Imaging Subsurface Civil Infrastructure with Smart Seismic Network

Maria Valero, Fangyu Li, WenZhan Song
Center for Cyber-Physical Systems
University of Georgia
Email: {maria.valero, fangyu.li, wsong}@uga.edu

XiangYang Li
School of Computer Science and Technology
University of Science and Technology China
Email: xiangyangli@ustc.edu.cn

Abstract—The ability to use networked seismic instruments to image subsurface civil infrastructure and activities in data-limited extreme environments is crucial for many civil and security applications. This paper presents a non-invasive smart seismic network for imaging subsurface structures using ambient noise only. An innovative in-network spatial auto-autocorrelation method is designed to image different layers of underground infrastructures at different frequency bands and finally result in a 3D image. The proposed approach is general and can characterize the near-surface sediment and the infrastructure simultaneously. The experiments demonstrate that underground utility lines affecting sediment can be imaged, and the potential of using the method for abnormal activities like leakages. An exhaustive evaluation regarding bandwidth utilization, communication cost, and system resilience were conducted to highlight the benefits of the proposed approach.

Index terms—seismic network, subsurface structures, ambient noise, spatial auto-correlation.

I. INTRODUCTION

Wireless sensor networks have the potential to greatly enhance the understanding of subsurface structures by permitting large distributed deployments of sensor nodes in difficult-to-reach or hazardous areas [1]. Wireless networks allow sensor nodes to communicate with each other by using a self-healing mesh network allowing fast problem solving, minimal user intervention, and real-time data processing. However, in data-limited environments, such as volcanoes, deserts, poles, and even other planets, the need of an autonomous sensor system to run continuously with almost zero-maintenance for a long period is a high priority.

In this paper, we present a system for collaborative subsurface infrastructure detection on top of a wireless seismic network. Instead to use earthquakes information as traditional travel-time tomography [2], 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 [3]. We use spatial auto-correlation (SPAC) [4] as the base of our ambient noise tomography. 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 exploration geophysics, 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.

The main innovation of this paper can be summarized as follows: (i) 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. 1; (ii) in-network tomography computing techniques that distribute the tomographic computing burdens to each node while performing real-time seismic image computation; (iii) seismic sensor geometry and techniques (SPAC) suitable for areas where is needed to study high frequency ranges for detecting shallow subsurface structures; (iv) autonomous system that can be deployed in harsh environment and can deliver results with minimal user intervention and almost zero-maintenance; (v) self-healing system based on a mesh network with fault-tolerance; and (v) an approach that can potentially be used for future extraterritorial experiments such as imaging planetary subsurface structures and activities using ambient noise.

![Wireless Mesh Network](image)

Fig. 1: Example of mesh network for subsurface imaging. The used geometry allows many rings in the network, each one with a center station.
The rest of the paper is organized as follows: section II presents the principles and algorithm design. Section III introduces the system implementation. We explain the system set up in Section IV. Section V presents the results of our experiments. We evaluate the system in terms of bandwidth and communication in Section VI. We discuss potential works of the system in Section VII. Finally, we conclude the paper in section VIII.

II. PRINCIPLES AND ALGORITHM DESIGN

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.

A. 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 [5], 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 [6]. 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 [5]. The time-domain normalization we use is running-absolute-mean normalization [5]. 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

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

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 [5]. Fig. 2 shows an example of ambient noise preservation after data preparation.

![Fig. 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.](image)

B. 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:

\[
C_{x_1x_2}(t) \equiv \frac{1}{2T} \int_{-T}^{T} u(x_1, \tau)u(x_2, \tau + t)d\tau \quad (2)
\]

In this way, the similarities of the signals can be determined. Fig. 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(b) shows the spectrogram of the cross-correlation with a bandpass between 17-20Hz.

![Cross-correlation example between two sensor's data.](image)

**Fig. 3:** Cross-correlation example between two sensor's data. (a) Casual (positive) and anti-casual (negative) symmetric cross-correlation. (b) Frequency-time spectrogram where is possible to distinguish dominant frequencies.

