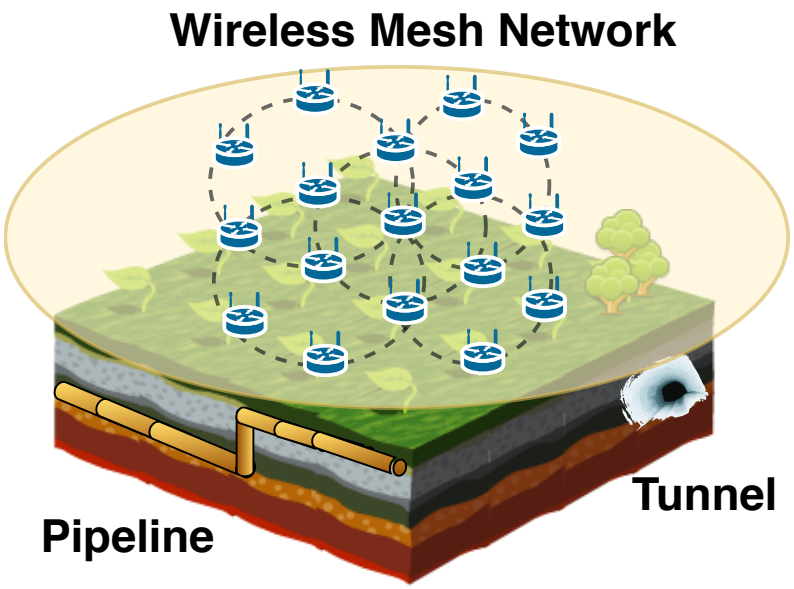


Figure 3.1: Example of mesh network for subsurface imaging. The used geometry allows many rings in the network, each one with a center station.

The proposed system is called dSPAC (distributed Spatial Autocorrelation), and its main innovations are:

- in-network processing techniques to correlate the noise signals between nodes and derive the phase velocity under the limited network resource constraints, using a mesh network like shown in Fig. 3.1;
- in-network tomography computing techniques that distribute the tomographic computing burdens to each node while performing real-time seismic image computation;
- seismic sensor geometry and techniques (SPAC) suitable for areas where is needed to study high frequency ranges for detecting shallow subsurface structures;

- autonomous system that can be deployed in harsh environment and can deliver results with minimal user intervention and almost zero-maintenance;
- self-healing system based on a mesh network with fault-tolerance.

3.2 Methodology

The ability to detect subsurface or buried infrastructure with a passive, non-invasive method that does not require any excitation of the medium represents a milestone for many engineering and security applications. Also, if the solution can be done in-situ, in real time, and with minimal user intervention, the contribution would be highly beneficial. To achieve this goal, in this paper a mesh network of sensors have been designed and integrated with signal processing and geophysics techniques. Every sensor in the network is responsible to sense, prepare, communicate, and process the ambient noise signal to calculate a subsurface image based on underground velocities of seismic waves.

In this section, we described the main principles of our algorithm and system design and the fundamental theory behind our approach. Every node starts with sensing and signal preparation; then, based in cross-correlation and the spatial autocorrelation of the signal, sink nodes would be able to detect subsurface velocity variations and generate a "picture" of underground infrastructures.

3.2.1 Signal Preparation

The ambient noise raw data sensed from each individual sensor need to be prepared to get a suitable individual waveform for future cross-correlation. As explained in [43], the purpose of this preparation is to accentuate ambient noise by attempting to remove earthquake signals and instrumental irregularities that tend to hide the ambient noise. The signal preparation

has three important steps: (i) removing instrumental error response and cutting data; (ii) time-domain normalization and (iii) spectral whitening.

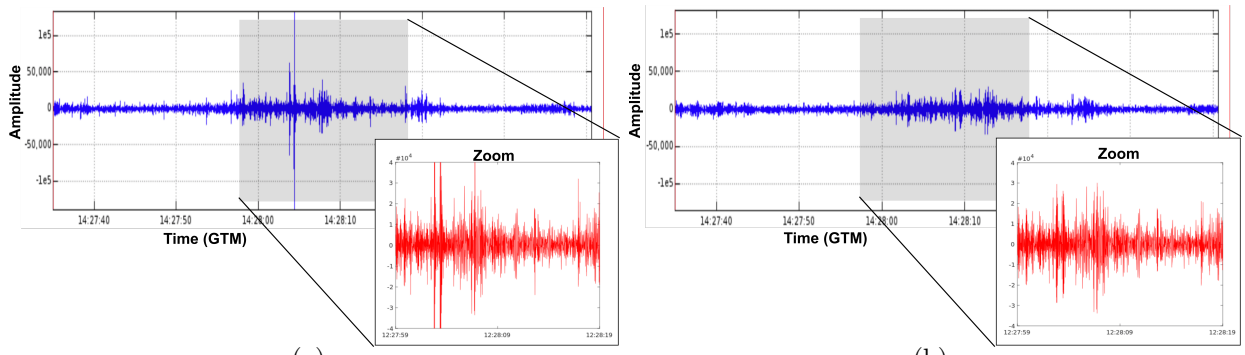
To remove instrumental irregularities, the first step is to remove the mean and the trend of the signal. Then a taper is applied to improve signal properties in the frequency domain [45]. A simple cosine taper filter that applies cosine-shaped attenuation function to specified frequencies at low and high frequencies is applied to remove instrument irregularities. Additionally, the data should be cut into a specific time-window to be analyzed in a window fashion. Data can be cut on one day, some hours, a few minutes. This window of time t will be used for posterior steps (cross-correlation) and stacked together until complete the total time T of the signal.

The next step is time-domain normalization, also called temporal normalization [43]. The time-domain normalization we use is running-absolute-mean normalization [43]. This method computes the running average of the absolute of the waveform in a normalization time window of fixed length and weight the waveform at the centre of the window by the inverse of this average. Given a discrete time-series f, the normalization weight is

$$w_n = \frac{1}{2N+1} \sum_{i=n-N}^{n+N} |f_i|, \tag{3.1}$$

and the normalized datum is $\tilde{f}_n = f_n / w_n$. The width of the normalization window is $2N + 1$.

Finally, a spectral normalization is applied. Spectral normalization seeks to reduce broad imbalances in single-station spectra to aid in the production of a broad-band dispersion measurement [43]. Fig. 3.2 shows an example of ambient noise preservation after data preparation.



(a) (b)

Figure 3.2: Signal preparation example. (a) Raw seismic data sensed by a sensor node. High picks represent possible events that obscure the ambient noise. (b) Data after preparation preserving ambient noise.

3.2.2 Cross-correlation

Cross-correlation of ambient noise can be used to determine the impulse response between two sensors considered receivers. Cross-correlation in time-domain is used to find similarities between two different time-series.

For two signal (here we assuming the signal has been prepared) from two different locations $f(t) = u(x_1, t)$ and $g(t) = u(x_2, t)$ getting by two sensor nodes, where $u(x, t)$ is the prepared seismogram at spatial location x, cross-correlation can be written as:

$$C_{x_1 x_2}(t) \equiv \frac{1}{2T} \int_{-T}^{T} u(x_1, \tau) u(x_2, \tau + t) \mathrm{d}\tau \tag{3.2}$$

In this way, the similarities of the signals can be determined. Fig. 3.3(a) shows an example of symmetric cross-correlation between two stations separated by 3 meters distance. If a bandpass filter is applied before cross-correlation, we can notice the dominant frequency in the spectrum band area. For example, Fig. 3.3(b) shows the spectrogram of the cross-correlation with a bandpass between 17-20Hz.

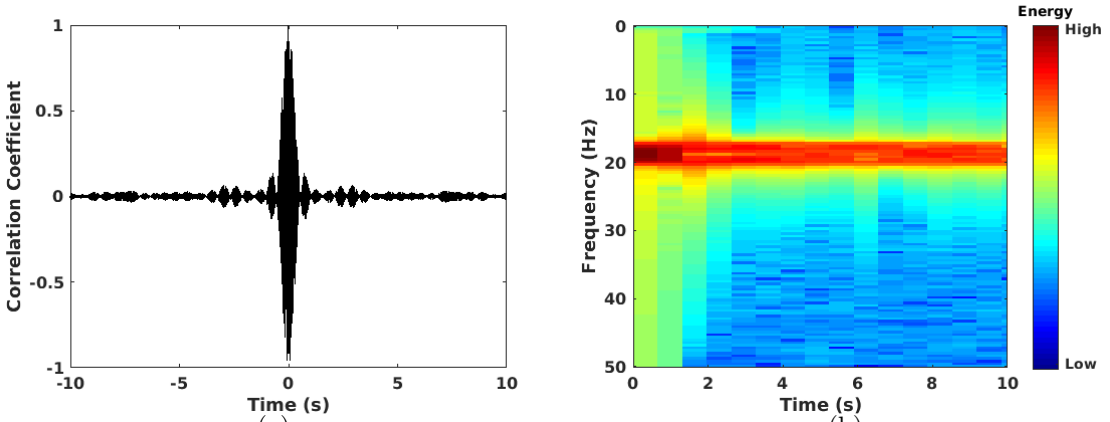


Figure 3.3: Cross-correlation example between two sensor's data. (a) Symmetric cross-correlation. (b) Frequency-time spectrogram where is possible to distinguish dominant frequencies.

3.2.3 Spatial Autocorrelation

SPAC method was first introduced by Akin [64] based on a statistical investigation of seismic waves. Modifications of the method have been presented overtime [65–67]. We chose SPAC because is a method that can be applied when the stations (sensors) separation is large or short. We integrated the cross-correlation coefficients into the SPAC solution, and we aggregated the method's results in a distributed fashion, where the collection of all raw data in a central server and then post-process them is no longer needed. Even though in the SPAC method some nodes will act as sink nodes, they only receive either processed/compressed data or partial velocity results; never raw data.

The SPAC method can extract the phase velocities of surface waves from microtremor array observations. The basic theory of the SPAC method [66] is summarized as follows. Having an array of sensors (called receivers) equally spaced on a circle of radius r and having an extra receiver at the center as shown in Fig. 3.4, the phase velocities can be calculated.

If microtremors are observed, the complex coherencies COH between a central and a

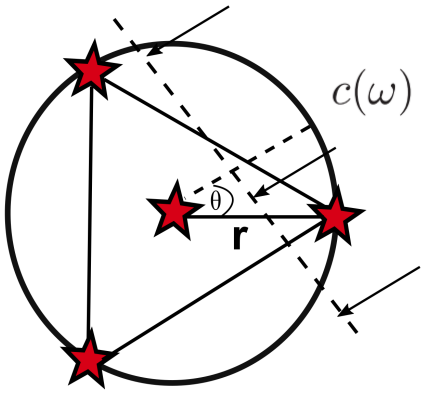


Figure 3.4: Geometry of sensor nodes and an incident plane wave. Red starts represent sensors.

circumferential receiver can be defined as:

$$\mathrm{COH}(r, \omega, \theta, \phi) = \exp\{irk\cos(\omega - \phi)\}, \tag{3.3}$$

where i is the imaginary number, ω is the angular frequency, k is the wavenumber, θ is the azimuthal angle and ϕ is the azimuth propagation of a single plane wave across the array. The SPAC coefficients, also called azimuthal average, is defined then by:

$$\rho(r, \omega) = \frac{1}{2\pi}\int_0^{2\pi} \exp\{irkcos(\theta - \phi)\}\mathrm{d}\theta = J_0\left[\frac{\omega}{c(\omega)}r\right], \tag{3.4}$$

where J_0 is the Bessel function of the first kind of zero order, and $c(\omega)$ denotes the phase velocity. Here r must be fixed. Because of the $cos(\theta-\phi)$ symmetry in Eq. 3.4, we can switch ω with ϕ and obtain the same result. This means that SPAC coefficient can be estimated as the average of the cross-correlation (cross-coherence) between every node pair in a fixed geometry with the same ratio $\mathbf{r}$, which remedies the biases in phase velocity measurements caused by a non-isotropic or directional wavefield. In other words, $\frac{1}{2\pi}\int_0^{2\pi} \exp\{irkcos(\theta - \phi)\}\mathrm{d}\theta$ is

equivalent to $\frac{1}{N}\sum(C_{x_i x_j}(t))\forall j \in N$, where N is the number of sensors in the circular array and i is the central sensor (Detail explanation of this equivalence can be found in Appendix). The phase velocities are estimated by fitting the observed SPAC coefficients to the Bessel function. This means, the coefficient of SPAC is related to the seismic phase velocity through the Bessel function of the first kind of order zero.

$$\rho(r,\omega) = J_0\left[\frac{2\pi\omega r}{c(\omega)}\right], \tag{3.5}$$

where $c(\omega)$ is the velocity at frequency ω. The curve of SPAC coefficients is fitted to the Bessel equation in order to obtain the argument for the Bessel function (x) which is correlated with the value of $2\pi\omega r$. Therefore for each argument of Bessel function x_i we can find phase velocity at frequency ω_i,

$$c(\omega_i) = 2\pi\omega r/x_i \tag{3.6}$$

Fig. 3.5 shows an example of calculated SPAC coefficients and the corresponding velocity estimation. We use the API of Matlab and the function fit() to fit the SPAC coefficients with the Bessel function. Then we estimate the velocity.

Equivalence between Cross-correlation and SPAC

We extract the equivalence theory between cross-correlation and SPAC from [68] , we summarize the key steps here:

It was observed by Jacobson [69] that for isotropic noise of a single frequency w,

$$C_{x_1x_2}(t,\omega) \equiv C_{x_1x_2}(t) = J_0(\omega/c)\cdot\cos(\omega). \tag{3.7}$$

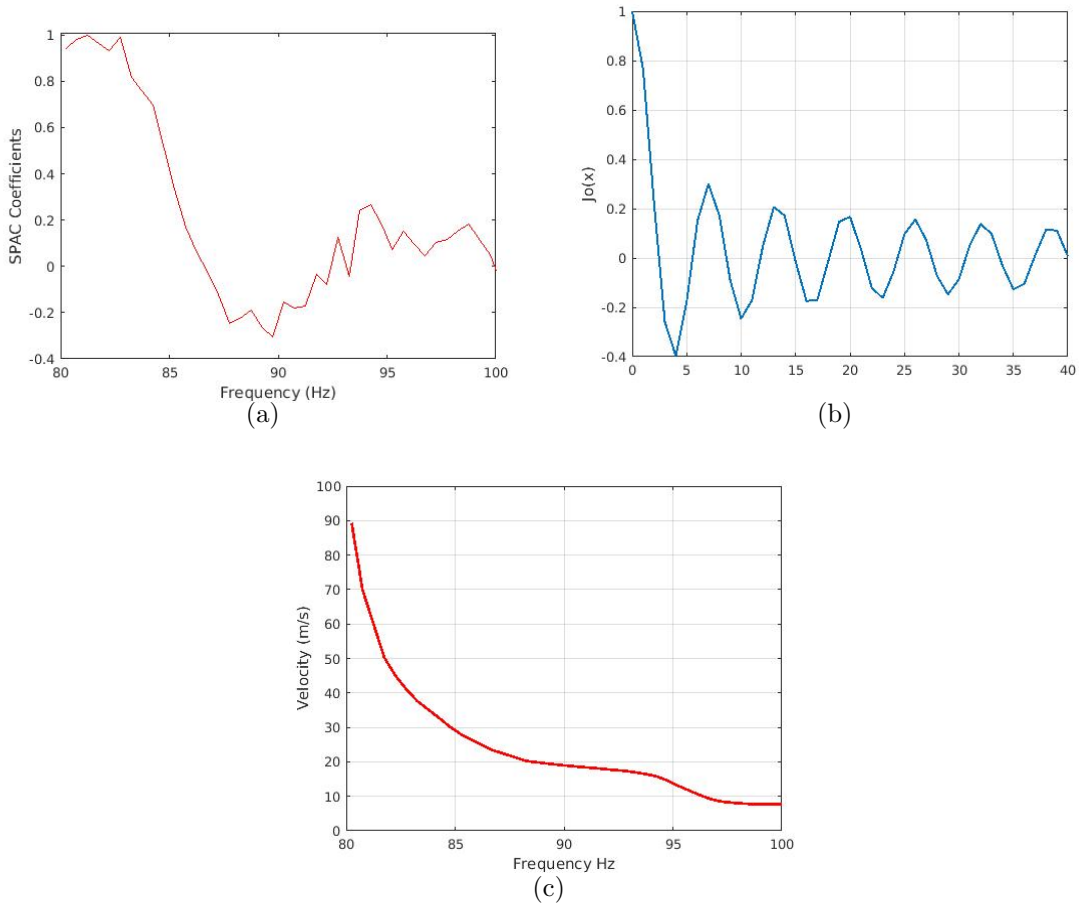


Figure 3.5: (a) SPAC coefficients. (b) Bessel function of order zero. (c) Velocities.

On the other hand, In Tsai et al. [68] it was shown that the azimuth average for SPAC is

$$\rho(r,\omega) = \frac{\bar{\Phi}(r,\omega)}{\Phi(0,\omega)} = \frac{2\pi\Phi(w)J_0(\omega r/c)}{2\pi\Phi(\omega)\cdot 1} = J_0(\omega r/c), \tag{3.8}$$

where $\bar{\Phi}$ is the azimuthal average. The same azimuthal averaging technique performed for getting $\bar{\Phi}$ can be applied to $C_{x_1x_2}$ to show that

$$\bar{C}_{x_1x_2}(t,\omega) = 2\pi\Phi(\omega)J_0(\omega r/c)\cos(\omega t), \tag{3.9}$$

where $\bar{C}_{x_1x_2}$ is also the azimuthal average. Thus, if cross-correlation measurements in a certain region are made for a range of different azimuth, as the circular array in the SPAC, the average of the cross-correlation can be used as SPAC coefficients.

3.3 System Implementation

We implemented our method on a wireless network system to obtain subsurface images of underground infrastructures. Using a mesh network of seismic but computational powerful sensors, we use wireless communication and in-situ computation to generate almost real-time subsurface velocity images. Due to SPAC constraints regarding the distance between sensor nodes, the mesh configuration must be a circular array such as the one in Fig. 3.1.

