A Fog Computing Framework for Cognitive Portable Ground Penetrating Radars

Dalei Wu[1], Maxwell M. Omwenga[1], Yu Liang[1], Li Yang[1], Dryver Huston[2], Tian Xia[3]
[1] Department of Computer Science and Engineering, University of Tennessee at Chattanooga, USA
[2] Department of Mechanical Engineering, University of Vermont, USA
[3] Department of Electrical and Biomedical Engineering, University of Vermont, USA
Email: {dalei-wu, yu-liang, li-yang}@utc.edu; dgh179@mocs.utc.edu; {dryver.huston, txia}@uvm.edu

***Abstract*—With distributed communication, computation, and storage resources close to end users/devices, fog computing (FC) makes it very promising to develop cognitive portable ground penetrating radars (GPRs) operating intelligently and adaptively under varying sensing conditions. However both strict performance requirement and tradeoffs between communication and computation pose significant challenges. This paper presents a fog computing framework for cognitive portable GPRs. Specifically, the system architecture of an FC-enabled cognitive portable GPR is developed. Based on the identification of various involved computation tasks, an offloading policy was proposed to determine whether computation tasks should be executed locally or offloaded to the fog server. Experimental results show the efficacy of the proposed methods. The framework also provides insight into the design of cognitive Internet of things (IoT) supported by fog computing.**

***Index Terms*—Fog computing, offloading, ground penetrating radar, cognitive sensing**

I. Introduction

Fog computing is a computing paradigm that uses one or more collaborative end-user clients or near-user edge IoT devices to carry out a substantial amount of storage, communication, control, configuration, measurement and management. With distributed communication, computation, and storage resources and services on or close to devices and systems in the control of end-users, fog computing may enable real-time autonomous configurations and operations of those devices and systems. [1]–[3].

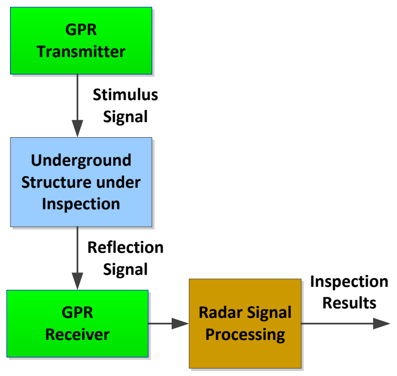
Portable GPRs, such as handheld or drone-borne GPRs, have been extensively used in many industrial applications, such as coal mining, structural health monitoring, subsurface utilities detection and localization [5], [6]. GPR is a non-destructive evaluation technique for effective assessment of subsurface conditions in large dielectric bodies, such as city streets, by launching and receiving electromagnetic (EM) waves from the same side of a structure. Location and nature of subsurface objects can be characterized by collecting and analyzing reflected and scattered waves [7], [8]. GPR-based subsurface survey is complicate as various sensing environment and subsurface targets have dissimilar features. Processing GPR data and extracting information of interest are challenging and involve a series of sophisticated steps. In nearly all existing GPR systems, the GPR data processing is performed off-line where the data are collected on field and stored, and then post-processed on a computer after the scanning. Such a processing approach does not adaptively adjust GPR operations in the survey.

To achieve optimum sensing performance, it is desired to design a GPR system that can operate adaptively under varying sensing conditions. For instance, to detect a shallowly buried object of small size, GPR radiating high-frequency EM waves can result in a fine sensing resolution; while to detect deep buried object, radiating lower-frequency EM waves have better ground penetrating capabilities. Based on such observations, cognitive GPRs have been proposed and investigated by dynamically tuning of GPR operational parameters to improve sensing performance [9].