**C. Spatial Autocorrelation**

SPAC method was first introduced by Akin [4] based on a statistical investigation of seismic waves. Modifications of the method have been presented overtime [7]–[9]. We chose SPAC because it 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 [8] 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. 4, the phase velocities can be calculated.

![Geometry of sensor nodes and an incident plane wave.](image)

**Fig. 4:** Geometry of sensor nodes and an incident plane wave. Red starts represent sensors.

If microtremors are observed, the complex coherencies \( \text{COH} \) between a central and a circumferential receiver can be defined as:

\[
\text{COH}(r, \omega, \theta, \phi) = \exp\{irk\cos(\omega - \phi)\}, \quad (3)
\]

where \( i \) is the imaginary number, \( \omega \) is the angular frequency, \( k \) is the wavenumber, \( \theta \) is the azimuthal angle and \( \phi \) is the azimuth propagation of a single plane wave across the array.

The SPAC coefficients, also called azimuthal average, is defined then by:

\[
\rho(r, \omega) = \frac{1}{2\pi} \int_0^{2\pi} \exp\{irk\cos(\theta - \phi)\}d\theta = J_0 \left[ \frac{\omega}{c(\omega)} \right], \quad (4)
\]

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 \( \cos(\omega - \phi) \) symmetry in Eq. 4, we can switch \( \omega \) with \( \phi \) and obtain the same result. **This means that SPAC 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, \( \frac{1}{2\pi} \int_0^{2\pi} \exp\{irk\cos(\theta - \phi)\}d\theta \) is equivalent to \( \frac{1}{2\pi} \sum(C_{x_i,x_j}(t))\forall j \in N \), 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. Fig. 5 shows an example of calculated SPAC coefficients and the corresponding velocity estimation.

![Example of calculated SPAC coefficients and corresponding velocity estimation.](image)

**Fig. 5:** (a) SPAC coefficients. (b) Velocity obtained from SPAC.

**III. 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 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
The diagram flow of the complete methodology is shown in Fig. 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 II-A. 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 [10] and we achieve a compression rate of \( \sim 50\% \).

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.

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. 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 \( (\omega) \). 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 1. The other nodes in the ring just gather the data, prepare the data for transmission, compress and broadcast the data to sink sensors.

**Algorithm 1 Distributed SPAC-based Subsurface Imaging**

1. **Input**: Size of correlation-window \( t \)
2. **Input**: Total size of cross-correlation \( T \)
3. **Input**: Station location \( x_i \), ratio \( r \)
4. **While** not reach time \( T \)
5. Read prepared \( u(x_i, \tau) \) at position \( x_i \)
6. **for** every \( t \)
7. Receive \( u(x_j, \tau) \) for every \( j \) in \( i \) ring
8. Do \( C_{x_i,x_j} = \frac{1}{T} \int_0^T u(x_i, \tau)u(x_j, \tau + t) d\tau \) for each \( j \)
9. Stack cross-correlation \( C_{x_i,x_j} \) for each \( j \)
10. **end for**
11. **End While
12. Calculate SPAC coefficient \( \rho(\tau, \omega) \) for each \( \omega \)
13. Estimate velocities by fitting \( \rho(\tau, \omega) = J_0[\frac{\omega}{c(\omega)}\tau] \)
14. Broadcast \( \hat{c}_i \) vector that contains all \( c_i(\omega) \)
15. Receive \( \hat{c}_k \) from other \( k \) sink sensors
16. Interpolate \( \hat{c}_i \) and \( \hat{c}_k \) to get each depth layer and generate 3D velocity structure \( M \)
17. **Output**: 3D velocity structure \( (M) \)

**IV. System Setup**

In our deployments, we use a mesh network of sensor nodes that use a prototype instrument shown in Fig. 7(a). It has geophone, GPS (global positioning system), computing board, wireless radio, solar panel, and battery. Each sensor node is equipped with a wireless radio to self-form a sensor network for communication and data exchanges. The GPS provides precise timestamp and location information for each node. The computing board inside is Raspberry Pi 3 [11]. It has 1.2 GHz CPU, 1GB RAM and GPU for intensive local computing when needed, yet can be put in sleep for very low power consumption. The main components are inside a waterproof box for protecting them from the harsh environment. We show a test of the instrument running our algorithm during extreme
The instruments are placed in the field in a ring-based topology as shown in Fig. 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 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.