The diagram flow of the complete methodology is shown in Fig. 3.6. Let t be the window-time of reading from the medium size. In our system, every node reads continuously, and every t time, it starts the *in-situ signal processing process*. The parameter t is configurable in the system. Every node individually performs a down-sampling to the t times of data and executes the data preparation described in section 3.2.1. After data preparation, a compression technique is applied to make the signal suitable to be transmitted in the network to improve the communication cost and meet bandwidth limitations. We use *zlib* data compression algorithm [51] and we achieve a compression rate of $\sim 50\%$.

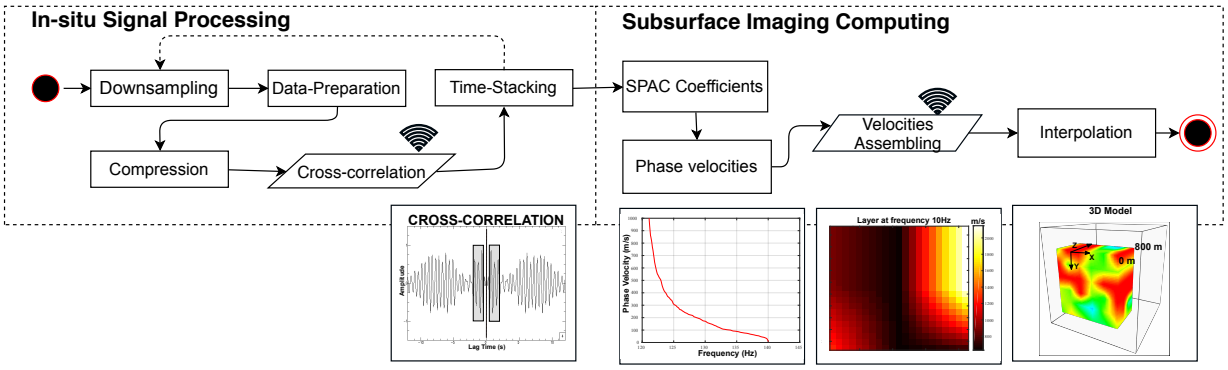


Figure 3.6: System methodology for SPAC-based subsurface imaging.

To perform the cross-correlation between a pair of signals, we transfer the window-prepared signal from each sensor in the ring to the sink sensor of that ring. Notice that this process is performed in parallel for all rings and sink nodes in the network. We use User Datagram Protocol (UDP) for broadcasting the data. When every sink sensor receives the data, it performs the cross-correlation and stacks the result of the window t with the previously stacked cross-correlations. Stacking over increasingly long time-series, on average, improves the signal-to-noise ratio (SNR).

The aforementioned procedure is performed during a time T (configured in the system) in which many windows of time t are correlated and stacked. For instance, we can perform the cross-correlation in windows of size $t = 5$ minutes, and continue doing that for $T = 10$ hours. After T, the system starts the *subsurface imaging computation*. Parameters t and T are setup in-advance in the system by the experts. However, we need to mention that having more time produce better imaging. Longer T values ensure a continuous imaging that is improving over the time as the SNR is better every time that the cross-correlation is stacked.

The subsurface imaging is performed by all sink sensors at each ring. The SPAC coefficients are then calculated by using the average of the cross-correlations as defined in Eq. 3.4. Then, the phase velocities are estimated by fitting the observed SPAC coefficients to the Bessel function. Here, the sink sensors are able to calculate a velocity for each frequency (ω). The sink sensors at each ring broadcast the velocity information to the other sink nodes, and they perform an interpolation process to form a 3D map of the subsurface with all the frequencies in consideration. Each layer of the 3D map represents a subsurface depth. With this information, we can analyze the velocity variations and determine the presence of subsurface structures within the subsurface.

The detail of the distributed algorithm from the perspective of a sink sensor is presented in Algorithm 5. The other nodes in the ring just gather the data, prepare the data for transmission, compress and broadcast the data to sink sensors.

Algorithm 5 Distributed SPAC-based Subsurface Imaging

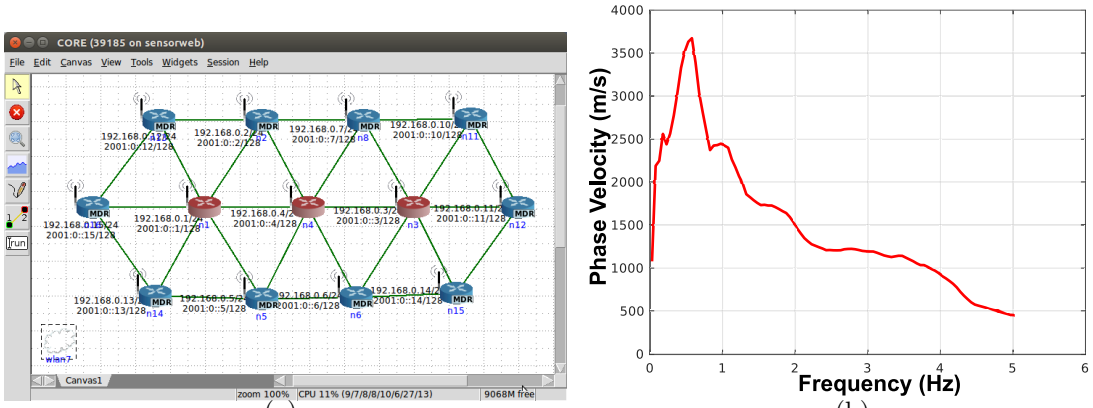
Input: Size of correlation-window t
Input: Total size of cross-correlation T
Input: Station location x_i, ratio r
While not reach time T
  Read prepared $u(x_i, \tau)$ at position x_i
  for every t
    Receive $u(x_j, \tau)$ for every j in i ring
    Do $C_{x_i x_j} = \frac{1}{2T} \int_{-t}^{t} u(x_i, \tau) u(x_j, \tau + t) \mathrm{d}\tau = \mathcal{F}(x_i) \cdot \overline{\mathcal{F}(x_j)}$ for each j
    Stack cross-correlation $\bar{C}_{x_i x_j}$ for each j
  end for
End While
Calculate SPAC coefficient $\rho(r, \omega)$ for each ω
Estimate velocities by fitting $\rho(r, \omega) = J_0[\frac{\omega}{c_i(\omega)} r]$
Broadcast $\hat{c}_i$ vector that contains all $c_i(\omega)$
Receive $\hat{c}_k$ from other k sink sensors
Interpolate $\hat{c}_i$ and $\hat{c}_k$ to get each depth layer and generate 3D velocity structure M
Output: 3D velocity structure (M)

3.4 Deployment 1: Pipeline Detection

3.4.1 Initial validation

For testing our algorithm, we performed a first experiment using CORE network emulator [60]. We use the data from Sweetwater, Texas [59] on each node in a suitable geometry area (Fig. 3.7(a)) and we calculated the phase velocities of the area with our proposed method as shown in Fig. 3.7(b). More specifically, we took 13 nodes that were in in a cuasi-circular area and with relative same distance between them just for comparison purposes. We compared the results with the knowing results gotten by the Eikonal Tomography and ANSI methodology we presented in Chapter 2. We wanted to evaluate if our method can generate similar results.

Notice that, for frequencies of 2Hz and 4Hz, the velocity is around 1500m/s and 1000m/s respectively. As mentioned, we also analyze Sweetwater dataset using eikonal tomography technique. For the same frequencies, we got similar results: 1550m/s for 2Hz and 985m/s



(a) (b)

Figure 3.7: (a) Core emulator performing SPAC-based subsurface imaging. Red nodes represent the sink sensors. (b) Phase velocity of frequencies between 0Hz and 5Hz calculated in one sink sensor.

for 4Hz. With these results, we experimentally can say our method can be equivalent and complementary to the eikonal tomography method.

3.4.2 Deployment Specifications

In our deployments, we use a mesh network composed by smart sensor that were shown in Fig. 2.20(a) and explained in Section 2.6. Expected hardware improvements are beneficial for our detection purposes, and our system can be easily adapted to more sophisticated instruments.

The instruments are placed in the field in a ring-based topology as shown in Fig. 3.8. For illustration purposes, we only show the geophone locations (without other instruments) and two rings illustration. The instruments must be placed over the structure to locate, for example, a pipeline or tunnel.

Once the battery is connected to the sensors, the system automatically calibrates itself and finds GPS signal for synchronization. The system parameters are read from configuration

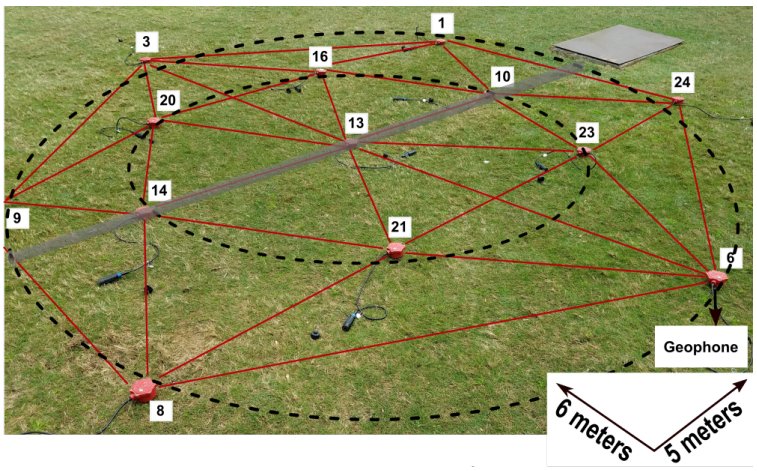


Figure 3.8: Example of instrument deployed in the field. Pipeline structure is under the deployment location. The thirteen sensor nodes form two main ring with radios r_1 and r_2. Notice other sub-rings can be formed inside the main rings.

files and the sensor reading from the medium starts. Human intervention is minimal, but the system is completely monitorable using a laptop connected to the mesh network.

For detection testing, we wanted to validate our results against a ground truth. We deployed thirteen seismic sensors that run our system in an open area inside the University of Georgia (UGA) Campus. The selected place is known for having underground pipelines. Our deployment geometry and location are shown in Fig. 3.8. The system parameters we used for this experiment are shown in Table 3.1.

Parameter	**Used Value**
Number of sensors	13
Number of sink sensor	7
Ratio (r)	Internal circle: 1.7 meters External circle: 3 meters
Frequency range (ω)	20Hz to 110Hz
Cross-correlation window (t)	5 minutes
Total time system running (T)	4 hours

Table 3.1: Deployment parameters used in UGA experiment

3.4.3 Results

The cross-correlation results were obtained and stacked every t minutes. After T hours of continuous system execution, the nodes cooperatively construct the velocity map for each depth layer. Then, the sink sensor used depth sensitivity kernel theory [63] to invert the frequency ranges in depth. Fig 3.9 shows four layers at different depths. The area with high velocity in the map indicates that it should be an isolated structure/facility, corresponding to the wanted pipeline we are aiming to detect under our deployment. Notice that shallow depths have better resolution. Between 1 to 1.5 meters, it is possible to distinguish a change in potential pipeline velocity. Depth layers, for example, Fig. 3.9(a), show changes in velocity but the resolution is low.

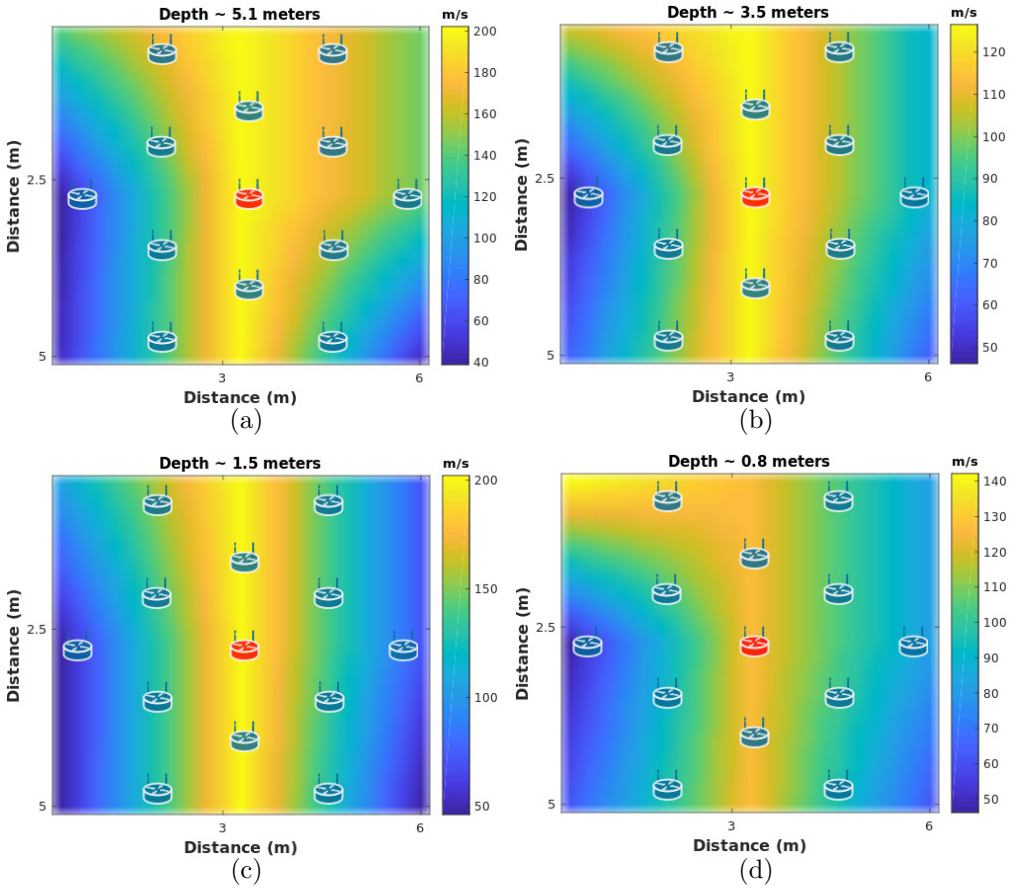


Figure 3.9: Velocity Maps. (a) Layer $\sim$ 5.1 meters depth. (b) Layer $\sim$ 3.5 meters depth. (c) Layer $\sim$ 1.5 meters depth. (d) Layer $\sim$ 0.8 meters depth. Sensor nodes locations are plotted as reference.

The sink sensor constructs a 3D subsurface velocity image, as shown in Fig. 3.10, by interpolating the velocity profiles from all the nodes. In this figure, only depths between 1 and 1.7 meters are shown. In the center of the velocity map, we can notice the high-velocity area is corresponding to the pipe location. Due to the high propagation velocity of the metal pipe, the surrounding soils also show high velocities than other areas. Horizontal resolution can be adjusted to a narrow frequency band which has the most significant responses with the pipe to obtain a better resolution. In addition, the vertical resolution can be further improved, if there are more stations. This result shows we are able to see structures under the subsurface and potentially extending our work for some security issues (for example, detecting broken pipelines, detecting tunnels, etc).

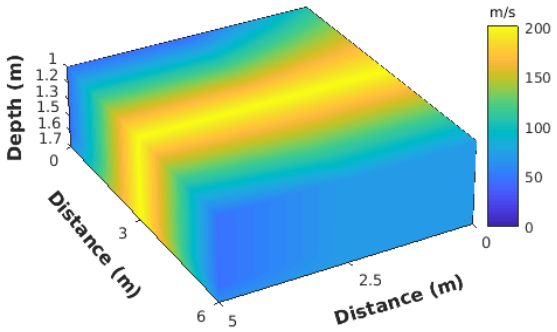


Figure 3.10: 3D velocity subsurface. Layers between 1 to 1.7 meters.

3.4.4 Evaluation

In this section, we evaluate the system in terms of bandwidth, communication cost, and resilience. The improvement of these three features makes our proposed system a reliable and attractive method for subsurface infrastructure imaging. For our evaluations, we compared our decentralized SPAC-based imaging process with a centralized approach. In the centralized approach, there is a central node, and all the other nodes send the raw data to this central node all the time. There is no data pre-processing on each node, all the computation is done

in the central node.

Bandwidth Analysis

In a mesh network, every "*hop*" (link) between sensors will decrease the bandwidth by half [70]. This happens because wireless links can only do one thing at a time - transmit or receive. In a long "chain" of mesh links, this results in a very slow connection from end to end. Even though this estimation (half of the bandwidth decreasing by every link) is widely accepted, in reality, other factors can impact the available bandwidth in a specific time; for example, communication range, other networks interference, etc.

We calculate the available bandwidth based on our hardware limitations and the throughput of the network at each time point. Then, we compare the distributed approach proposed in this paper, with the centralized approach.

Our instruments are based on a Raspberry Pi 3 as computer board. The wireless communication bandwidth of Raspberry Pi 3 is estimated at ∼10Mbps (Megabytes per second) [71]. Due to the number of links in our topology (some nodes may have 5 or 6 links, which reduced the available bandwidth), we based our observations on a maximum available bandwidth of ∼2Mbps.

Fig. 3.11 shows the comparison between the distributed and centralized approaches. This throughput was recorded for 120 seconds in which nodes in the distributed approach exchange information with the neighbors every $t = 20s$ to perform cross-correlation later. In the centralized approach, the nodes are all the time sending raw data to the central place, and we can notice that the average available bandwidth is very low all the time. On the other hand, with our distributed approach, the available bandwidth only has a small decrease during transmission for cross-correlation. Our approach meets the bandwidth limitations, and the sent packages are small due to data preparation and compression.