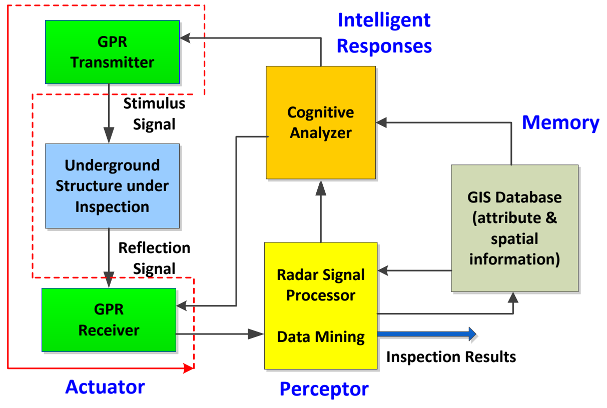
Fog computing provides ideal support for implementing and operating cognitive portable GPRs. The functions of GPR signal processing and intelligence generation require significant computation and storage capabilities, which could poses significant challenges to portable GPRs that have limited energy, computing, and storage resources. With fog computing, a promising solution is to offload some or all of the computation and storage tasks to a fog server. A cognitive GPR requires contiguous low-latency communications for real-time transmission of data and control feedback. The proximity of fog servers to end users/devices may satisfy such communication requirement. In contrast, traditional remote cloud computing services have difficulty providing uninterrupted services to cognitive portable GPRs due to the intermittent network connectivity and long communication latency.

Although fog computing makes it very promising to develop a cognitive portable GPR, there are still several significant research challenges that need to be addressed. Online intelligence generation requires continuous and real-time transmission and analysis of vast volumes of GPR data. A roadway GPR inspection can produce 100 or more gigabytes of data per hour. To reduce the amount of data to be transmitted from the GPR to the fog servers, some local data-processing functions could be performed at the transceiver side. However, the local computation time will increase the overall latency of the feedback loop. Therefore, the tradeoff between communication

This work was supported by the National Science Foundation under grant numbers 1647175 and 1663105.



(a) Traditional GPR



(b) Cognitive GPR

Fig. 1. The workflows of a traditional GPR (a) and a cognitive GPR (b).

and computation of the perception-action cycle needs to be studied by considering resource constraints and performance requirement. As the GPR moves, service might need to migrate from one fog server to another, posing challenges on continuous operation and control of GPRs.

There has been some research conducted on computation offloading for fog/edge computing [10]–[12]. In most of existing research on offloading, delay performance is evaluated either by using delay bound values or average delay values based on statistical information of the involved stochastic processes, such as the computation task arrival and the wireless channels, which may make them inapplicable in practical applications.

This paper is focused on the development of a fog computing framework that enables cognitive portable GPRs. First, a fog computing architecture for cognitive portable GPRs is developed and the functions of different modules are explained. Then, different computation tasks in a cognitive GPR are identified. A computation task offloading policy is designed to determine whether a computation task should be executed at the local GPR computer or at a remote fog server. To the best of our knowledge, this is the first work on fog computing for the development of cognitive portable GPRs.

II. The Proposed System Architecture

A conventional mode of GPR operation is that an expert sets the operational parameters into a proper configuration based on previous experience, which is an iterative time-consuming process, as shown in Figure 1(a). An alternative is to use a cognitive GPR where intelligence is generated on the fly to adaptively adjust the operational parameters based on data analysis and feedback control [9].

As shown in Figure 1(b), a cognitive GPR consists of an adaptive GPR transceiver, a perceptor module, a memory module, and a cognitive analyzer. The operation of the cognitive GPR follows a perception-action cycle: first, the GPR transceiver collects the reflected wave data about subsurface objects and sends them to the preceptor. Then, the preceptor processes and analyzes the data to extract signature patterns and format a perception of subsurface conditions. The memory module has a GIS database containing urban subsurface condition attributes and spatial locations. The cognitive analyzer carries out machine learning based on both the processing results from the perceptor and the prior knowledge about GPR measurement from the memory module to produce intelligent response for the control of radar transceiver reconfigurations. Once receiving the intelligent feedback from the cognitive analyzer, the adaptive GPR transceiver changes its operational parameters. During this process, collected GPR data can also be integrated with other data acquired by IoT devices such as positioning sensors.