V. EXPERIMENTS

For testing our algorithm, we performed a first experiment using CORE network emulator [12]. We deployed data from Sweetwater, Texas [13] on each node in a suitable geometry area (Fig. 9(a)) and we calculated the phase velocities of the area with our proposed method as shown in Fig. 9(b). We compared the results with the knowing results gotten by the Eikonal Tomography method [14], [15]. 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 for 4Hz. With these results, we experimentally can say our method can be equivalent and complementary to the eikonal tomography method.

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. 8.

The system parameters we used for this experiment are shown in Table I.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Used Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of sensors</td>
<td>13</td>
</tr>
<tr>
<td>Number of sink sensor</td>
<td>7</td>
</tr>
<tr>
<td>Ratio ($r$)</td>
<td>Internal circle: 1.7 meters</td>
</tr>
<tr>
<td></td>
<td>External circle: 3 meters</td>
</tr>
<tr>
<td>Frequency range ($\omega$)</td>
<td>20Hz to 110Hz</td>
</tr>
<tr>
<td>Cross-correlation window ($t$)</td>
<td>5 minutes</td>
</tr>
<tr>
<td>Total time system running ($T$)</td>
<td>4 hours</td>
</tr>
</tbody>
</table>

TABLE I: Deployment parameters used in UGA experiment

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 [16] to invert the frequency ranges in depth. Fig 10 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. 10(a), show changes in velocity but the resolution is low.

The sink sensor constructs a 3D subsurface velocity image, as shown in Fig. 11, by interpolating the velocity profiles from all the nodes. In this figure, only depths between 1 and
Fig. 10: Velocity Maps. (a) Layer ∼ 5.1 meters depth. (b) Layer ∼ 3.5 meters depth. (c) Layer ∼ 1.5 meters depth. (d) Layer ∼ 0.8 meters depth. Sensor nodes locations are plotted as reference.

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).

Fig. 11: 3D velocity subsurface. Layers between 1 to 1.7 meters.

VI. 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-preprocessing on each node, all the computation is done in the central node.

A. Bandwidth Analysis

In a mesh network, every “hop” (link) between sensors will decrease the bandwidth by half [17]. 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) [11]. 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. 12: Throughput and bandwidth availability in (a) distributed approach, (b) centralized approach.

Fig. 12 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.
B. 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.

![Image](image_url)

Fig. 13: 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.

From Fig. 13(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 \( \sim 75\% \). This also has an impact in the energy consumption of each node. According to to [18], 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.

C. 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. 14 from a sink sensor perspective.

![Image](image_url)

Fig. 14: 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.

VII. 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 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 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” [19]. 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 [20], 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 [21]. We think, our system can be deployed in environment-resistant sensors.
that can use a sensor network to study underground properties in other planets.

VIII. CONCLUSION

In this paper, 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 envision the system can be used for studying subsurface structures in other extreme environments including extraterritorial bodies.

APPENDIX

EQUIVALENCE BETWEEN CROSS-CORRELATION AND SPAC

We extract the equivalence theory between cross-correlation and SPAC from [22], we summarize the key steps here:

It was observed by Jacobson [23] that for isotropic noise of a single frequency \( w \),

\[
C_{x_1x_2}(t,\omega) \equiv C_{x_1}(t) = J_0(\omega / c) \cdot \cos(\omega t).
\]  
(\text{A.1})

On the other hand, In [22] it was shown that the azimuth average for SPAC is

\[
\rho(r,\omega) = \frac{\Phi(r,\omega)}{\Phi(0,\omega)} = \frac{2\pi \Phi(w) J_0(\omega r / c)}{2\pi \Phi(w) \cdot 1} = J_0(\omega r / c),
\]  
(\text{A.2})

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

\[
\overline{C}_{x_1x_2}(t,\omega) = 2\pi \Phi(\omega) J_0(\omega r / c) \cos(\omega t),
\]  
(\text{A.3})

where \( \overline{C}_{x_1x_2} \) 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.

ACKNOWLEDGMENT


REFERENCES