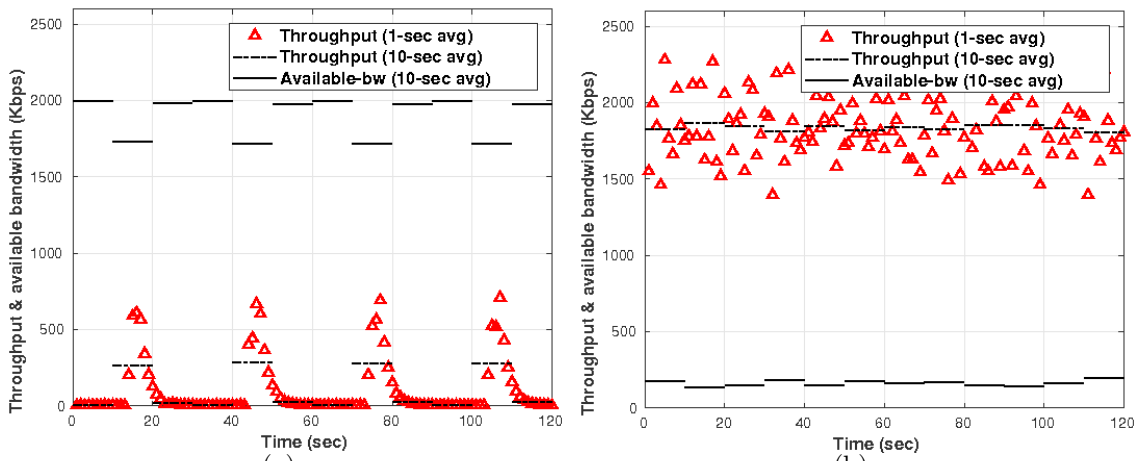
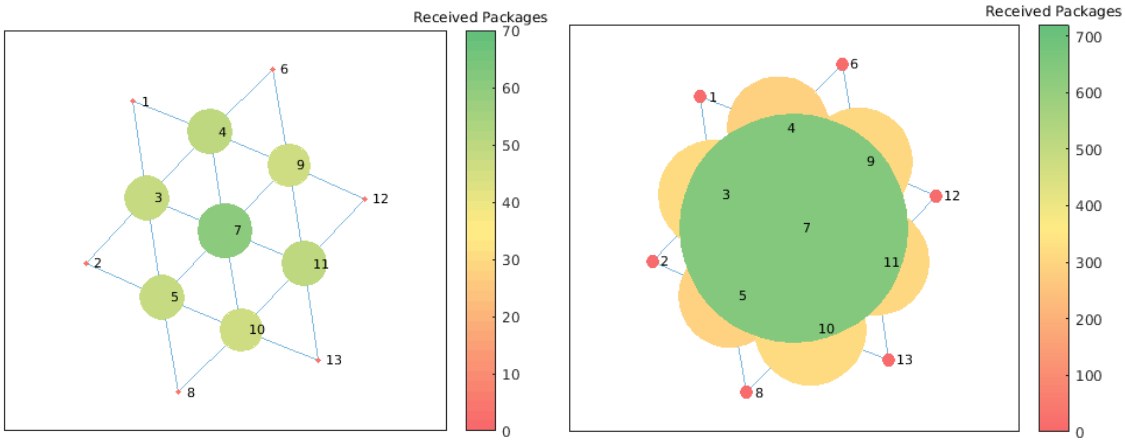


Figure 3.11: Throughput and bandwidth availability in (a) distributed approach, (b) centralized approach.

Communication Cost Analysis

An analysis of the system performance based on communication cost was also analyzed for the proposed approach. Because the most intensive communication scenario occurs when the data is continuously transmitted for cross-correlation, we present the communication cost after 1 hour of transmission.

From Fig. 3.12(a) and (b), we can see that communication cost in a centralized setup is high near the "sink node" as all the raw data are transferred over the network. It is notorious that the distributed approach improves significantly the communication cost between nodes. The reduction in the number of received messages is ∼75%. This also has an impact in the energy consumption of each node. According to Pottie et al. [29], the energy of transmitting 1KB a distance of 100m is approximately the same that executing 3 million of instruction by processor. Hence, local data processing is crucial for also saving sensors energy. This implies that our approach besides reducing communication cost, it also helps to avoid extra energy utilization.



(a) (b)

Figure 3.12: Communication cost in terms of number of received messages by each node. Communication of data for cross-correlation after 1 hour of execution. (a) Distributed approach - Number of messages between 0 and 70. (b) Centralized approach - Number of messages between 0 and 700.

Resilience Analysis

An important feature we want to mention is the behavior of the system in the scenario of sensor failures. Suppose that during the time T, one or more nodes fail. The system has been designed to restart automatically the operations after failure. However, during the time the node is down, the other nodes continue working broadcasting data for cross-correlation. At the moment the node is automatically restarted, it synchronizes itself via GPS with the rest of the nodes, and it continues the cross-correlation of the data from that point. Because, after cross-correlation, the system stack the results (*time-stacking*), the short-time failure does not affect the reliability of the cross-correlations. This guarantees self-healing and resilience of the system.

However, for the "subsurface imaging" process, the loss of one of the sink sensors is crucial for the velocity assembling and interpolation. For this reason, we have designed a recovery scheme for recalculating the velocity map after a sink sensor failure. The scheme is described

in Fig. 3.13 from a sink sensor perspective.

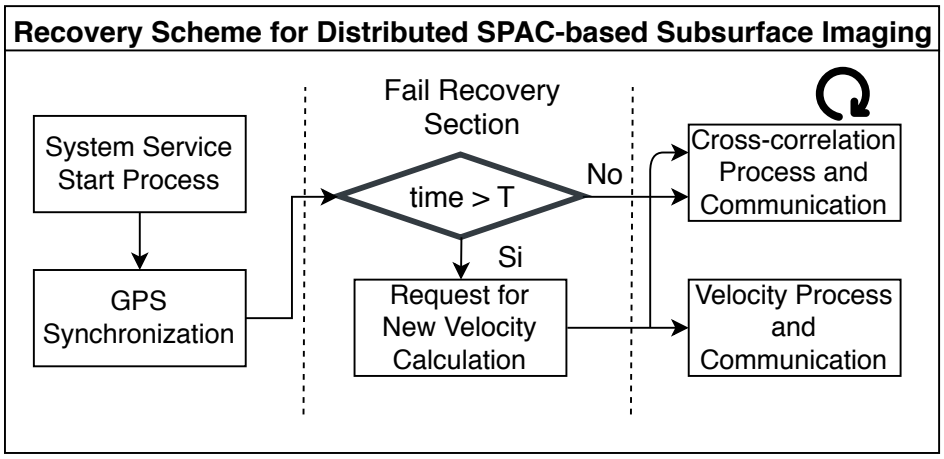


Figure 3.13: Recovery scheme for system resilience after failures from the sink nodes perspective.

In this scheme, after a sensor is automatically started with a system service, and it has been synchronized with the other sensors, the sensor checks if there a velocity calculation and imaging has been done during the time it was down. This is done by checking is the current time is greater than the time the process supposes to be performed. If this happens, the sink node sends a request to other sink nodes for recalculation of the velocities and interpolation. The cross-correlation process also starts in any case. With this scheme, we introduce resilience to the system, and we aim to guarantee that the results will be computed with the maximum numbers of available sink sensors.

3.5 Deployment 2: Water leakage detection

Water loss is mainly attributed to leakages within the water distribution systems [72]. Obviously, detecting and locating the water leaks can help save significant water and avoid the possible damages from water leaks to infrastructure. Commonly, the difference between dry soils and water-saturated soils around the pipeline can be used to detect leakage, based on electrical resistivity tomography [73, 74], ground penetrating radar [72], electromagnetic

profile [75] and etc. Similarly, near surface seismic imaging helps monitor shallow buried objects [76–78], for example, very shallow seismic reflection and refraction experiments were conducted to investigate groundwater level changes in beach sand in situ [79].

Compared with the intrusive investigation using borehole survey, passive surface wave methods have gained increasing popularity recently due to their non-intrusive nature [78]. Besides, the increasing number of dense arrays have been used for imaging the shallow buried objects recently [80]. The same methodology applied for dSPAC based on the spatial autocorrelation (SPAC) method proposed by Aki [64] and restated in Tsai et al. [68] is adopted because of its ability to generate high-resolution images. Moreover, another ambient noise based method-seismic interferometry has also been used to obtain 1D velocity model in Japan [81] and the depth of bedrock in Singapore [77]. Comparing SPAC and seismic interferometry based on their mathematical relationship [82], SPAC is more suitable to be applied to heterogeneous media.

To clearly image the temporal evolution of the water leakage process, we adopt our dSPAC system methodology in a multistage experiment consisted of a buried pipeline that we buried in an open field area. Our belief is that our method can illuminate the velocity variation produced by water saturation. We developed an experiment that includes: i) made two small holes to a pipeline of 4 ft; ii) bury the pipeline in an open area in a deep of 0.5 meters; ii) deploy 13 smart sensors in the area; iii) inject pressure water to the pipeline to produce a leakage; iv) evaluate the results.

3.5.1 Deployment Specifications

To evaluate if dSPAC algorithm and system is able to detect velocity variations produced by water, we buried a pipeline at Watkinsville, GA. during September 21-25, 2018. Fig. 3.14 shows the Google Maps view of the deployment and the location of the water supply. The process of burying the pipeline is shown in Fig. 3.15

Figure 3.14: Deployment location in Watkinsville, GA. Water supply location.

Figure 3.15: Buried Pipeline at Watkinsville, GA area.

Thirteen seismic nodes were used for this test, and they formed a seismic mesh network for communication and collaboration. Due to SPAC constraints regarding the distance between sensor nodes, the mesh configuration must be a circular array. The approximate distance between adjacent sensors was 3 meters; they were located over the buried pipeline area. The target underground pipeline is located under the surface at an approximate depth 0.5 m, as the schematic map shown in Fig. 3.16.

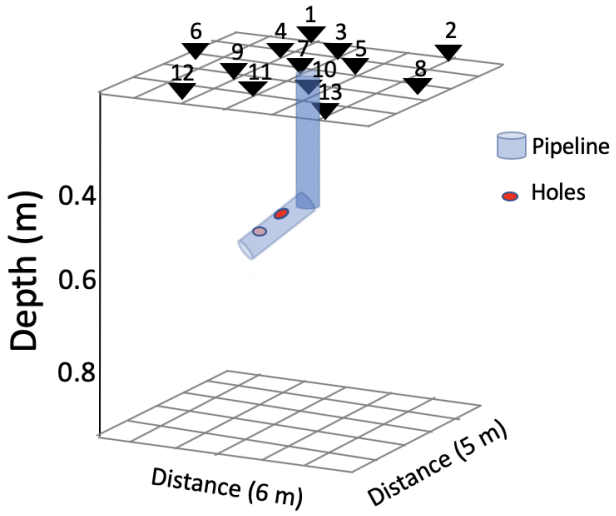


Figure 3.16: Schematic map of buried pipeline and the positions of the holes for water leakage.

The deployment (Fig. 3.17(a)) has three stages: first, we calculate the velocity model of the dry soil without pipe buried; second, after we buried the pipe, the near surface velocity was calculated; third, we use the hose on the pipe, as shown in Fig. 3.17(b), to inject water, then the water leaked through the holes. We apply the dSPAC to monitor the velocity variations over the time. The whole deployment lasted for 4 hours.

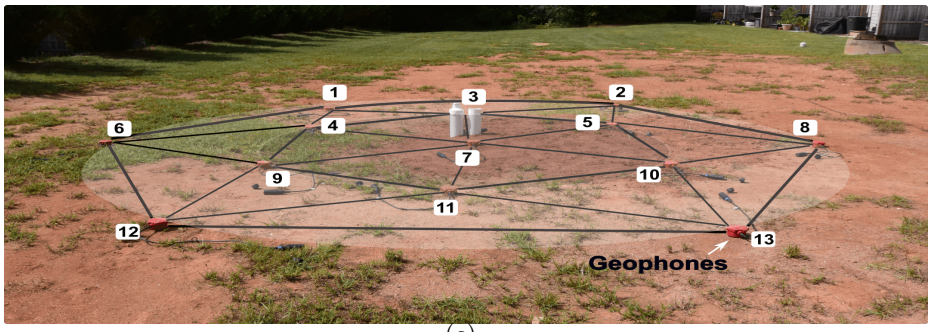
3.5.2 Results

We calculate the correlations between the recorded signals between each node pairs. Each node correlates its signal with its neighboring signals every λ time and stacks them together according to the user needs, which enhances the signal-to-noise ratio (SNR)[1]. We configured λ to be 5 minutes, which means that the system correlates every 5 minutes of data. Fig. 3.18 shows an example of symmetry correlation we obtain between nodes 5 and 7.

Because of the deployment design shown in Fig. 3.17(c), there are multiple ways to construct correlation pairs. Thus, the correlation calculation density is different for different

[1]SNR is the ratio of the strength of a signal carrying information to that of interference, generally expressed in decibels.

(a) (b)



(c)

Figure 3.17: (a) Deployment location at Watkinsville, GA. (b) Injection of water into the pipeline via a hose. (c) The thirteen sensors form a circular seismic survey. For illustration purposes, we only show the geoohones.

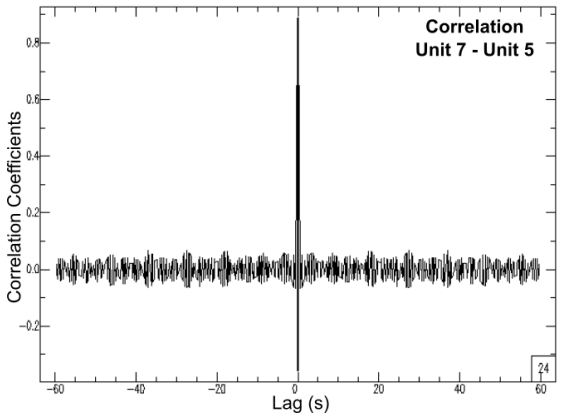


Figure 3.18: Symmetric correlation result from nodes 5 and 7 in the deployment.

nodes, resulting in different imaging confidences. We show the number of correlation pairs in Fig. 3.19. The deployment center has the largest correlation density, so the confidence is the highest, and so forth the boundaries of the deployment have lower confidences. According to the largest correlation pair number, we use 80 % of the peak pair number as a threshold to circle out a confident area. The dished line in Fig. 3.19 is the 80 % confidence contour, which is also marked in the following figures.

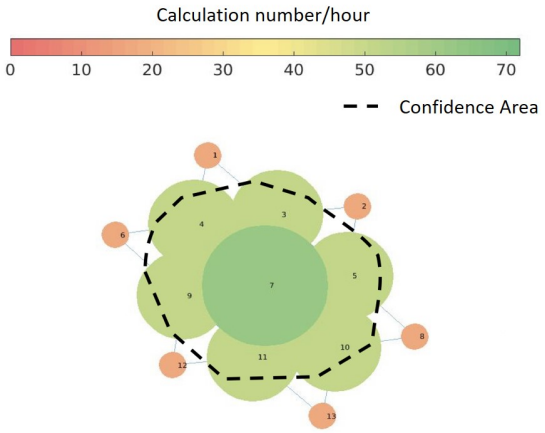


Figure 3.19: Confidence map of the SPAC deployment.

Commonly, seismology imaging techniques intend to adopt lower frequencies to map deep structures and avoid spatial aliasing [6]. Lower frequency components penetrate deeper than higher frequency components, while higher frequency components with shorter wavelengths are sensitive to properties of shallower layers [83]. In our study, because of small inter-station distances (interval is only 3 meters), high frequency components are available for near surface imaging.

Fig. 3.20 shows the SPAC coefficients at different frequencies. According to velocity sensitivity kernel theory [84] and our experiment setup, the frequency component between $\sim$ 80 Hz and 110 Hz is used to calculate the velocity. The velocity $c(\omega)$ can be estimated from the SPAC coefficients ρ. Since a regularly circular array is adopted, frequency ω is the only

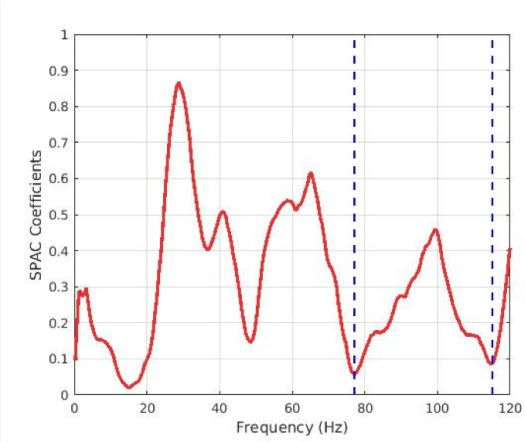


Figure 3.20: SPAC coefficient against frequency.

variable. Thus, we can use least-squares estimation to solve the phase velocity:

$$\min \quad \text{RMS}\left\{\sum^{\omega}\left[\rho(r,\omega_i) - A(\omega_i)J_0\left(\frac{r\omega_i}{c(\omega_i)}\right)\right]\right\}, \tag{3.10}$$

where r is fixed for a given deployment, $A(\omega)$ is a frequency dependent factor defined in Weemstra et al. [85] and RMS$\{\cdot\}$ denotes the root mean square function. According to Weemstra et al. [85], $A(\omega)$, with a value between 0 and 1, needs to be introduced to account for the cross-terms. For example, $u(x_i,\omega)$ and $u(x_j,\omega)$ are not individually ensemble averaged and hence have a high probability of being different valued for different time windows. This is due to the random phase of the cross-terms.

Note that the velocity estimation is constrained by the frequency range. As the imaging resolution and sensitive depth range are directly related to the frequency component selection, we can select the specific frequency range to image certain objects at a given depth [86]. Because different correlation pairs and different time periods could have different velocities, we show not only the final velocity curve but also the confidence area in Fig. 3.21.