Next, a system architecture for the proposed FC-enabled cognitive portable GPR is presented. As shown in Figure 2, the GPR mainly includes two parts: the front end and the back end. The front end is portable and includes a GPR transceiver for launching and receiving electromagnetic waves, a microcomputer for local computation, and a wireless access point for communicating with the fog server. The back end of the cognitive GPR, including the perceptor module, the memory module, and the cognitive analyzer, resides at the fog server.

The perception-action cycle of the FC-enabled cognitive GPR can be described as follows. The GPR transceiver collects the reflected wave data about subsurface objects. Based on the types of computation tasks, the delay performance requirement, and the resource constraints, a scheduler, running within the microcomputer at the GPR front end, decides whether each task should be performed locally or offloaded to the fog server. Following the decision, the microcomputer either offloads a task or executes the task locally. With the corresponding information from the GPR front end, the cognitive analyzer at the fog server generates control command for the GPR transceiver reconfigurations. The control command will be wirelessly sent back to the GPR front-end. As a result, the operational parameters of the GPR changes in a self-adaptive manner.

As the GPR moves across a field of interest, it may walk out of the coverage of a fog server. To ensure continuous sensing, multiple fog servers can be deployed in the field. As shown in Figure 3 the GPR approaches the coverage boundary of two neighboring fog servers, service migration [13] from one fog server to another can be carried out for the GPR. The service migration process can be coordinated by a controller. Due

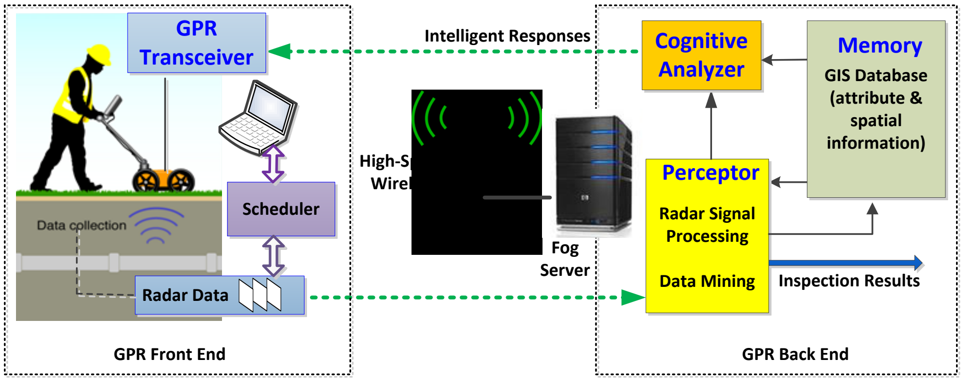


Fig. 2. The architecture of the proposed FC-enabled cognitive portable GPR.

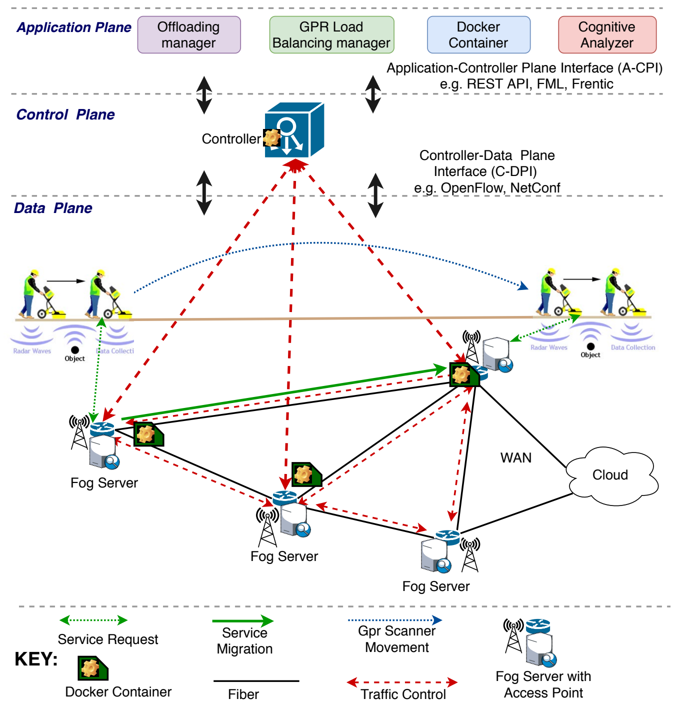


Fig. 3. Service migration from one fog server to another for continuous operation of a cognitive portable GPR.

to space limitation, further discussion on service migration is beyond the scope of this paper. In the following, the discussion is focused on the interaction between the GPR and one fog server.