Based on the selected frequency range, we can generate continuous near surface velocity

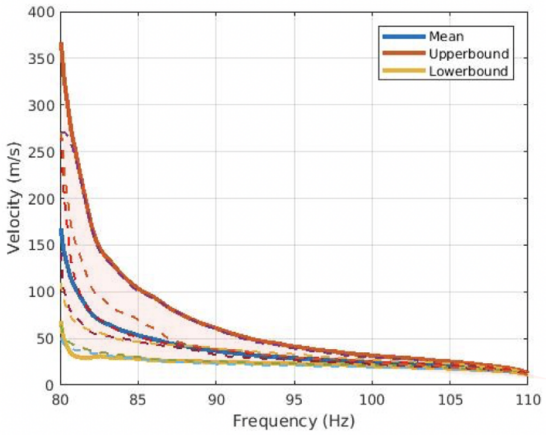


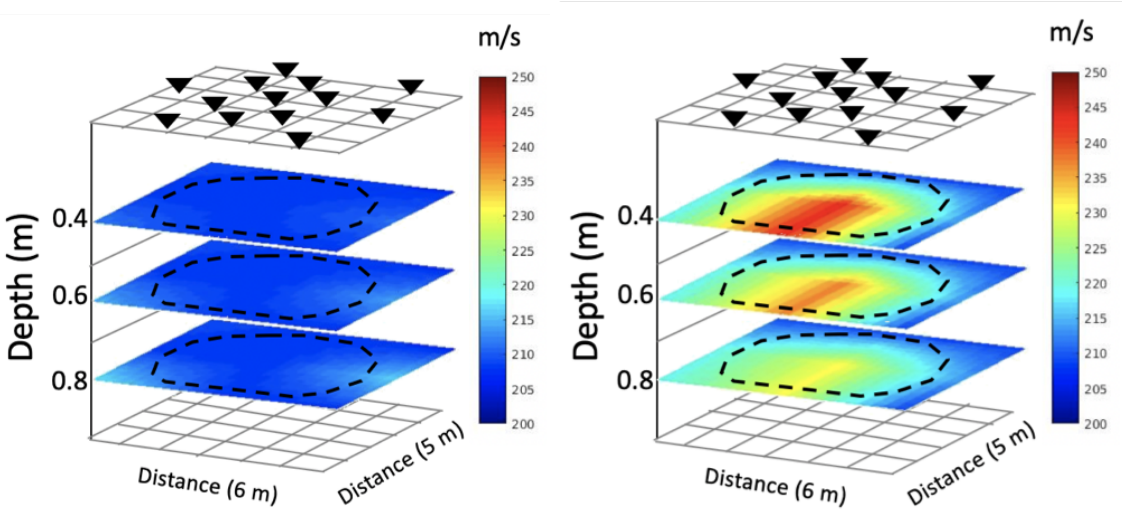
Figure 3.21: Velocity estimated from Eq. (3.10) on node 7. The shadow area indicates the possible velocities, and the blue line is the final velocity curve.

models. Fig. 3.22(a) shows the initial velocity before we buried the pipe, and Fig. 3.22(b) shows velocity model after the pipe was buried but before the water injection. It is clear that before the pipe was buried, the velocity is quite uniform in the deployment area, then the velocity model changes because of the buried pipe.

The subfigures in Fig. 3.23 display the time variant velocity models. Because of the water leakage, the velocity near the pipe location decreases with the time. Notice that, because of the holes shown in Fig. 3.16, the velocity variations are not uniform in all directions and areas.

3.5.3 Evaluation

To further analyze the velocity variation pattern, we show the velocity changes in Fig. 3.24. It is interesting to notice that the velocity differences were larger between 1 hr and 3 hr than the beginning and the end of the deployment. The histograms in Figs. 3.25(a) and 3.25(b) also validate our observations that the velocity decrease rate is larger between 1 hr and 3 hr that the other time periods. Our interpretation is that at the beginning of water leakage the



(a) (b)

Figure 3.22: (a) Velocity structure without pipelines. (b) Velocity structure with buried pipelines.

water saturation is not high enough to significantly affect the velocity of the whole survey, and in the end the soil is close to full saturation, the velocity change is also small. In addition, Fig. 3.24(a) shows the initial velocity variations in the first hour, and we can find besides the pipeline surroundings, another velocity decreasing area is perpendicular to the pipe and located at the hole location shown in Fig. 3.16, which means although the velocity variation over the whole survey in the first hour is small, the water leakage location can be highlighted.

Fig. 3.26 shows the velocity variations on different nodes. The solid curves belong to the nodes within the confident zone, while the dishes curves show the velocity changes outside the confident zone. As we expected, in the confident zone, the velocity changes follow our understanding, but in the boundary areas the velocity could vary randomly within a small range or increase because of unknown reasons.

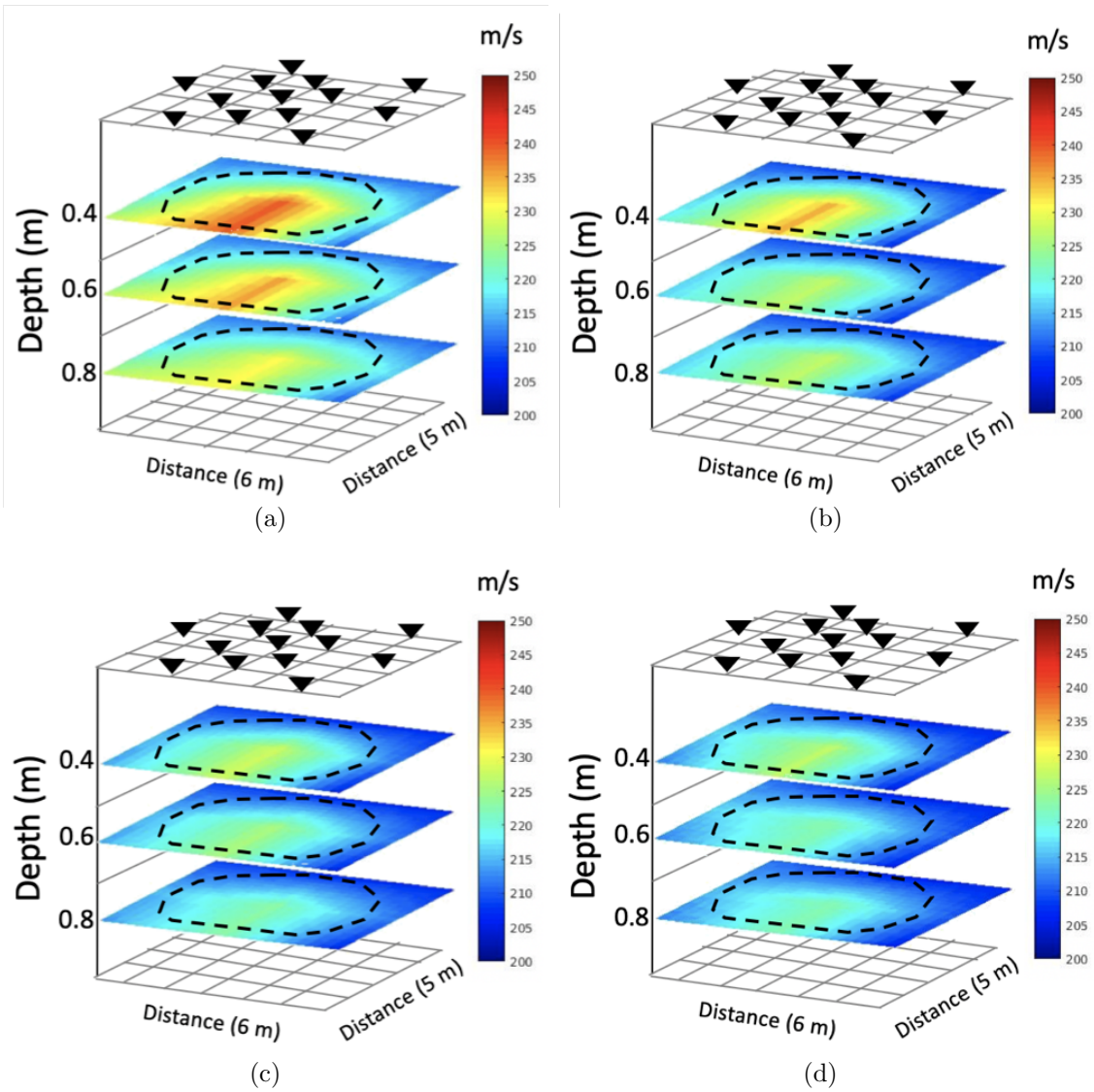


Figure 3.23: Velocity structures after (a) one (1) hour, (b) two (2) hours, (c) three (3) hours, and (d) four (4) hours of water leakage.

3.6 Potential Work

Pipe detection and location is not the end of our work. The principle of the proposed subsurface imaging is to highlight the velocity difference of various structures and subjects. Thus, our system is promising for steam/water leakage detection since the fluid will dramatically reduce

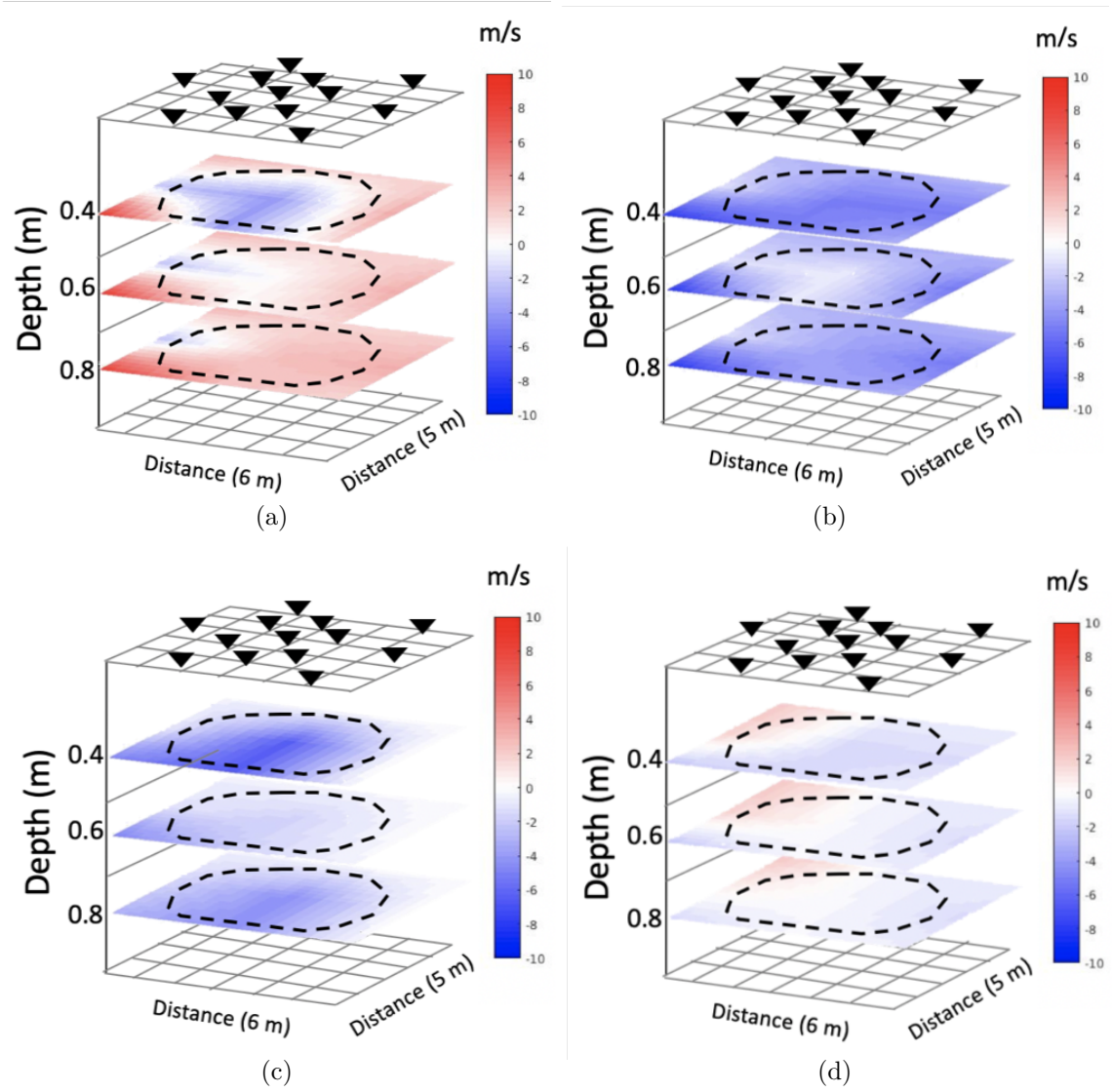


Figure 3.24: Velocity variations from (a) 0 hour to 1 hour, (b) 1 hour to 2 hours, (c) 2 hours to 3 hours, and (d) 3 hours to 4 hours. Histogram of velocity variations from (e) 0 to 2 hours and (f) 2 to 4 hours.

the seismic propagation velocity. In addition, the leakage location should be along the pipe system, so the high-velocity pipe shape imaging result associated with a low-velocity area can infer the leakage. Furthermore, since the imaging technique is sensitive to the fluid. Our

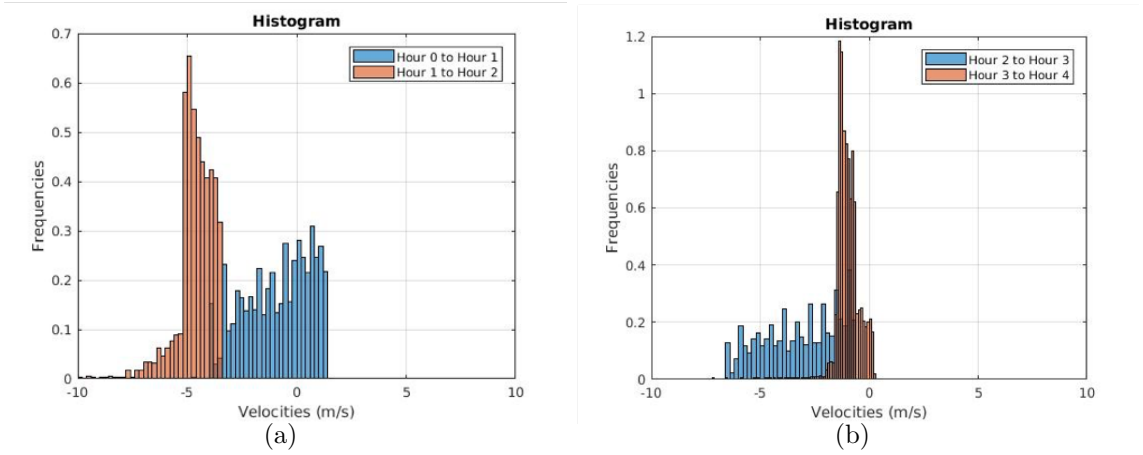


Figure 3.25: Histogram of velocity variations from (a) 0 to 2 hours and (b) 2 to 4 hours.

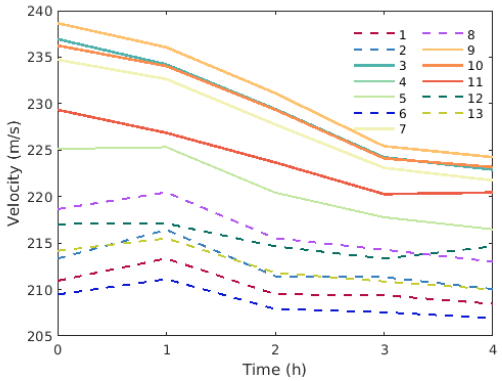


Figure 3.26: Velocity changes over the 4 hours deployment at different node locations due to the water leakage.

system can also be used for underground water detection, for agriculture, watering system and infrastructure security. Another important key point of this work is the geometry of the sensor network. In SPAC methodology, the array configuration plays an important role. Nodes require to be placed at the same distance from a central station.

Other studies have been reported using a seven-station "hexagonal array" [87]. The hexagonal array has the advantage of yielding independent estimates of the SPAC coefficients

over four radial distances simultaneously. As future work, we want to propose a new method that allows different kinds of array configurations to apply SPAC. On the other hand, in anticipation of the upcoming InSight mission [88], which is expected to deploy a single seismic station on the Martian surface in November 2018, we envision our system design and implementation can be used for extraterritorial explorations. In the past, the correlation of seismic noise has been utilized to measure the subsurface velocities in extraterritorial bodies like moon [89]. We think, our system can be deployed in environment-resistant sensors that can use a sensor network to study underground properties in other planets.

3.7 Conclusion

In this chapter, an autonomous, collaborative and non-invasive system for subsurface structure detection is presented. We use spatial autocorrelation as the main methodology for correlating the ambient noise of the subsurface and obtain reliable velocity maps and 3D shapes to analyze the underground structures. Our approach is general and can be applied to detect structures and study underground properties at the same time. We integrate in-network signal processing techniques and in-network tomography computing in a self-healing system that can deliver results autonomously and can be deployed in harsh environments. We demonstrated the system performance by deploying the sensor nodes in a zone with underground pipelines and recovering the velocity variation that represents the pipeline structure.

We also applied the method for water leakage detection. The principle of this application on subsurface imaging is to highlight the velocity variations caused by the fluid leakage in the soil. Based on the seismic ambient noise imaging method, we implement a dSPAC monitoring, and the result clearly shows the velocity change with high spatio-temporal resolution. Thus, our system is promising for steam/water leakage detection by monitoring its associated seismic velocity change. In addition, the leakage location should be along the pipe system.

Combining the imaged pipe and the velocity drop area, we can further infer the leakage location. Furthermore, since the imaging technique is sensitive to the fluid. Our system can also be used for watering system monitoring in agriculture.

3.8 Publications

The work presented in this Chapter has led to the following publications

- Valero, M., Li, F., Clemente, J., & Song, W. (2019). **Non-invasive water leakage detection via ambient noise processing in distributed sensor networks**. In preparation for submission to: IEEE Transactions on Signal and Information Processing over networks.

- Valero, M., Li, F., Li, X., & Song, W. (2018). **Imaging Subsurface Civil Infrastructure with Smart Seismic Network.** In Proceedings of the 37th IEEE International Performance Computing and Communications Conference (IPCCC). (pp. 1-8). IEEE. DOI: 10.1109/PCCC.2018.8711309

Chapter 4

Optimized Communication for In-situ Ambient Noise Imaging

4.1 Motivation

Seismic sensors are currently used for gathering seismic data that is later processed to obtain subsurface images using ambient noise methods [90]. Current approaches employ image reconstruction methods that rely on a centralized approach for processing the raw data captured by these seismic sensors. A solution can be the use of wired sensor communication approaches; however, the length limitation in cable communication represents a problem in large deployments. In addition, the centralized processing and computing style is not capable of being implemented in-situ and real-time subsurface imaging in all circumstances, especially in harsh environments [91]. It seems that a good solution for introducing in-situ and real-time imaging on sensors is wireless communication. For example, wirelessly connected sensors were deployed using an air-dropped way to monitor live volcano activities, where communication and computation become bottlenecks [92]. Recently, seismic tomography has been implemented using advanced wireless sensor networks with distributed computing

algorithms [10, 78, 93]. The distributed style has advantages in reducing the data loss risk in case of node and cable failures, because the sensing, computing, and data storage tasks can be operated in the sensor nodes [94]. Instead of collecting data into a processing center, distributed seismic data processing and computing can be performed on individual sensors with communications among the local sensor array. Even though system-level challenges of deploying wireless sensor networks exist, focusing on distributed in-network signal processing and computation can help support real-time tomographic imaging.

In distributed ambient noise imaging the most intensive communication process is the transfer of data for cross-correlation [10]. To cross-correlate data, every node has to send its own data to the neighbor nodes. Even though some communication-reduction techniques can be applied (reduction of data using a pre-specific narrow band of frequencies, compression techniques, etc.), sending data to all neighbors can be inefficient. Furthermore, some nodes waste computational performance doing cross-correlations that can be done in other nodes.

In this Chapter, besides utilizing distributed and in-network computing to imaging shallow subsurface, we propose a new communication-reduced method that can be applied in ambient noise imaging based on Distributed Spatial Auto-correlation (dSPAC). The method is designed as a combinatorial optimization problem that first transforms the topology of the mesh network in a suitable graph for a transportation problem [11]. Then, the optimization problem is solved to get the best nodes for computing cross-correlation to meet network limitations. Constraints regarding bandwidth are added. Furthermore, constraints regarding the energy of the sensors can be added too to make a more energy-efficient selection. By applying the proposed communication-reduced method, we show that energy and computational cost of the nodes is also preserved. We show in our experiments that it is possible to image buried pipelines using our method, and potentially detect water leakage.

4.2 In-situ Cross-correlation and dSPAC for Ambient Noise Imaging

Similar to the dSPAC workflow presented in Chapter 3, the following Fig. 4.1 represents the algorithm scenario used in this methodology.

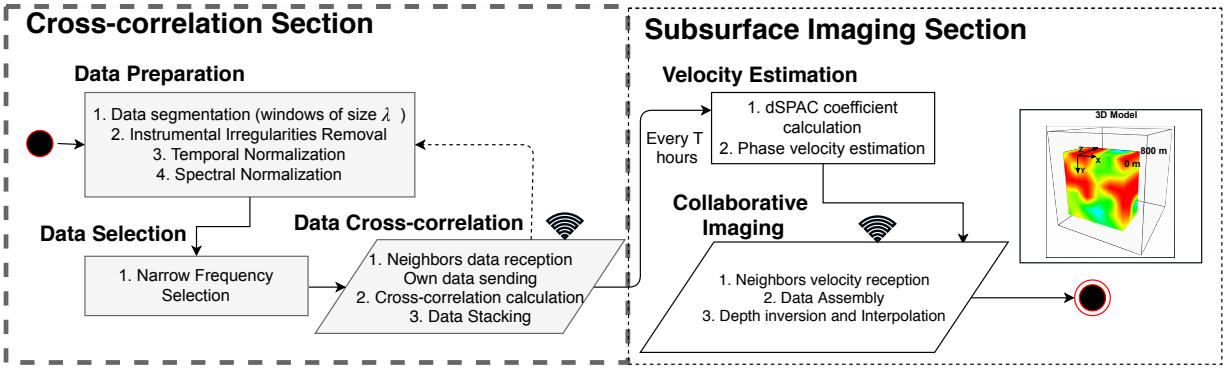


Figure 4.1: System methodology for dSPAC subsurface imaging.

he main of the methodology idea is every sensor cross-correlate its data with the data of its neighbors in a time window λ. To do so, nodes broadcast their data. However, instead, to broadcast the raw data, nodes broadcast a prepared and selected data to diminish the communication cost. The cross-correlation process is continuous. Once a time T is complete, nodes perform the subsurface imaging by estimating the velocity variation using dSPAC method. Then, a collaborative image is produced by aggregating in a collaborative and distributed way the velocity estimation of each node. In this section, we introduce the complete overflow by presenting the mathematical framework and algorithm that includes each step. Then, we discuss why even though the method produces promising results, the communication cost needs to be improved.

4.2.1 System Model

Each sensor is provided with a radio system to communicate with the rest of the network, but each radio system has only a limited range for transmissions and receptions. It is assumed that the transmission range and the reception range are the same, and it is referred to as communication range. Consequently, each node is able to communicate with a restricted number of other sensors, the ones deployed within its communication range. In this work, we use an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to model the topology of the network. Each node in $\mathcal{V}$ represents a sensor node, and the link $(i, j) \in \mathcal{E}$ represents that nodes i and j are in communication range and they are considered as neighbors. Let us denote with $\mathcal{N}(i)$ the set of neighbors of node i. Let $|\mathcal{V}|$ denote the number of nodes in the network.

Let $x_i(t)$ be the raw signal of sensor node i in the time t. Every node in $\mathcal{V}$ gathers $\mathbf{x}_i(t)$ and started the preparation for cross-correlation. Here, $\mathbf{x}$ is a vector that contains the readings of ambient noise for a time λ; e.g. λ can be equal to 1 minute, 2 minutes, 5 minutes, etc. (Note that because we are doing continuous monitoring we choose time no-overlapping following [6, 43, 52]. If the seismic survey is time-limited, only 1 or 2 hours, we can use overlapping for improve the convergence of the cross-correlations.)

4.2.2 Signals Cross-correlation and Frequency Selection

After gathering $\mathbf{x}_i(t)$, every sensor performs a *data preparation*. The purpose of this preparation is to accentuate ambient noise by attempting to remove earthquake signals and instrumental irregularities that tend to hide the ambient noise [43]. To *remove instrumental irregularities*, we withdraw the mean and the trend of the signal [45]. Then, we apply running-absolute-mean method [43] for *temporal normalization*. For the raw data $\mathbf{x}_i(t)$, the

normalization weight is

$$W(t) = \frac{1}{2q+1} \sum_{j=-q}^{q} \mathbf{x}_i(m-j) \qquad \text{for } m = q+1, q+2, ..., t-q \tag{4.1}$$

and the normalized datum is $\hat{\mathbf{x}}_i(t) = \mathbf{x}_i(t)/W(t)$. The width of the normalization window is $2q+1$. Finally, *spectral normalization* [43] is applied to reduce broad imbalances in single-station spectra to aid in the production of a broad-band dispersion.

Once data has been prepared, a *data selection* process is conducted. In the data selection, a narrowed band-pass filter is applied to keep only a range of frequency components we need to study. This process is known as *narrow frequency selection*. Let

$$\Psi = \{f_1, ..., f_m\} \subset \{0, 1, ..., N-1\}$$

denote the indices of the narrowed frequency components, which is the same across all sensors. Sensor i only transmits the subset of frequency samples $\{\mathbf{X}_i(f_k)\}$, where $f_k \in \Psi$. Assuming that $m = |\Psi|$ frequency samples are selected; then, the amount of data to be transmitted is reduced from $\mathcal{O}(N)$ to $\mathcal{O}(m)$. In this case, $N \gg m$ because we have to observe long enough noise sequence and the frequency band we are interested is usually narrow. Most importantly, if the time t of stacking cross-correlation is no large (for example between 1-10 minutes), and the frequency band is narrow - for example, between 80-100Hz, we are able to achieve $80\% \sim 90\%$ of reduction, and the data to transmit can be sent in only one UDP or TCP packet. This achievement significantly reduces the total communication overhead.

Node i then broadcast $\mathbf{X}_i(f_k)$ to $\mathcal{N}(i)$. Note that, at the same time, node i receives data that come from every node than belongs to $\mathcal{N}(i)$. For each node $j \in \mathcal{N}(i)$, node i computes

$$C_{\hat{\mathbf{x}}_i\hat{\mathbf{x}}_j} = \hat{\mathbf{x}}_i \otimes \hat{\mathbf{x}}_j \equiv \frac{1}{2T} \int_{-T}^{T} \hat{\mathbf{x}}_i(\tau)\hat{\mathbf{x}}_j(\tau+t)\mathrm{d}\tau = \mathbf{X}_i(f_k) \cdot \overline{\mathbf{X}_j(f_k)} \tag{4.2}$$

where T is the total time of cross-correlation and $\overline{\mathbf{X}}$ indicates the complex conjugate of $\mathbf{X}$. The cross-correlation is $C_{\tilde{\mathbf{x}}_i\tilde{\mathbf{x}}_j}$ then stacked with itself every time nodes send and receive data from its neighbors. The stacking process is usually employed to increase the signal-to-noise ratio (SNR) of the signal [47]. In this case, we stack the cross-correlation results every time it is performed every λ minutes, where λ can be 1,2 or 5 minutes depending on the system configuration, until completing T time. For instance, if λ is 5 minutes, the node correlate the data and then stack it with the previous stacked 5 minutes and so on.

We use the symmetric component of the cross-correlation that is the average of the cross-correlation at positive and negative lags [43]. Thus, note that $C_{\tilde{\mathbf{x}}_i\tilde{\mathbf{x}}_j} = C_{\tilde{\mathbf{x}}_j\tilde{\mathbf{x}}_i}$. Also, note that since nodes broadcast to all their neighbors the cross-correlation $C_{\tilde{\mathbf{x}}_i\tilde{\mathbf{x}}_j}$ is calculated in both node i and node j, which implies a wasting of computation (See Fig. 4.2(a)). When the system has complete T time that involves multiple cross-correlations and stacking processes, the subsurface imaging section begins the distributed calculation of the velocity variation structure.

4.2.3 Subsurface Imaging

After cross-correlation, every node in $\mathcal{V}$ estimates locally the SPAC coefficients [64].The SPAC method has exhibited a good performance in heterogeneous and isotropic media [66], and has shown a comparable output with other tomography methods like interferometry [82]. Even if we use high frequencies study, the SPAC method holds even for a non-isotropic wavefield, because the normalized cross-spectrum can be averaged with respect to various incident directions by using the wavefield at the centre of a circle and the wavefields on a circumference of the circle [82]. We called the method dSPAC because it is the distributed version of SPAC. The dSPAC method can extract the phase velocities of surface waves from microtremor array observations. The basic theory of the spatial auto-correlation method [66] is summarized as follows. Having an array of sensors (called receivers) equally spaced on a

circle of radius r and having an extra receiver at the center as shown in Fig. 3.4, the phase velocities ($c(\omega)$) can be calculated.

As explained in Chapter 3, the dSPAC coefficient can be estimated as the average of the cross-correlation between every node pair in a fixed geometry with the same ratio r, which remedies the biases in phase velocity measurements caused by a non-isotropic or directional wavefield. In other words, Eq. 3.4 can be rewritten as:

$$\rho_i(r,\omega) = \frac{1}{|\mathcal{N}(i)|} \sum_{j=1}^{|\mathcal{N}(i)|} C_{\tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_j} \equiv J_0\left[\frac{\omega}{c_i(\omega)} r\right] \tag{4.3}$$

where i is the central sensor of the circular array. *The phase velocities are estimated by fitting the observed dSPAC coefficients to the Bessel function.* Note that, a larger array in a circular topology can have multiple circular sink nodes.

After the velocities $c_i(\omega)$ are estimated, the sink nodes at each ring broadcast the velocity information to the other sink nodes, and they perform an interpolation process to form a 3D map of the subsurface with all the frequencies in consideration. Each layer of the 3D map represents a subsurface depth. With this information, we can analyze the velocity variations and determine the presence of structures, like pipelines, within the subsurface.

4.2.4 Limitations of broadcasting to all neighbor nodes

The main limitation of distributed subsurface imaging [10] is the broadcasting to all neighbors without distinguishing between them, and the multiple and needless computation of the same cross-correlation in different nodes. If $C_{\tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_j} = C_{\tilde{\mathbf{x}}_j \tilde{\mathbf{x}}_i}$, it should be convenient to process the cross-correlation only one time. Furthermore, if we propose a mechanism to select the best nodes to compute cross-correlation meeting the bandwidth and communication constraints, some nodes can help others by computing cross-correlation between neighbor nodes too. In the next section, we introduce the proposed communication-reduce model for subsurface

imaging using dSPAC to get the best nodes for computing cross-correlation to meet network limitations.

4.2.5 Scope of the Proposed Model

The proposed communication-reduced method focuses the attention in the cross-correlation section. We aim to reduce the number of cross-correlation and select the best nodes to compute them. Our results and analysis are based on the performance of the distributed system during this correlation section. The subsurface imaging section results are discussing results section. The idea is to present a model that can improve the communication cost and bandwidth utilization in the first part of the ambient noise process, which is the most communication intensive. In the next section, we detail the proposed model and defined the main mathematical framework of the solution.

4.3 Communication-reduced Model for dSPAC

In this section, we present the communication-reduced model for dSPAC in ambient noise imaging. Specifically, we improve the communication cost and bandwidth utilization in the *correlation section* of the system, where the communication for cross-correlation and stacking is the most intensive process. First, we explain why even though a reduction of the data is performed in dSPAC methodology, the communication is still inefficient. Furthermore, the computation cost is high on all the nodes in the network. Later, we present the model for selecting the best nodes to compute cross-correlation, and the optimization model to guarantee an appropriate solution.

The naive communication pattern (Fig. 4.2(a)) may include each sensor broadcasts its narrowed-data to its neighbors, and each one of them perform cross-correlation. However, after stacking cross-correlation the result is symmetric [4]. This means the cross-correlation

coefficients will be the same from sensor 1 to sensor 2, and vice-versa.

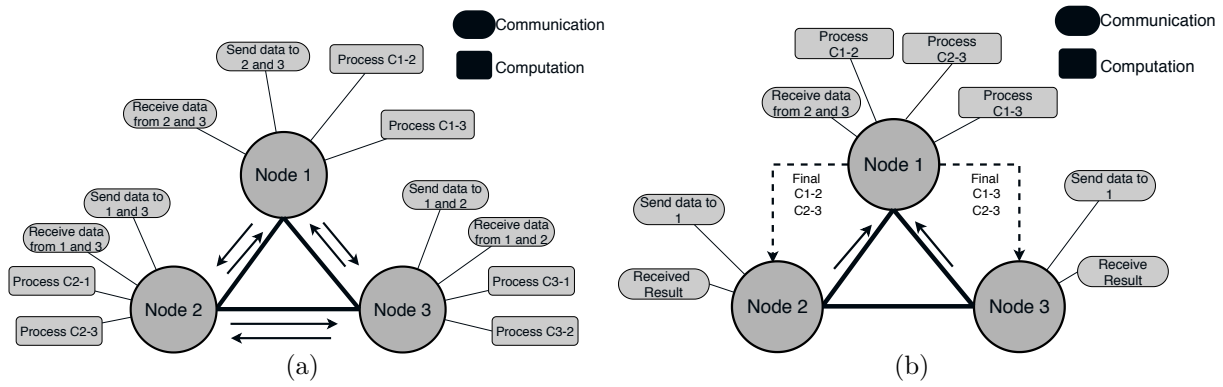


Figure 4.2: (a) Current communication pattern. (b) Proposed communication-reduced method example.

By taking advantages of symmetric cross-correlation [95], we can assume that one node (A) may compute the cross-correlation of its neighbor (B), and then forward the results. That means (A) does not send narrowed-data, only receive from (B) and computes C_{AB}. Furthermore, if (K) is neighbor of (A) and (B), sensor (A) may also compute the C_{BK} among (B) and (K). Fig. 4.2(a) shows the current communication and computation pattern to perform cross-correlation. Each sensor is represented as an independent "node". Every t time (t could be equal to five minutes for example), nodes broadcast its data to their neighbors. For example, node 1 receives data from nodes 2 and 3 every 5 minutes. Node 1 computes the cross-correlation between itself and nodes 2 and 3. The process is the same in node 2 and 3. Notice that node 2, for example, also computes cross-correlation 1 and 2. This is also a waste of computation. If only node 1 receives and computes, the communication cost and computation cost would be reduced significantly.

The proposed method for solving this problem is then illustrated in Fig. 4.2(b) where nodes 2 and 3 send data only to node 1. Furthermore, to compute the cross-correlation between node 2 and 3, there is no need for communication between those nodes because node 1 can compute this cross-correlation too. The number of packets sent over the network

is significantly reduced. Node 1 continues receiving data from 2 and 3 and stacking the cross-correlations until completing time T that is the time to begin the subsurface imaging section. At that moment, node 1 forwards back the results to node 2 and 3.

However, in larger mesh networks, with more complex topology, finding the solution is not straightforward. An optimization scheme has to be formulated to solve the problem in the most efficient way. We designed a new communication-reduced method for cross-correlation. The method is designed as a combinatorial optimization problem that first transforms the topology of the mesh network in a suitable graph for transportation problem. Then, the optimization problem is solved to get the best nodes for computing cross-correlation. Constraints regarding bandwidth are added. Furthermore, constraints regarding the energy of the sensors can be added too to make an energy-efficient selection.

4.3.1 Problem Definition

In our model, we have a mesh network of sensors represented as a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \lambda)$, where each edge $(u, v) \in \mathcal{E}$ has a transmission cost λ_{uv}. The network $\mathcal{G}$ is assumed to have a suitable topology for dSPAC computation. Fig. 4.3 shows an extended topology example that can be deployed for dSPAC-based ambient noise imaging. Note that other kind of topologies, like hexagons or triangles, can be also used. Nodes only can communicate directly with their direct neighbors, and the data they send should be computed at most in one hop of distance. We consider the data that each node sends during the *correlation section*.

Assumption 1 (Fixed Packet Size)**.** *We assume without loss of generality that the size of the packets generated at the data producers have the same size and we consider each packet as a data element.*

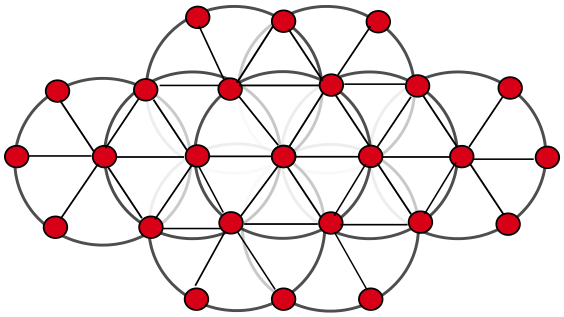


Figure 4.3: dSPAC topology example.

4.3.2 Network Transformation

Before we can find the optimal solution to our problem, we need to transform our original network graph $\mathcal{G}$ into a flow network $\mathcal{G}'$. We propose to use neighborhood information for this network transformation. A series of steps have to be performed in order to transform $\mathcal{G} \rightarrow \mathcal{G}'$:

- We set up every node in $\mathcal{G}$ as a *transport node*. The set of transport nodes is represented as $\mathcal{P}$ (Fig. 4.4 - Layer 1). The transport nodes are considered to be the nodes that may compute the cross-correlations, and they can be constrained with the maximum number of cross-correlations to compute on it ($maxC_i$ where $i \in \mathcal{P}$). If we assume that the computational cost is unimportant, then $maxC_i$ can be infinite. However, that is not the case for many nodes due to energy consumption. *Every transport node produces a cost of λ for every unit that sends to the second layer of nodes.* By default, nodes in $\mathcal{P}$ sends one unit to the second layer.