III. Identification of Computation Tasks of Cognitive Portable GPRs

A. GPR Data Preprocessing

In real time cognitive sensing, the first type of computation task is background removal, noise filtering and clutter mitigation. It has been widely demonstrated that averaging and subtraction is effective in noise reduction [14]–[16]. The averaging operation is performed by stacking every a number of adjacent A-scan waveforms [8], which can reduce random noise and improve radargram signal-to-noise-ratio (SNR). For the subtraction calculation, an A-scan waveform obtained from the averaging calculation is selected as the reference and subtracted from all other averaging A-scan waveforms. Such calculations can effectively eliminate stationary background signals, antenna direct couplings, and mitigate the clutter resulting from ground surface reflection. The averaging and subtraction is a simple process and does not require intensive computing resource. Therefore in our proposed FC-enabled cognitive portable GPR system, this type of computation task can either be executed by the local computer at the GPR front end, or offloaded to the fog server, depending on the applied task scheduling policy.

B. Regions-Of-Interest (ROI) Detection

In radargram, the ROI data have dissimilar features from the background. By performing statistical analysis to evaluate data singularity, ROI data segments can be identified. Then by checking corresponding coordinates, the location and burying depth of ROI can be determined. In this study, Renyi entropy analysis [14], [17] is implemented to search for ROI.

For our GPR data processing, Renyi entropy characterization is developed to identify the singular region. In particular, a high Renyi entropy value indicates high degree of data similarity while a low entropy value highlights high degree of data singularity. Assume the received GPR reflection signal is $Y(t)$, it can be described as

$$Y(t) = D(t) + S(t). \tag{1}$$

where $D(t)$ represents the reflection signal from the object of interest; $S(t)$ models remaining interference and noise upon preprocessing. In calculation, power normalization is first performed with the summation of the power of the same time index data points on different traces. The normalization equation is expressed as

$$||Y_i(t)|| = \frac{|Y_i(t)|^2}{\sum_{i=1}^{m} |Y_i(t)|^2} \tag{2}$$

where $||Y_i(t)||$ is the normalized signal, i denotes the trace index and m is the total number of traces included; t specifies the time index of pulse data on each reflection trace waveform.

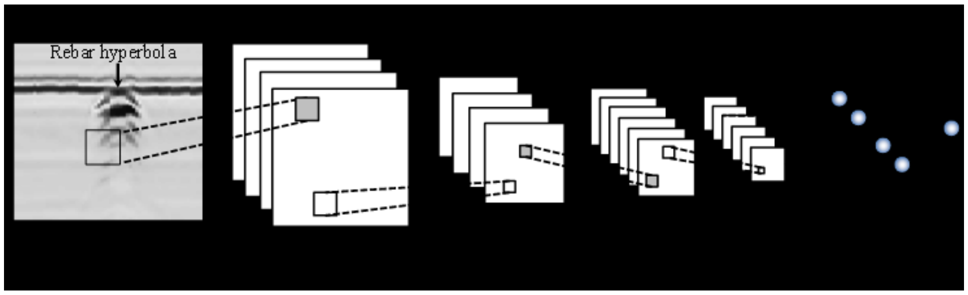


Fig. 4. DBNN processing GPR B-scan at multiple scales to detect subsurface infrastructure conditions. Each layer of feature detectors maps more complex relationships within the streaming GPR data.