- We set up a set of *intermediate nodes with gain $\mathcal{Z}$* as the second layer of the flow network (Fig. 4.4 - Layer 2). This layer is composed for the neighbors of each node $i \in \mathcal{P}$. For example, if in the original network node 2 ($i = 2$ and $i \in \mathcal{P}$) has two neighbors (node 3 and node 5), then it will be added two new nodes to layer 2 ($\mathcal{Z}$),

called $r3$ and $r5$ and those are going to be directly connected to node 2 in layer 1 ($\mathcal{P}$). Every node in $\mathcal{Z}$ receives one unit from the transport nodes, and it generates half unit (0.5) for each connection with the third layer, or they generate 0 units if they do not have neighbors. There is no cost of transporting data from layer 2 to other layers.

- We set up the third layer as the set of *intermediate nodes without gain* $\mathcal{Y}$. This layer of nodes (Fig. 4.4 - Layer 3) is composed by the neighbors of node $j \in \mathcal{Z}$ in layer 2 that also are neighbors of $i \in \mathcal{P}$ in layer 1. This layer is used to analyze the neighbors that can be also capable to compute the cross-correlation of other neighbors but not more than one hop of difference. For instance, the node $r3$ from $\mathcal{Z}$ that is connected to node 2 from $\mathcal{P}$ is, itself, neighbor of node 5 in the original network; node 2 is also neighbor of node 5 in the original network, then, we add the node $r3r5$ to the layer 3 because node 3 is neighbor of 5 and both are neighbors of node 2 in the original network. This layer does not generate any cost for unit.

- The four and last layer is composed of all the possible cross-correlation between neighbors the system needs to compute. For example, in the original network of Fig. 4.4, we need to compute the cross-correlations C_{1-5}, C_{2-5}, C_{2-3}, C_{3-5}, C_{3-4}, C_{4-5} because those are the neighbors (there is exists a edge) of the nodes in the network.

After the node transformation, we can set the optimization problem over $\mathcal{G}'$ by minimizing the communication cost in that network. An important comment regarding layer 2 is that, as mentioned, those nodes generates 1/2 or 0.5 unit for each connection to layer 3. To clarify this issue, consider the following example: In Fig. 4.4, node 2 (layer 1) is connected to nodes $r3$ and $r5$ in layer 2 and transfer one unit at a cost λ. Then node $r3$ generates 1/2 unit and send it to node $r3r5$ (layer 3); similarly, node $r5$ generates 1/2 unit and send it to node $r3r5$ too. This makes a total of 1 unit to node $r3r5$ and guarantees that the optimization problem can study this possible solution.

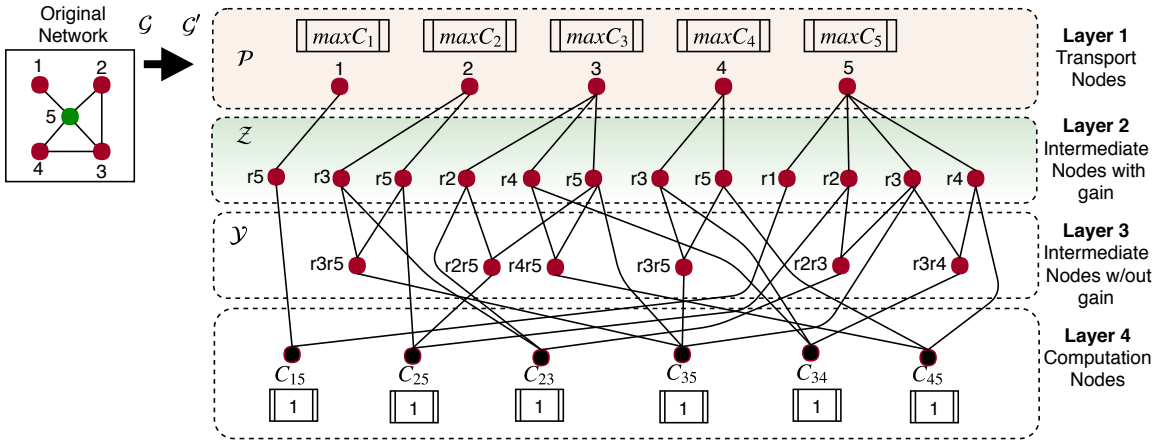


Figure 4.4: Network transformation example.

After graph transformation, we can set up the optimization problem for minimizing the communication cost between the nodes by selecting the best nodes to computing cross-correlations.

4.3.3 Model Design

We want to minimize the cost of transmission (λ) of the number of packets (β) that require cross-correlation. The idea is to perform all the needed cross-correlation with the minimum cost of transmission. Let λ_{uv} be the cost of transmission from node u to node v, and β_{uv} the number of packets to transmit from node u to node v. The maximum number of packets to transmit simultaneously by a node u is defined by $\chi_u = \Omega_u / S_u$ where Ω_u is the available bandwidth in node u and S_u is the size of the packet.

Table 4.1 summarize the main variables of our optimization problem. We define the optimization problem as:

Table 4.1: Communication model variables.

Variable	Description
λ_{uv}	Communication cost between nodes u and v.
β_{uv}	Number of Packets between nodes u and v.
χ_u	Maximum number of packets to transmit simultaneously by a node u. ($\chi_u = \Omega_u / S_u$).
Ω_u	Available bandwidth.
S_u	Size of the packet to be sent by u.
h	Gain for each node connection between layer 2 and layer 3 in $\mathcal{G}'$.
Γ_u	Set of nodes connected by output edges with node u Outflow.
Υ_u	Set of nodes connected by input edges with node u Inflow.
r	Set of intermediate nodes with gain.
g	Set of intermediate nodes without gain.

$$\text{minimize} \quad \sum_{uv \in E} \lambda_{uv} \beta_{uv} \tag{4.4a}$$

s.t:

$$-\sum_{l \in \Gamma_u} \beta_{ul} \leq \chi_u \tag{4.4b}$$

$$\sum_{k \in \Upsilon_u} h\beta_{ku} - \sum_{l \in \Gamma_u} \beta_{ul} = 0 \tag{4.4c}$$

$$\sum_{k \in \Upsilon_u} \beta_{ku} - \sum_{l \in \Gamma_u} \beta_{ul} \geq 0 \tag{4.4d}$$

$$\beta_{ku} = \frac{1}{2} \forall k \in r, u \in g$$

$$\sum_{k \in \Upsilon_u} \beta_{ku} = 1 \tag{4.4e}$$

The objective function (4.4a) is to minimize the communication cost between nodes in

the network to meet bandwidth specifications. Constraint 4.4b is established for layer 1 (*transport nodes*), and it guarantees that the number of packets to be transmitted will be less or equal to the maximum number of packets to transmit simultaneously in the edge. Note that l in β_{ul} belongs to the set of nodes connected by output edges with node u where u is the node in consideration in layer 1. Constraint 4.4c is established for layer 2 (*Intermediate nodes with gain*). This constraint guarantees that the inflow packets will be the same than the outflow packets, but it will be a gain of h, where h is 1/2 for each connection to layer 3, and it is calculated using:

$$h = 1 + \frac{|N(k) \cap N(u)|}{2}. \tag{4.5}$$

Note that $N(k)$ is set of neighbors of k, and $N(u)$ is the set of neighbors of u; in consequence $|N(k) \cap N(u)|$ represents the number of neighbors of k that also are neighbors of u.

Constraint 4.4d is established for layer 3 (*Intermediate nodes without gain*). This constraint guarantees that the inflow packets will be equal or greater than outflow packets in this layer; The inflow is equal to the inflow when the node in layer 3 receives packets from its two connection with layer 2 (that means that the cross-correlation can be done); otherwise, the outflow is 0. Note that in this case, for all intermediate nodes without gain the input should be 1/2 for all $k \in r$, where r is the set of intermediate nodes with gain, and $u \in g$ where g is the set of intermediate nodes without gain. Finally, constraint 4.4e is established for layer 4 (*computation nodes*). This is equal to 1 because we want to compute only one cross-correlation per pair.

4.4 Experiments and Evaluation

We conduct a series of experiments to test our communication-reduced model. In this section, we explain the main results and improvements in terms of bandwidth, energy consumption, and computational cost.

4.4.1 Topology Design

Because dSPAC-based method requires a ring topology, we use this kind of arrangement in our experiment. Later, we will show that this topology is also suitable for real-world experiments. The used topology is shown in Fig. 4.5.

To execute the dSPAC method with this specific topology, the total number of cross-correlation that needs to be calculated is 24. This number is based on the number of neighbors that are formed in the mesh network. The list of needed cross-correlations is shown in Table 4.2. For the sake of space, we do not show the graph transformation of the topology in Fig. 4.5.

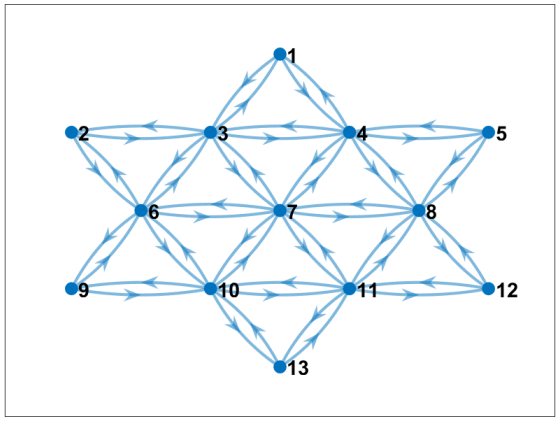


Figure 4.5: Communication pattern without the proposed communication-reduced method. Nodes broadcast data to their neighbors.

Table 4.2: Needed cross-correlations for the Fig. 4.5 topology

Needed Cross-correlations					
C_{1-3}	C_{3-4}	C_{4-7}	C_{6-9}	C_{7-11}	C_{10-11}
C_{1-4}	C_{3-6}	C_{4-8}	C_{6-10}	C_{8-11}	C_{10-13}
C_{2-3}	C_{3-7}	C_{5-8}	C_{7-8}	C_{8-12}	C_{11-12}
C_{2-6}	C_{4-5}	C_{6-7}	C_{7-10}	C_{9-10}	C_{11-13}

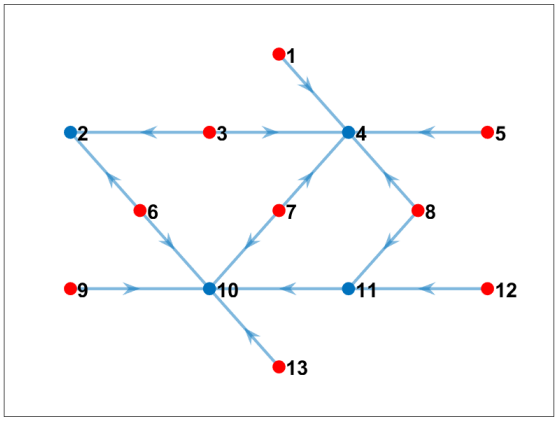
4.4.2 Experiment 1: Unlimited Bandwidth

If we assume all nodes have unlimited bandwidth, and they can send and receive from/to any neighbor node, the proposed communication-reduced model estimates the optimal solution like the one shown in Fig. 4.6. Note that all 24 cross-correlations were calculated in the "best node" and the communication cost is minimum for this scenario. In the Figure, red nodes were selected to be just sender of data, and blue nodes are nodes that compute cross-correlations. Note also, from Table 4.3 that only five (5) nodes use computational resources. However, some nodes, like numbers 8 and 10, compute more than 7 or 8 cross-correlations from its neighbor nodes, which may imply a bottleneck and affect the energy consumption of those nodes. The advantage of this solution is the main sink node (node 7), that is the one that computes the 3D interpolation of the subsurface image, do not waist computation time and energy during the cross-correlation process.

To further measure the bandwidth constraint in the performance of the proposed model, we did two more experiments by limiting the number of neighbor nodes data that a node can receive to compute cross-correlations. We mention the results in experiment 2 and 3.

4.4.3 Experiment 2: Limited bandwidth.

We perform two different tests with limited bandwidth considering that one node only can receive data from up to (i) 4 neighbor nodes, and (ii) 2 neighbor nodes.

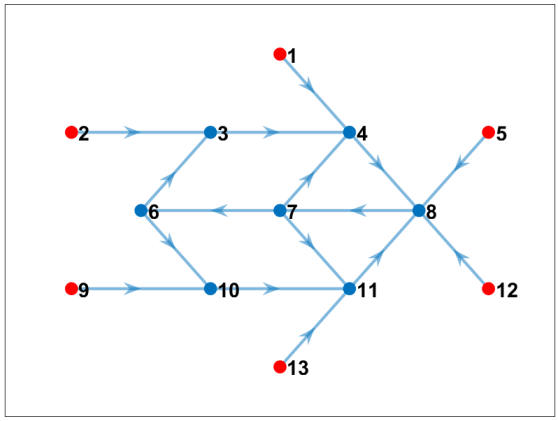


Node Number	Computed Cross-correlation	Node Number	Computed Cross-correlation
2	C_{2-3}	10	C_{6-7}
	C_{2-6}		C_{6-9}
	C_{3-6}		C_{6-10}
4	C_{1-3}		C_{7-10}
	C_{1-4}		C_{7-11}
	C_{3-4}		C_{9-10}
	C_{3-7}		C_{10-11}
	C_{4-5}		C_{10-13}
	C_{4-7}		C_{11-13}
	C_{4-8}	11	C_{8-11}
	C_{5-8}		C_{8-12}
	C_{7-8}		C_{11-12}

Figure 4.6 & Table 4.3: Communication pattern the using the proposed model if we consider unlimited bandwidth (blue nodes are nodes that computes cross-correlation; red nodes just send data to the corresponding network.) & Nodes that compute cross-correlations.

Maximum 4 Neighbors

We conduct an experiment by considering that one node only can receive data from up to 4 neighbor nodes to compute cross-correlation. When we add this new constraint, the optimal solution is shown in Fig. 4.7. Note that, in this case, the communication cost continues being the minimum possible under the constraints, and the computational cost has been more



Node Number	Computed Cross-correlation	Node Number	Computed Cross-correlation
4	C_{1-3}	3	C_{2-3}
	C_{1-4}		C_{2-6}
	C_{3-4}		C_{3-6}
	C_{3-7}	7	C_{7-8}
	C_{4-7}	10	C_{6-9}
6	C_{6-7}		C_{6-10}
8	C_{4-5}		C_{9-10}
	C_{4-8}	11	C_{7-10}
	C_{5-8}		C_{7-11}
	C_{8-11}		C_{10-11}
	C_{8-12}		C_{10-13}
	C_{11-12}		C_{11-13}

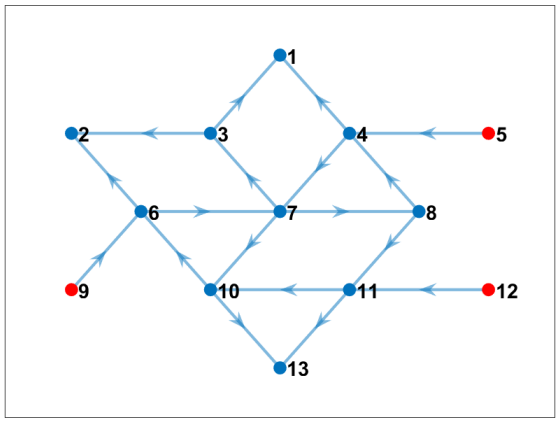
Figure 4.7 & Table 4.4: Communication pattern using the proposed model if we consider that one node only can receive data from up to 4 neighbor nodes (blue nodes are nodes that computes cross-correlation; red nodes just send data to the corresponding network.) & Nodes that compute cross-correlations.

evenly balanced between the nodes in the network as is shown in Table 4.4.

Maximum 2 neighbors

We also conducted another experiment by increasing the bandwidth limitations. We consider that one node only can receive data from up to 2 neighbor nodes to compute cross-correlation.

Fig. 4.8 shows the optimal communication pattern for this scenario. Note that the number of nodes that compute cross-correlations has increased with this bandwidth limitation. From Table 4.5 we can confirm that, for this specific topology and cross-correlation requirements, the maximum number of cross-correlation that a single node needs to compute is 4. Note also that the computational cost for sink nodes is relatively low.



Node Number	Computed Cross-correlation	Node Number	Computed Cross-correlation
1	C_{1-3}	6	C_{6-9}
	C_{1-4}		C_{6-10}
	C_{3-4}		C_{9-10}
2	C_{2-3}	8	C_{7-8}
	C_{2-6}	10	C_{7-10}
	C_{3-6}		C_{7-11}
3	C_{3-7}	11	C_{8-11}
4	C_{4-5}		C_{8-12}
	C_{4-7}		C_{11-12}
	C_{4-8}	13	C_{10-11}
	C_{5-8}		C_{10-13}
7	C_{6-7}		C_{11-13}

Figure 4.8 & Table 4.5: Communication pattern using the proposed model if we consider that one node only can receive data from up to 2 neighbor nodes (blue nodes are nodes that computes cross-correlation; red nodes just send data to the corresponding network.) & Nodes that compute cross-correlations.

4.4.4 Experiment 3: Variable Bandwidth.

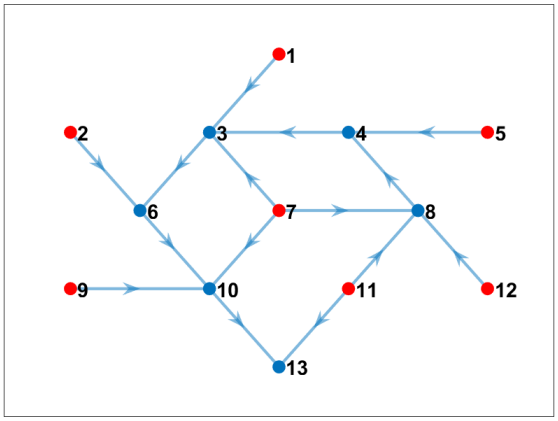
In real scenarios, the bandwidth is variable and depends, among other things, of the number of connections in the topology. In a mesh network, every "*hop*" (link) between sensors will decrease the bandwidth by half [70]. This happens because wireless links can only do one thing at a time - transmit or receive. In a long "chain" of mesh links, this results in a very slow connection from end to end. Even though this estimation (half of the bandwidth decreasing by every link) is widely accepted, in reality, other factors can impact the available bandwidth in a specific time; for example, communication range, other networks interference, etc.

As will explain, our real-world sensors are based on a Raspberry Pi 3 as computer board. The wireless communication bandwidth of Raspberry Pi 3 is estimated at ∼10Mbps (Megabytes per second) [71]. Due to the number of links in our topology (some nodes may have 5 or 6 links, which reduced the available bandwidth), we based our observations on a maximum available bandwidth of ∼2Mbps.

We set an experiment in which we vary randomly the available bandwidth between each pair of nodes depending on the number of links or hops, and we run our optimization method. The "best nodes" to compute cross-correlations and the communication pattern is shown in Fig. 4.9. Note that the central node, which has more neighbor connections, was not selected as one of the best nodes to compute. Table 4.6 shows that only 6 nodes will compute the all needed cross-correlations. We can conclude that it is possible to apply our communication-reduced method on nodes where variable bandwidth is present.

4.4.5 Bandwidth and Energy Analysis.

To further analyze the results of the proposed method, we compare the original communication pattern (Fig. 4.5) with the results of our experiments using the proposed communication-



Node Number	Computed Cross-correlation	Node Number	Computed Cross-correlation
3	C_{1-3}	8	C_{7-8}
	C_{1-4}		C_{7-11}
	C_{3-4}		C_{8-11}
	C_{3-7}		C_{8-12}
	C_{4-7}		C_{11-12}
4	C_{4-5}	10	C_{6-7}
	C_{4-8}		C_{6-9}
	C_{5-8}		C_{6-10}
6	C_{2-3}		C_{7-10}
	C_{2-6}		C_{9-10}
	C_{3-6}	13	C_{10-11}
			C_{10-13}
			C_{11-13}

Figure 4.9 & Table 4.6: Communication pattern using the proposed model if we consider variable bandwidth (blue nodes are nodes that computes cross-correlation; red nodes just send data to the corresponding network.) & Nodes that compute cross-correlations.

reduced method. We analyze three main aspects: (i) throughput of the network; (ii) computational cost in terms of the number of computed cross-correlations; and (iii) percentage of energy saving in the network after applying the proposed method.

We measure the throughput on every sensor node based on the number of packets received to verify if our method improves bandwidth utilization. Fig. 4.10 shows for each node the

number of kilobytes per second (kbps) transmitted. Note that the original communication pattern (without optimization method) utilizes much more bandwidth that our method with or without bandwidth restriction. Only the unlimited bandwidth experiment have 2 nodes that utilize similar bandwidth than the original pattern. Fig. 4.11 show the total throughput of the network in all cases. Our method significantly improves the communication cost. Between our experiments, experiment 1 (assuming unlimited bandwidth) reduce communication the most. However, in reality, we do not have unlimited bandwidth, and the option of our four experiments (variable bandwidth that depends on the number of links/hops) results more attractive.

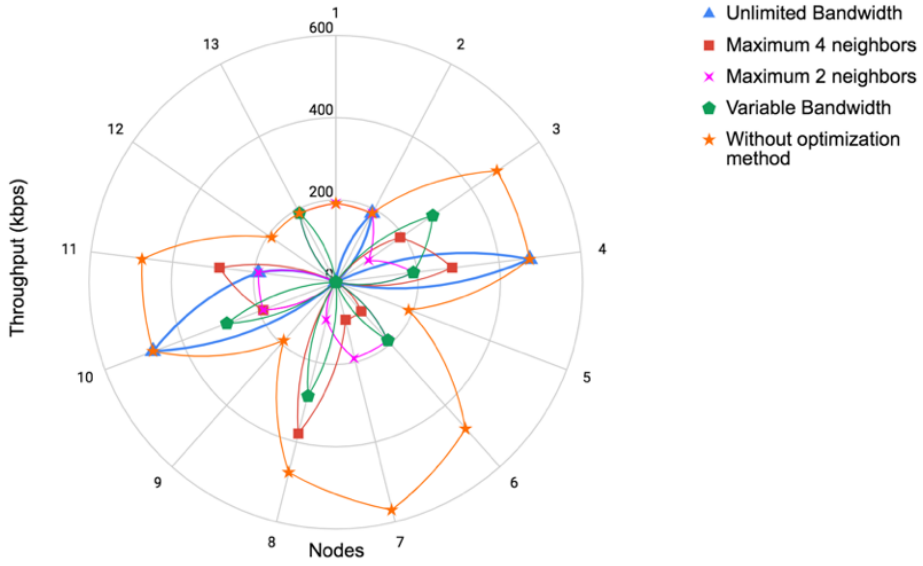


Figure 4.10: Throughput comparison of every sensor node among different available bandwidth using our proposed communication-reduced method vs communication pattern without communication-reduced method. Note that our method improves bandwidth utilization in all the tested scenarios.

The improvement in the communication cost is more than expected in this case; because of that, we analyze how our method impacts computation and energy saving. Fig. 4.12 shows the computation cost of each experiment and the original communication pattern. Note that

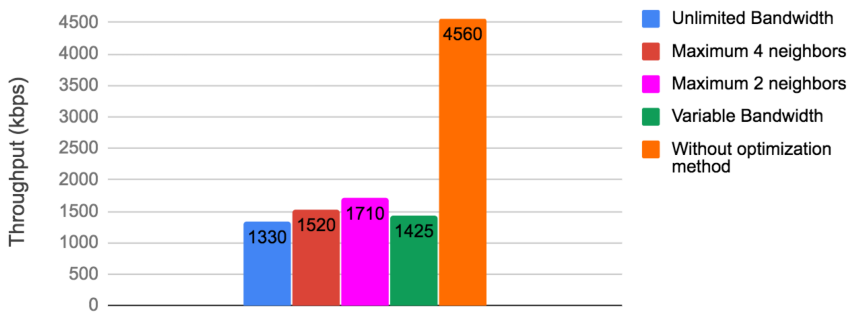


Figure 4.11: Total throughput comparison node among different available bandwidth using our proposed communication-reduced method vs communication pattern without communication-reduced method. Note that in the whole system level our method meets bandwidth constraints.

our method besides reduces communication cost also reduces computational cost. It is because the cross-correlations are computed only once and at the "best nodes". In this case, the experiment using a maximum of 2 neighbors is the one that balances better the computation cost; however, the difference is not too significant respecting the other experiments that also use our method. Finally, we compute the percentage of energy saving when our method is applied. For doing this, our comparison is made respecting the original communication pattern. Fig. 4.13 illustrates energy saving results. According to [29], the energy of transmitting 1KB a distance of 100m is approximately the same that executing 3 million of instruction by the processor. Hence, local data processing and reducing communication cost is crucial for saving sensors energy. Note that, in terms of communication, in all our experiments, we save at least more than 60% of energy. This is a very promising reduction. Furthermore, note that, in terms of computation, the energy saving in our experiments is around 50%. These results imply that our approach besides reducing communication cost, it also helps to avoid extra energy utilization.

Based on our evaluation, we remark that when we have a variable bandwidth that mainly relies on the number of links/hops of the nodes, our method performs very well in terms on

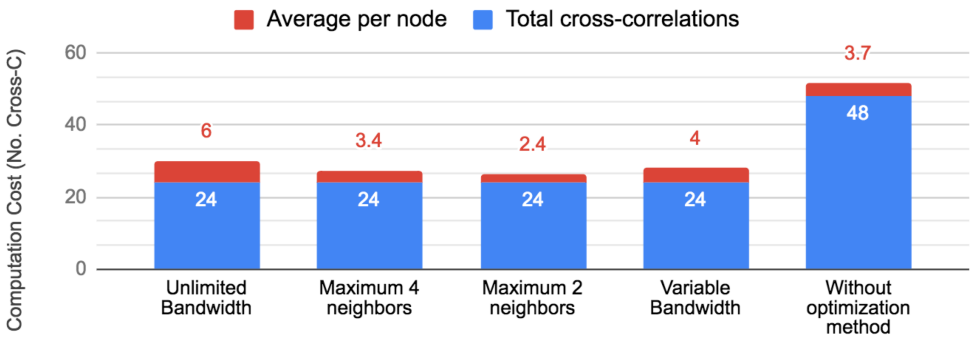


Figure 4.12: Computational cost in terms of number of cross-correlations computed by nodes. Comparison between our proposed communication-reduced method and communication pattern without reduction.Note that in the case "unlimited bandwidth", only 4/13 nodes are computing cross-correlations, and in the case "without optimization method", all 13/13 nodes are computing.

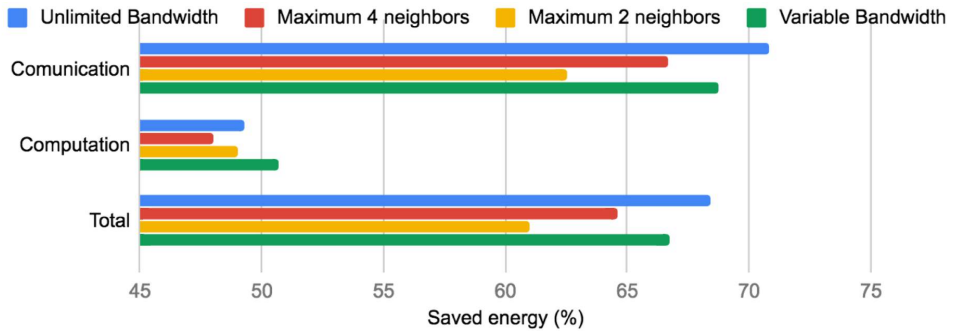


Figure 4.13: Percentages of energy saving respecting to the communication pattern without using out method.

communication reduction, computation reduction, and energy saving.

4.5 Conclusion

In this chapter, we have presented a communication-reduced method for cross-correlation of ambient noise data for subsurface imaging using distributed spatial auto-correlation (dSPAC). The main idea is to reduce the communication cost between nodes when they are working

together to correlate data. The subsurface methodology includes two main sections: cross-correlation section and subsurface imaging section. The main contribution of this methodology is based on the cross-correlation section. We have shown that is possible to select "best nodes" to correlate the needed data for imaging, and at the same time, meet bandwidth constraints. We also present analysis regarding computational cost and energy consumption of the nodes. The potential scientific and social impact of our method is significantly and broadly widespread.

4.6 Publications

The work presented in this Chapter has led to the following publications

- Valero, M., Li, F., Clemente, J., & Song, W. (2019). **Distributed and Communication-Efficient Spatial Auto-Correlation Subsurface Imaging in Sensor Networks.** Sensors, 19(11), 2427. DOI: 10.3390/s19112427

Chapter 5

Conclusion and Future Work

Ambient noise imaging is a technique widely used in geophysical exploration for investigating subsurface structures using recorded background raw ambient noise data. The current state of the art of ambient noise monitoring relies on gathering these high volumes of raw data back to a centralized server or base station to post-processing. The transformative integration of sensor networks and geophysical imaging techniques enables the creation of a system to monitor and analyze seismic data in real time as well as the generation of images of various subsurface structures, properties, and dynamics. This dissertation aims to present an integrated approach for subsurface ambient noise imaging in networks that investigates the capacity of in-situ and cooperative computing in the generation reliable underground velocity maps. We also investigated and proposed a communication-reduced method to meet bandwidth and energy constraints in the network. Using the proposed framework, we can generate subsurface images without the need for transmitting all data to a central place, making the results resilient and fault tolerant.

In Chapter 2, a distributed eikonal tomography methodology was presented. It was concluded that the distributed technique was able to generate underground velocity maps in an emulator and real scenarios. The comparison against a centralized method demonstrated

that our approach reduces the communication cost and meet bandwidth limitation since sensor nodes only communicate prepared and reduced data. We reached more than 66% of reduction in communication, and error compared with the centralized approach is less than 15%. We also showed that aggregating exchanged data using a "consensus" algorithm, the system is more resilient than using a tree-aggregation approach; however, in the consensus approach, the communication cost is directly proportional to the number of neighbor nodes. More neighbor nodes imply more communication. The main drawback of this proposed methodology is the effect of the topology. In traditional ambient noise imaging, the cross-correlation of the signals is performed between all pair of sensors. We proposed cross-correlation with only neighbor nodes to reduce communication and computation complexity. In strongly connected topologies, the subsurface image will be almost the same as computing all-to-all cross-correlations. In sparse topologies, when we are only able to observe a subset of entries, the resolution may be reduced. However, when the underlying true map varies smoothly, the quality of the recovered image using partial data will no degrade much from that recovered using full data.

In Chapter 3, we introduce dSPAC (distributed spatial autocorrelation) in sensor networks for computing subsurface imaging in a local area only with neighbors and using an interpolation technique for larger areas. dSPAC does not require all-to-all cross-correlations, and because high frequencies can be used, shallow subsurface infrastructures can be imaged. We showed that this technique can be used to illuminated some underground structures like pipelines and variation on the underground velocities. For example, we used the technique to clearly image temporal evolution of the water leakage process. We showed that this technique uses less bandwidth and communication that the centralized approach, which make it suitable for in-situ computing. With this approach, we presented the first autonomous, collaborative and non-invasive system for subsurface structure detection that can be applied in numerous scenarios.

In Chapter 4, the way to improving communication for our approach was investigated. A new communication-reduced method based on a combinatorial optimization was proposed. The optimization problem is solved to get the best nodes for computing cross-correlation and avoid unneeded transmissions. We showed that the method also has an impact on reducing the energy consumption of the sensor nodes. The experiments revealed that our method reduced by more than half the communication cost and it can be adjusted to different available bandwidth. The computation cost of the units is also diminished in more than 70%. These reductions also have a beneficial impact on the energy consumption of the sensors. In our experiments, sensor nodes reached up to 60% of saved-energy with the proposed optimization method. The potential scientific and social impact of our method is significantly and broadly widespread.

According to the observations from this dissertation, the following hypotheses will be investigated:

- Can the dSPAC method be utilized for watering system monitoring in agriculture?
- Can we combine the imaged infrastructures (example pipelines) and the velocity drop to further infer the leakage locations?
- Can we use mathematical approximation techniques to approximate distributed eikonal tomography results from partial observations?

Bibliography

[1] M. Valero, G. Kamath, J. Clemente, F.-C. Lin, Y. Xie, and W. Song, "Real-time ambient noise subsurface imaging in distributed sensor networks," in *The 3rd IEEE International Conference on Smart Computing (SMARTCOMP 2017)*. IEEE, 2017, pp. 1–8. [Online]. Available: http://dx.doi.org/10.1109/SMARTCOMP.2017.7947040

[2] V. C. Tsai, "On establishing the accuracy of noise tomography travel-time measurements in a realistic medium," *Geophysical Journal International*, vol. 178, no. 3, pp. 1555–1564, 2009.

[3] C. Li and R. D. Van Der Hilst, "Structure of the upper mantle and transition zone beneath southeast asia from traveltime tomography," *Journal of Geophysical Research: Solid Earth*, vol. 115, no. B7, 2010.

[4] F.-C. Lin, M. P. Moschetti, and M. H. Ritzwoller, "Surface wave tomography of the western united states from ambient seismic noise: Rayleigh and love wave phase velocity maps," *Geophysical Journal International*, vol. 173, no. 1, pp. 281–298, 2008.

[5] M. P. Moschetti, M. H. Ritzwoller, F. Lin, and Y. Yang, "Crustal shear wave velocity structure of the western United States inferred from ambient seismic noise and earthquake data," *Journal of Geophysical Research*, vol. 115, no. B10, Oct. 2010. [Online]. Available: http://dx.doi.org/10.1029/2010JB007448

[6] 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, 2009.

[7] G. Ekström, G. A. Abers, and S. C. Webb, "Determination of surface-wave phase velocities across USArray from noise and Aki's spectral formulation," *Geophysical Research Letters*, vol. 36, no. 18, 2009.

[8] L. Fang, J. Wu, Z. Ding, and G. F. Panza, "High resolution rayleigh wave group velocity tomography in North China from ambient seismic noise," *Geophysical Journal International*, vol. 181, no. 2, pp. 1171–1182, 2010.

[9] P. Arroucau, N. Rawlinson, and M. Sambridge, "New insight into cainozoic sedimentary basins and palaeozoic suture zones in southeast Australia from ambient noise surface wave tomography," *Geophysical Research Letters*, vol. 37, no. 7, 2010.

[10] M. Valero, F. Li, S. Wang, F.-C. Lin, and W. Song, "Real-time cooperative analytics for ambient noise tomography in sensor networks," *IEEE Transactions on Signal and Information Processing over Networks*, 2018. [Online]. Available: http://sensorweb.engr.uga.edu/wp-content/uploads/2018/10/Dropbox_valero2018real.pdf

[11] T. Tawanda, "A node merging approach to the transhipment problem," *International Journal of System Assurance Engineering and Management*, vol. 8, no. 1, pp. 370–378, 2017.

[12] A. Villasenor, Y. Yang, M. H. Ritzwoller, and J. Gallart, "Ambient noise surface wave tomography of the Iberian Peninsula: Implications for shallow seismic structure," *Geophysical Research Letters*, vol. 34, no. 11, 2007.

[13] G. Kamath, L. Shi, W.-Z. Song, and J. M. Lees, "Distributed travel-time seismic tomography in Large-Scale sensor networks," *Journal of Parallel and Distributed Computing*, vol. 89, 2016.

[14] L. Zhao, W.-Z. Song, L. Shi, and X. Ye, "Decentralized seismic tomography computing in Cyber-Physical sensor systems," *Cyber-Physical Systems, Taylor and Francis*, 2015.

[15] G. Kamath, L. Shi, E. Chow, and W.-Z. Song, "Distributed tomography with adaptive mesh refinement in sensor networks," *International Journal of Sensor Network*, 2015.

[16] G. Kamath, W.-Z. Song, P. Ramanan, L. Shi, and J. Yang, "DRISTI: Distributed Real-Time In-Situ seismic tomographic imaging," in *14th International Conference on Ubiquitous Computing and Communications (IUCC)*, Liverpool, UK, 2015.