Upon power normalization, a generalized Renyi's entropy is calculated to assess data singularity:

$$E_a(t) = \frac{1}{1-a}\log_e \sum_{i=1}^{m} ||Y_i(t)||^2 \tag{3}$$

where $E_a(t)$ is the entropy quantification, and a denotes the entropy order. Eq. (3) is equivalent to the basic Shannon entropy when a equals 1.

The entropy analysis is an intensive computation process that highly demands for computing power and CPU time. To leverage the calculation efficiency, in our FC-enabled cognitive GPR system, the entropy analysis is most likely implemented on the fog server instead of on the local computer.

C. DBNN Processing of GPR B-Scan Images

The cognitive analyzer combines the inspection data and prior knowledge about GPR measurement of structural features to produce intelligent responses to control radar transceiver reconfigurations. The cognitive analyzer can be implemented by applying machine learning to GPR B-scan images [8]. Deep Belief Neural Networks (DBNNs) [18] are a state-of-the-art machine learning approach that meets the speed and complex signal cognitive abilities required for the proposed cognitive GPR. As shown in Figure 4, A DBNN processes GPR B-scan images at multiple scales to detect subsurface infrastructure conditions. Each layer of feature detectors maps more complex relationships within the streaming GPR data. The output from the DBNN is used in the feedback loop embodying perception and action mechanisms to equip the GPR with intelligence to maximize the inspection information gain. Different deterioration stages can be classified and confidence values for the classifications can be provided in real time.

IV. Dynamic Computation Task Scheduling

A. The Communication Models of the GPR Front End

As discussed previously, GPR data pre-processing, analysis, and intelligence generation involves different computation tasks. Let D_i^{in} and L_i^{in} denote the input data and data size of computation task i, respectively. If computation task i is executed by the local microcomputer, output data D_i^{out} with size L_i^{out} will be produced. Let C_i be the CPU processing cycles of computation task i. Assume that the CPU at the local microcomputer is operating at frequency f^l with power consumption P^l. Then, the computation time needed to execute computation task i at the GPR front end is $T_i^{l\_cp} = C_i/f^l$; and the local energy consumption on computation is $E_i^{l\_cp} = T_i^{l\_cp} \cdot P^l$.

Assume that if the computation task i is executed locally at the GPR front end, output data D_i^{out} needs to be transmitted to the fog server. Let P^{tx} be the transmit power of the wireless transmitter at the GPR front end to communicate with the fog server and B the system bandwidth. The achievable throughput for transmitting D_i^{out} is $r_i = B\log_2\left(1+\frac{\gamma_i \cdot P^{tx}}{N_0 \cdot B}\right)$ where γ_i is the channel power gain which is assumed to be constant during transmitting the data of computation task i; and N_0 is the noise power spectral density at the receiver of fog server. The communication delay and energy consumption of transmitting D_i^{out} can be calculated as $T_i^{l\_cm} = L_i^{\text{out}}/r_i$, and $E_i^{l\_cm} = P^{tx} \cdot T_i^{\text{out}}$, respectively. Since the computation delay on input data D_i^{in} and the transmission delay on output data D_i^{out} of task i could overlap in time, the overall delay of processing task i at the GPR front end, $t_i^{l\_pr}$, satisfies

$$\max\{T_i^{l\_cp}, T_i^{l\_cm}\} \leq t_i^{l\_pr} \leq T_i^{l\_cp} + T_i^{l\_cm} \tag{4}$$

where $\max\{T_i^{l\_cp}, T_i^{l\_cm}\}$ corresponds to the case where the maximum time overlap between computation and transmission takes place; $T_i^{l\_cp} + T_i^{l\_cm}$ is the case where the transmission of D_i^{out} starts right after the computation on input data D_i^{in} ends without time overlap.

The total energy consumption of processing computation task i locally, defined as the sum of energy consumption on both computation and communication, is

$$E_i^l = E_i^{l\_cp} + E_i^{l\_cm}. \tag{5}$$

B. The Communication Models of the GPR Back End

Assume that if the computation task i is offloaded to the fog server, input data D_i^{in} needs to be transmitted to the fog server for task execution. With the throughput r_i, the delay of transmitting input data D_i^{in} to the fog server is $T_i^{f\_cm} = L_i^{\text{in}}/r_i$. The corresponding energy consumption on data transmission is $E_i^{f\_cm} = P^{tx} \cdot T_i^{f\_cm}$. We assume that the fog server has powerful computation capability through parallel computing. Thus, the delay of executing a computation task at the fog server is negligible.

C. The Proposed Offloading Policy

Next, an offloading policy will be presented for the scheduler at the GPR front end to dynamically determine whether a computation task should be executed locally or offloaded to the fog server.

The scheduler maintains a computation task buffer. We assume that different tasks in the buffer are scheduled on a first-come, first-served basis. Let $\mathbf{B} = \{1, 2, \cdots, i-1, i\}$ denote the task buffer having i tasks at the present time with the ith task being the latest one to be processed. It is also assumed that both the local computation resources and channel side information for the already scheduled tasks are available to the scheduler so that based on these information the scheduler could estimate a timeline of the completion of the scheduled tasks. Let $t^q_{1:i-1}$ be the overall time needed to complete the processing of the $i-1$ tasks existing in the buffer when the ith task enters the buffer. $t^q_{1:i-1}$ can be considered as the queuing delay of task i before it is processed. Note that among these $i-1$ tasks, some tasks may be locally executed at the GPR front end, and others may be offloaded to the fog server. Let $d^l_i, d^f_i \in \{0,1\}$ denote the offloading decision indicator for computation task i, i.e., if the task is decided to be executed by the local microcomputer at the GPR front end, $d^l_i = 1$ and $d^f_i = 0$; otherwise $d^l_i = 0$ and $d^f_i = 1$. Thus, the overall time needed to complete all of the i tasks present in the buffer can be estimated as

$$t^q_{1:i} = t^q_{1:i-1} + d^l_i \cdot t^{l\_pr}_i + d^f_i \cdot T^{f\_cm}_i. \tag{6}$$

Algorithm 1: The proposed computation task offloading policy.

```
1  Calculate $t_i^{l\_pr}$, $T_i^{f\_cm}$, and $t^q_{1:i-1}$;
2  if $T_i^{f\_cm} + t^q_{1:i-1} \le T_i^{\max}$ then
3      if $E_i^{f\_cm} \le E_i^l$ then
4          Offload computation task $i$ to the fog server by
           transmitting its input data $D_i^{in}$;
5      else
6          Execute computation task $i$ at the GPR front end,
           and transmit the resulting output data $D_i^{out}$;
7      end
8  else if $t_i^{l\_pr} + t^q_{1:i-1} \le T_i^{\max}$ then
9      Execute computation task $i$ at the GPR front end, and
       transmit the resulting output data $D_i^{out}$;
10 else
11     Drop computation task $i$.
12 end
```

The proposed computation task offloading policy for scheduling task i is shown in Algorithm 1. The offloading policy takes into account the energy limitation of the mobile GPR front end. Assume that each computation task i is associciated with a deadline $T^{\max}_i$. At the beginning, the scheduler calculates the estimated delay $T^{f\_cm}_i$ of transmitting data D^{in}_i, the estimated overall delay $t^{l\_pr}_i$, and the overall time $t^q_{1:i-1}$ needed to complete the $i-1$ task present in the buffer. If computation task i can be executed at the fog server before its deadline $T^{\max}_i$ (line 2), and at the same time, the energy consumption $E^{f\_cm}_i$ on transmitting input data D^{in}_i is less than the energy consumption E^l_i on local processing of task i (line 3), the computation task will be offloaded to the fog server (line 4). If computation task i can be executed at the fog server before its deadline $T^{\max}_i$ (line 2), and at the same time, the energy consumption $E^{f\_cm}_i$ on transmitting input data D^{in}_i is larger than the energy consumption E^l_i on local processing of task i (line 5), computation task i will be locally executed at the GPR front end; after the execution, the resulting output data D^{out}_i will be sent to the fog server (line 6). This way, the energy consumption at the GPR front end can be reduced as much as possible. If computation task i can not be executed at the fog server before its deadline $T^{\max}_i$ but it can be processed locally before the deadline (line 8), computation task i will be locally executed at the GPR front end; after the execution, the resulting output data D^{out}_i will be sent to the fog server. If computation task i can not be completed either at the GPR front end or at the fog server (line 10), it will be dropped from the buffer (line 11).