[17] L. Shi, W.-Z. Song, M. Xu, Q. Xiao, J. M. Lee, and G. Xing, "Imaging seismic tomography in sensor network," in *IEEE SECON*, 2013.

[18] A. L. Stork, C. Allmark, A. Curtis, J.-M. Kendall, and D. J. White, "Assessing the potential to use repeated ambient noise seismic tomography to detect CO 2 leaks: Application to the aquistore storage site," *International Journal of Greenhouse Gas Control*, vol. 71, pp. 20–35, 2018.

[19] L. Liu, Q.-f. Chen, W. Wang, and E. Rohrbach, "Ambient noise as the new source for urban engineering seismology and earthquake engineering: a case study from beijing metropolitan area," *Earthquake Science*, vol. 27, no. 1, pp. 89–100, 2014.

[20] N. M. Shapiro, M. Campillo, L. Stehly, and M. H. Ritzwoller, "High-resolution surface-wave tomography from ambient seismic noise," *Science*, vol. 307, no. 5715, pp. 1615–1618, 2005.

[21] G. A. Prieto, J. F. Lawrence, A. I. Chung, and M. D. Kohler, "Impulse response of civil structures from ambient noise analysis," *Bulletin of the Seismological Society of America*, vol. 100, no. 5A, pp. 2322–2328, 2010.

[22] N. Nakata and R. Snieder, "Monitoring a building using deconvolution interferometry. II: Ambient-vibration analysis," *Bulletin of the Seismological Society of America*, vol. 104, no. 1, pp. 204–213, 2013.

[23] H. Sun, A. Mordret, G. A. Prieto, M. N. Toksöz, and O. Büyüköztürk, "Bayesian characterization of buildings using seismic interferometry on ambient vibrations," *Mechanical Systems and Signal Processing*, vol. 85, pp. 468–486, 2017.

[24] A. Mordret, H. Sun, G. A. Prieto, M. N. Toksöz, and O. Büyüköztürk, "Continuous monitoring of High-Rise buildings using seismic interferometry," *Bulletin of the Seismological Society of America*, vol. 107, no. 6, pp. 2759–2773, 2017.

[25] G. Olivier, F. Brenguier, M. Campillo, R. Lynch, and P. Roux, "Body-wave reconstruction from ambient seismic noise correlations in an underground mine," *Geophysics*, vol. 80, no. 3, pp. KS11–KS25, 2015.

[26] F. De Paola, E. Galdiero, M. Giugni, R. Papa, and G. Urciuoli, "Experimental investigation on a buried leaking pipe," *Procedia Engineering*, vol. 89, pp. 298–303, 2014.

[27] T. S. T. Amran, M. P. Ismail, M. R. Ahmad, M. S. M. Amin, S. Sani, N. A. Masenwat, M. A. Ismail, and S.-H. A. Hamid, "Detection of underground water distribution piping system and leakages using ground penetrating radar (gpr)," in *AIP Conference Proceedings*, vol. 1799, no. 1. AIP Publishing, 2017, p. 030004.

[28] A. De Coster, J. P. Medina, M. Nottebaere, K. Alkhalifeh, X. Neyt, J. Vanderdonckt, and S. Lambot, "Towards an improvement of gpr-based detection of pipes and leaks

in water distribution networks," *Journal of Applied Geophysics*, vol. 162, pp. 138–151, 2019.

[29] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, 2000.

[30] S. Savazzi, U. Spagnolini, L. Goratti, D. Molteni, M. Latva-aho, and M. Nicoli, "Ultra-wide band sensor networks in oil and gas explorations," *IEEE Communications Magazine*, vol. 51, no. 4, pp. 150–160, 2013.

[31] S. Conceição, F. Ribeiro, R. Campos, and M. Ricardo, "A ns-3 based simulator of tcp/ip wireless underground networks," in *2014 IFIP Wireless Days (WD)*. IEEE, 2014, pp. 1–6.

[32] S. Conceição, C. Pendão, A. Moreira, and M. Ricardo, "Evaluation of medium access and a positioning system in wireless underground sensor networks," in *2016 Wireless Days (WD)*. IEEE, 2016, pp. 1–6.

[33] H. T. H. Trang, S.-G. Choi, S. O. Hwang *et al.*, "Impact of soil medium on the path connectivity of sensors in wireless underground sensor networks," in *2016 International Conference on Advanced Technologies for Communications (ATC)*. IEEE, 2016, pp. 60–64.

[34] G. Liu, Z. Wang, and T. Jiang, "Qos-aware throughput maximization in wireless powered underground sensor networks," *IEEE Transactions on Communications*, vol. 64, no. 11, pp. 4776–4789, 2016.

[35] L. Xie, Y. Xie, S.-M. Wu, F.-C. Lin, and W. Song, "Communication efficient signal detection for distributed ambient noise imaging," in *The 52nd Asilomar Conference on Signals, Systems and Computers*, 2018.

[36] F. Brenguier, N. M. Shapiro, M. Campillo, V. Ferrazzini, Z. Duputel, O. Coutant, and A. Nercessian, "Towards forecasting volcanic eruptions using seismic noise," *Nature Geoscience*, vol. 1, no. 2, pp. 126–130, Jan. 2008. [Online]. Available: http://dx.doi.org/10.1038/ngeo104

[37] F. Brenguier, M. Campillo, C. Hadziioannou, N. M. Shapiro, R. M. Nadeau, and E. Larose, "Postseismic Relaxation Along the San Andreas Fault at Parkfield from Continuous Seismological Observations," *Science*, vol. 321, no. September, pp. 1478–1481, 2008.

[38] Z. Duputel, V. Ferrazzini, F. Brenguier, N. M. Shapiro, M. Campillo, and A. Nercessian, "Real time monitoring of relative velocity changes using ambient seismic noise at the Piton de la Fournaise volcano (La Réunion) from January 2006 to June 2007," *Journal of Volcanology and Geothermal Research*, vol. 184, no. 1-2, pp. 164–173, Jul. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.jvolgeores.2008.11.024

[39] 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.

[40] 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

[41] M. P. Barmin, M. H. Ritzwoller, and A. L. Levshin, "A Fast and Reliable Method for Surface Wave Tomography," *Pure and Applied Geophysics*, vol. 158, pp. 1351–1375, 2001.

[42] H. Nicolson, A. Curtis, B. Baptie, and E. Galetti, "Seismic interferometry and ambient noise tomography in the british isles," *Proceedings of the Geologists' Association*, vol. 123, no. 1, pp. 74–86, 2012.

[43] G. D. Bensen, M. H. Ritzwoller, M. P. Barmin, A. L. Levshin, F. Lin, M. P. Moschetti, N. M. Shapiro, and Y. Yang, "Processing seismic ambient noise data to obtain reliable broad-band surface wave dispersion measurements," *Geophysical Journal International*, vol. 169, no. 3, pp. 1239–1260, 2007.

[44] M. H. Ritzwoller, F.-C. Lin, and W. Shen, "Ambient noise tomography with a large seismic array," *Comptes Rendus Geoscience*, vol. 343, no. 8, pp. 558–570, 2011.

[45] V. Madisetti, *The digital signal processing handbook*. CRC press, 1997.

[46] R. Snieder, "Extracting the green's function from the correlation of coda waves: A derivation based on stationary phase," *Physical Review E*, vol. 69, no. 4, p. 046610, 2004.

[47] S. Grion and A. Mazzotti, "Stacking weights determination by means of SVD and cross-correlation," in *SEG Technical Program Expanded Abstracts 1998*. Society of Exploration Geophysicists, 1998, pp. 1135–1138.

[48] J. Rhie and B. Romanowicz, "Excitation of earth's continuous free oscillations by atmosphere–ocean–seafloor coupling," *Nature*, vol. 431, no. 7008, p. 552, 2004.

[49] L. Stehly, M. Campillo, and N. M. Shapiro, "A study of the seismic noise from its long-range correlation properties," *Journal of Geophysical Research: Solid Earth*, vol. 111, no. B10, 2006.

[50] Y. Yang and M. H. Ritzwoller, "Characteristics of ambient seismic noise as a source for surface wave tomography," *Geochemistry, Geophysics, Geosystems*, vol. 9, no. 2, 2008.

[51] N. Kimura and S. Latifi, "A survey on data compression in wireless sensor networks," in *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*, vol. 2. IEEE, 2005, pp. 8–13.

[52] S.-M. Wu, K. M. Ward, J. Farrell, F.-C. Lin, M. Karplus, and R. B. Smith, "Anatomy of old faithful from subsurface seismic imaging of the yellowstone upper geyser basin," *Geophysical Research Letters*, vol. 44, no. 20, 2017.

[53] K. J. Seats, J. F. Lawrence, and G. A. Prieto, "Improved ambient noise correlation functions using welch's method," *Geophysical Journal International*, vol. 188, no. 2, pp. 513–523, 2012.

[54] W. H. F. Smith and P. Wessel, "Gridding with continuous curvature splines in tension," *Geophysics*, vol. 55, no. 3, pp. 293–305, 1990.

[55] L. Zhao, W.-Z. Song, X. Ye, and Y. Gu, "Asynchronous broadcast-based decentralized learning in sensor networks," *International Journal of Parallel, Emergent and Distributed Systems*, pp. 1–19, 2017. [Online]. Available: http://dx.doi.org/10.1080/17445760.2017.1294690

[56] L. Zhao and W. Song, "Decentralized consensus in distributed networks," *International Journal of Parallel, Emergent and Distributed Systems*, 2016. [Online]. Available: http://dx.doi.org/10.1080/17445760.2016.1233552

[57] L. Zhao, W.-Z. Song, and X. Ye, "Fast decentralized gradient descent method and applications to in-situ seismic tomography," in *IEEE International Conference on Big Data (IEEE BigData 2015)*, 2015.

[58] F.-C. Lin, D. Li, R. W. Clayton, and D. Hollis, "High-resolution 3D shallow crustal structure in long beach, california: Application of ambient noise tomography on a dense seismic array," *Geophysics*, vol. 78, no. 4, pp. Q45–Q56, 2013.

[59] D. F. Sumy, R. Woodward, M. Barklage, D. Hollis, N. Spriggs, J. M. Gridley, and T. Parker, "Sweetwater, texas large n experiment," in *AGU Fall Meeting Abstracts*, 2015.

[60] J. Ahrenholz, T. Goff, and B. Adamson, "Integration of the CORE and EMANE network emulators," pp. 1870–1875, 2011.

[61] Wickert, "Sampling and aliasing," 2011. [Online]. Available: http://www.eas.uccs.edu/~mwickert/ece2610/lecture_notes/ece2610_chap4.pdf

[62] C. Peraki and S. D. Servetto, "On the maximum stable throughput problem in random networks with directional antennas," in *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*. ACM, 2003, pp. 76–87.

[63] P. Chen and E.-J. Lee, *Full-3D Seismic Waveform Inversion: Theory, Software and Practice*. Springer, 2015.

[64] K. Aki, "Space and time spectra of stationary stochastic waves, with special reference to microtremors," *Bull. Earth. Res. Inst.*, vol. 35, pp. 415–456, 1957.

[65] I. Cho, T. Tada, and Y. Shinozaki, "A new method to determine phase velocities of rayleigh waves from microseisms," *Geophysics*, vol. 69, no. 6, pp. 1535–1551, 2004.

[66] M. W. Asten, "On bias and noise in passive seismic data from finite circular array data processed using SPAC methodsBias and noise in passive seismic data," *Geophysics*, vol. 71, no. 6, pp. V153–V162, 2006.

[67] I. Cho, T. Tada, and Y. Shinozaki, "Centerless circular array method: Inferring phase velocities of rayleigh waves in broad wavelength ranges using microtremor records," *Journal of Geophysical Research: Solid Earth*, vol. 111, no. B9, 2006.

[68] V. C. Tsai and M. P. Moschetti, "An explicit relationship between time-domain noise correlation and spatial autocorrelation (SPAC) results," *Geophysical Journal International*, vol. 182, no. 1, pp. 454–460, 2010.

[69] M. J. Jacobson, "Space-Time correlation in spherical and circular noise fields," *The Journal of the Acoustical Society of America*, vol. 34, no. 7, pp. 971–978, 1962.

[70] M. Jain and C. Dovrolis, "End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput," *IEEE/ACM Transactions on Networking (TON)*, vol. 11, no. 4, pp. 537–549, 2003.

[71] E. Upton, "Raspberry pi 3," *URL https://www.raspberrypi.org/products/raspberry-pi-3-model-b*, 2016.

[72] S. Demirci, E. Yigit, I. H. Eskidemir, and C. Ozdemir, "Ground penetrating radar imaging of water leaks from buried pipes based on back-projection method," *Ndt & E International*, vol. 47, pp. 35–42, 2012.

[73] A. S. Alamry, M. van der Meijde, M. Noomen, E. A. Addink, R. van Benthem, and S. M. de Jong, "Spatial and temporal monitoring of soil moisture using surface electrical resistivity tomography in mediterranean soils," *Catena*, vol. 157, pp. 388–396, 2017.

[74] T. Jouen, R. Clément, H. Henine, C. Chaumont, B. Vincent, and J. Tournebize, "Evaluation and localization of an artificial drainage network by 3d time-lapse electrical resistivity tomography," *Environmental Science and Pollution Research*, vol. 25, no. 24, pp. 23 502–23 514, 2018.

[75] Z. He, Z. Hu, Y. Gao, L. He, C. Meng, and L. Yang, "Field test of monitoring gas reservoir development using time-lapse continuous electromagnetic profile method," *Geophysics*, vol. 80, no. 2, pp. WA127–WA134, 2015.

[76] T. Ozdemir, S. Roy, and R. S. Berkowitz, "Imaging of a shallow subsurface objects: an experimental investigation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 30, no. 3, pp. 472–481, 1992.

[77] Y. Zhang, Y. E. Li, H. Zhang, and T. Ku, "Optimized passive seismic interferometry for bedrock detection: A singapore case study," in *SEG Technical Program Expanded Abstracts 2018*. Society of Exploration Geophysicists, 2018, pp. 2506–2510.

[78] W. Song, F. Li, M. Valero, and L. Zhao, "Toward creating a subsurface camera," *Sensors*, vol. 19, no. 2, 2019. [Online]. Available: http://www.mdpi.com/1424-8220/19/2/301

[79] R. Bachrach and A. Nur, "High-resolution shallow-seismic experiments in sand, part i: Water table, fluid flow, and saturation," *Geophysics*, vol. 63, no. 4, pp. 1225–1233, 1998.

[80] C. Geiss, M. Jilge, T. Lakes, and H. Taubenböck, "Estimation of seismic vulnerability levels of urban structures with multisensor remote sensing," *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 9, no. 5, pp. 1913–1936, 2016.

[81] N. Nakata, R. Snieder, T. Tsuji, K. Larner, and T. Matsuoka, "Shear wave imaging from traffic noise using seismic interferometry by cross-coherenceShear wave imaging from traffic noise," *Geophysics*, vol. 76, no. 6, pp. SA97–SA106, 2011.

[82] H. Nakahara, "Formulation of the spatial autocorrelation (spac) method in dissipative media," *Geophysical Journal International*, vol. 190, no. 3, pp. 1777–1783, 2012.

[83] J. Xia, R. D. Miller, C. B. Park, and G. Tian, "Inversion of high frequency surface waves with fundamental and higher modes," *Journal of Applied Geophysics*, vol. 52, no. 1, pp. 45–57, 2003.

[84] Y.-Y. Song, J. P. Castagna, R. A. Black, and R. W. Knapp, "Sensitivity of near-surface shear-wave velocity determination from rayleigh and love waves," in *SEG Technical*

Program Expanded Abstracts 1989. Society of Exploration Geophysicists, 1989, pp. 509–512.

[85] C. Weemstra, L. Boschi, A. Goertz, and B. Artman, "Seismic attenuation from recordings of ambient noiseAttenuation from ambient noise," *Geophysics*, vol. 78, no. 1, pp. Q1–Q14, 2013.

[86] M. Valero, F. Li, X. Li, and W. Song, "Imaging subsurface civil infrastructure with smart seismic network," in *37th IEEE International Performance Computing and Communications Conference (IPCCC) 2018*, 2018.

[87] M. W. Asten, "Passive seismic methods using the microtremor wave field," *ASEG Extended Abstracts*, vol. 2004, no. 1, pp. 1–4, 2004.

[88] W. B. Banerdt, S. Smrekar, P. Lognonné, T. Spohn, S. W. Asmar, D. Banfield, L. Boschi, U. Christensen, V. Dehant, W. Folkner, and Others, "InSight: a discovery mission to explore the interior of mars," in *Lunar and Planetary Science Conference*, vol. 44, 2013, p. 1915.

[89] E. Larose, A. Khan, Y. Nakamura, and M. Campillo, "Lunar subsurface investigated from correlation of seismic noise," *Geophysical Research Letters*, vol. 32, no. 16, 2005.

[90] A. Mordret, P. Roux, P. Boué, and Y. Ben-Zion, "Shallow three-dimensional structure of the san jacinto fault zone revealed from ambient noise imaging with a dense seismic array," *Geophysical Journal International*, vol. 216, no. 2, pp. 896–905, 2018.

[91] W.-Z. Song, R. Huang, M. Xu, B. A. Shirazi, and R. LaHusen, "Design and deployment of sensor network for Real-Time High-Fidelity volcano monitoring," *IEEE Transaction on Parallel and Distributed Systems*, vol. 21, no. 11, pp. 1658–1674, 2010. [Online]. Available: http://dx.doi.org/10.1109/TPDS.2010.37

[92] 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)*. ACM, 2009, pp. 305–318.

[93] L. Shi, W.-Z. Song, M. Xu, Q. Xiao, J. M. Lees, and G. Xing, "Imaging volcano seismic tomography in sensor networks," in *The 10th Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (IEEE SECON)*, 2013.

[94] F. Li, Y. Qin, and W. Song, "Waveform inversion-assisted distributed reverse time migration for microseismic location," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 4, pp. 1327–1332, April 2019.

[95] H. Yao, R. D. van Der Hilst, and V. Maarten, "Surface-wave array tomography in SE Tibet from ambient seismic noise and two-station analysis–I. Phase velocity maps," *Geophysical Journal International*, vol. 166, no. 2, pp. 732–744, 2006.