Fig. 5. Picture of the test site

TABLE I
DATA TRANSMISSION DELAY AND DATA PREPROCESSING DELAY AT THE GPR FRONT END AND AT THE FOG SERVER WITH DIFFERENT NUMBER OF A-SCANS.

Measurement	Number of A-scan		
	150	300	600
GPR Data Size (MB)	45.55	89.42	177.30
Delay in transmitting data to the fog server (s)	10.00	18.44	39.25
Delay in preprocessing at the GPR front end (s)	7.25	8.00	8.51
Delay in preprocessing at the fog server (s)	1.61	1.67	1.79

V. Performance Evaluation

A. System Setup

In the experiment, we use a dual-band GPR system with GSSI SIR-30 control unit which contains all radar electronics to control GPR signal generation and receiving signal acquisition and transmission. The GPR front end hardware is configured to be operable in two frequency bands, 400 MHz and 1.6 GHz. By making the frequency band selectable, it facilitates to achieve optimum sensing resolution and sensing depth. A laptop at the GPR front end is utilized to coordinate the system operation, perform necessary preliminary data processing, and communicate with a fog server (Intel NUC Mini PC). The GPR back end resides at the fog server which is connected to the GPR front end through a street WiFi network.

In the context of fog computing, signal processing and intelligence generation tasks are partitioned and processed based on the task scheduling policy. The tasks that demand less computation resources are executed locally on the GPR front-end computer while sophisticated computation tasks are offloaded to the fog server. The processing results direct the cognitive analyzer to select GPR operational parameters. In this work, GPR operating frequency is considered as the parameter to be adjusted at the GPR transceiver module.

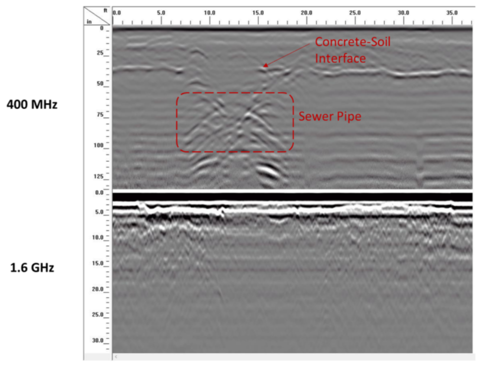


Fig. 6. B-Scan Images of sewer pipe obtained with 400 MHz and 1.6 GHz frequency settings

B. Experimental Results

For design evaluation, an underground sewer inspection is performed on an institutional campus. The picture of the site view is shown in Figure 5. In the operation, the GPR radiates short pulses toward the ground. The reflection signal at each location produces an A-scan waveform whose amplitude and phase parameters record the features of subsurface objects that the pulse encounters during its propagation. By moving GPR antennas to scan the survey area, numerous A-scan waveforms are collected. By assembling all A-scan waves together, B-scan images are obtained to produce more comprehensive views of the subsurface objects.

Table I shows the communication delay in transmitting GPR data from the GPR front end to the fog server as well as the delay in performing the computation task of data preprocessing at either the GPR front end or the fog server.

Figure 6 shows B-scan images obtained by the dual-band GPR. For comparison, images acquired with both frequency band settings are plotted. In the 400-MHz image, the sewer pipe pattern is detected and labeled. The pipe's burying depth is approximately 75 inches. While for the 1.6-GHz configuration, as GPR sensing depth can not surpass 30 inches, the sewer pipe is not detectable. As a result, in the following scan, the 400-MHz setting was selected by the cognitive analyzer at the fog server and sent back to the GPR transceiver.

VI. Conclusions

This work presented a fog computing framework for the development of cognitive portable GPRs. The system architecture of the proposed FC-enable cognitive portable GPR was developed. Different computation tasks in a typical perception-action cycle of cognitive GPRs were identified and explained. A computation task offloading policy was designed to determine whether a computation task should be executed at the local GPR computer or at the remote fog server. Experiment was conducted to demonstrate the efficacy of the proposed system.

References

[1] M. Chiang and T. Zhang, "Fog and IoT: An Overview of Research Opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[2] S. Mishra, D. Putha, J. Rodrigues, B. Sahoo, and E. Dutkiewicz, "Sustainable Service Allocation using Metaheuristic Technique in Fog Server for Industrial Applications," *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, Jan. 2018.

[3] D. Miao, L. Liu, R. Xu, J. Panneerselvam, Y. Wu, and W. Xu, "An Efficient Indexing Model for the Fog Layer of Industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, Jan. 2018.

[4] F. Tseng, M. Tsai, C. Tseng, Y. Yang, C. Liu, and L. Chou, "A Lightweight Auto-Scaling Mechanism for Fog Computing in Industrial Applications," *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, Jan. 2018.

[5] X. Feng, M. Sato, M. Butler, and C. Liu, "Subsurface Imaging Using a Handheld GPR MD System," *IEEE Geoscience and Remote Sensing Letter*, vol. 9, no. 4, pp. 659–662, July 2012.

[6] M. Chandra and T. J. Tanzi, "Drone-borne GPR Design: Propagation Issues," *Comptes Rendus Physique*, vol. 19, no. 1, pp. 72 – 84, 2018.

[7] H. M. Jol, *Ground Penetrating Radar Theory and Applications*. Elsevier, 2009.

[8] A. Turk, A. Hocaoglu, and A. Vertiy, *Subsurface Sensing*. Wiley, 2011.

[9] S. Haykin, "Cognitive Radar: A Way of the Future," *IEEE Signal Processing Magazine*, pp. 30–40, Jan. 2006.

[10] Y. Mao, J. Zhang, and K. Letaief, "Dynamic Computation Offloading for Mobile-Edge Computing With Energy Harvesting Devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, Dec. 2016.

[11] J. Liu, Y. Mao, J. Zhang, and K. Letaief, "Delay-Optimal Computation Task Scheduling for Mobile-Edge Computing Systems," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Jul. 2016.

[12] M. Chen, B. Liang, and M. Dong, "Joint Offloading and Resource Allocation for Computation and Communication in Mobile Cloud with Computing Access Point," in *Proc. IEEE INFOCOM*, May 2017.

[13] L. Ma, S. Yi, and Q. Li, "Efficient Service Handoff Across Edge Servers via Docker Container Migration," in *Proc. The Second ACM/IEEE Symposium on Edge Computing*, San Jose, CA, USA, Oct. 2017.

[14] Y. Zhang, P. Candra, G. Wang, and T. Xia, "2-D Entropy and Short-Time Fourier Transform to Leverage GPR Data Analysis Efficiency," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 1, pp. 103–111, 2015.

[15] X. Xu, T. Xia, A. Venkatachalam, and D. Huston, "Development of High-Speed Ultrawideband Ground-Penetrating Radar for Rebar Detection," *Journal of Engr. Mechanics*, vol. 139, no. 3, pp. 272–285, 2012.

[16] Y. Zhang, A. S. Venkatachalam, T. Xia, Y. Xie, and G. Wang, "Data Analysis Technique to Leverage Ground Penetrating Radar Ballast Inspection Performance," in *Proc. IEEE Radar Conference*, 2014.

[17] P. Jizba and T. Arimitsu, "On observability of Renyi's Entropy," *Physical Review*, vol. 69, no. 2, 2004.

[18] G. Hinton, S. Osindero, and Y. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation*, vol. 18, pp. 1527–1554, 2006.

[19] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, Jan 1979.