Multi-Sensor Underground Autonomous Inspection Platform

Chuong Phuoc Le[1], Pratik Walunj[1], Van Chung Nguyen[1], Yongyi Zhou[1],
An Duy Nguyen[1], Anton Netchaev[2], and Hung Manh La[1]

***Abstract*— The effective operation of civil infrastructure is crucial for economic stability. To ensure continued performance, regular maintenance is essential. However, underground infrastructure, like culverts and tunnels, posed more significant challenges in managing and preserving these critical assets such as maneuverability and danger, slow labor-intensive, defect localization, and superficial assessment. In this paper, we propose a cost-effective solution for infrastructure inspection through the development of an autonomous multi-sensor inspection robot platform. Our solution integrates multiple RGBD cameras, a LiDAR, a deep learning based defect detection system, lighting systems, and non-destructive evaluation (NDE) techniques for a comprehensive condition assessment. In addition, we integrate a Partially Observable Markov Decision Process (POMDP) framework to find and assess defect areas in confined and unstructured underground environments. Furthermore, we provide an open-source implementation of our framework on GitHub. Source codes are available*.**

I. Introduction

Effective maintenance of civil infrastructure is crucial for driving economic growth and promoting social progress. However, infrastructure such as culverts and tunnels are susceptible to deterioration over time, which can lead to serious issues if not properly managed. Global estimates indicate that infrastructure-related challenges could cost up to $79 trillion by 2040 [1]. To address this, it is essential to invest in both new infrastructure projects and the upkeep of existing ones. Maintaining current infrastructure is often more cost-effective than constructing new ones. This has led to increased efforts to automate the inspection process, aiming to reduce costs and minimize the risk of injuries associated with conventional inspection methods. While there have been many maintenance programs for infrastructure like bridges [2]–[10], underground like culverts and tunnels have not received comparable attention.

We interviewed several states to gather information on their culvert inspection methodologies. The Mississippi Department of Transportation (DOT) employs a manually controlled robot equipped with a 20-40 camera for inspecting small culverts, localizing defects based on the robot's traveled distance. The Nevada DOT has inspectors physically crawl into culverts with a camera to take pictures of defect.

In our previous work [11], we categorized the challenges of underground inspection, like culvert, into four main issues: (1) accessibility and safety risks (see Fig. 1), (2) labor-intensive procedures, (3) accurate defect localization, and (4) reliance on superficial information. The Mississippi DOT's approach addresses (1) and (2) by utilizing a robot, which eliminates the need for manual entry and reduces labor demands. However, it falls short in terms of (3) and (4). The reliance on distance traveled by the robot for defect localization is not robust, and the images captured are purely visual, often insufficient for a comprehensive assessment. Furthermore, the use of a 20-40 camera results in low-quality images, limiting the effectiveness of their inspections. In contrast, the Nevada DOT's approach falls short on all four issues. Their method of physically crawling into culverts with a camera not only compromises safety and accessibility but also remains labor-intensive. In addition, defect localization relies heavily on manual assessment, and the captured images provide only superficial information.

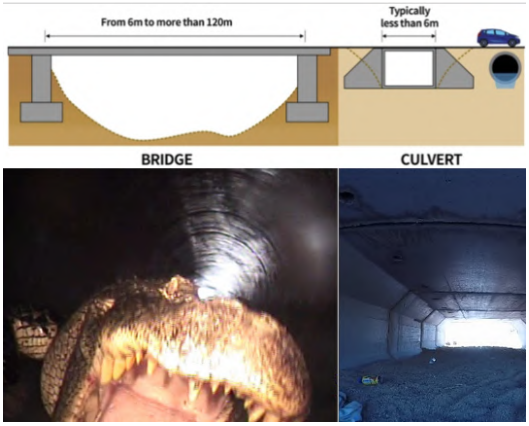

Fig. 1. The top image shows an example of underground tunnels like culvert [12]. The bottom left shows an alligator that attacked the inspection robot in MS. The bottom right shows the difficulty of traversing due to sand and mud. The bottom images were collected with/during DOT inspections.

Furthermore, to the author's best knowledge, most existing research, such as [13]–[17], continues to focus on visual (surface-level) assessments while requiring manual control. [17] utilize an unmanned aerial vehicles (UAV) to inspect culverts from the outside, restricting it to shorter culverts while outputting limited data. Likewise, [16] uses an autonomous mobile robot for surface inspections.

The challenge of autonomous underground inspection

This work was funded by the U.S. National Science Foundation (NSF) under grants NSF-CAREER: 1846513 and NSF-PFI-TT: 1919127, and the U.S. Army's Engineer Research and Development Center. The views, opinions, findings, and conclusions reflected in this publication are solely those of the authors and do not represent the official policy or position of the NSF and the U.S. Army.

[1]are with the Advanced Robotics and Automation (ARA) Lab, Department of Computer Science and Engineering, University of Nevada, Reno, NV 89557, USA.

[2]is with the USACE Engineer Research and Development Center (ERDC), Information Technology Lab, Vicksburg, MS 39180, USA.

* https://github.com/aralab-unr/MUAIP

Corresponding author: Hung La, email: hla@unr.edu

involves the robot traversing through rough environment, searching for unknown defects and assess its condition, similar to static object search. Unlike object detection and tracking, which focuses on perception, object search is primarily a decision-making problem; a robot to make decisions about where and how to search for a target [18].

Developing a solution is complicated, as real-world object search requires robots to make decisions with partial information. [19] uses a heuristic approach, where the goal is to reduce the distance traveled by the robot by determining an optimal exploration order of rooms in the environment and then guiding the robot to search for objects from viewpoints in order of each room. [20] employs a greedy algorithm that leverages prior knowledge of object relationships, creating a semantic link graph to facilitate object search. [21] introduced a Correlational Object Search POMDP (COS-POMDP) framework for the detection of hard-to-find objects, using YOLOv5 [22], by efficiently modeling object correlations while maintaining a reduced state space and using a hierarchical planning algorithm ensure optimal outcome.

These approaches - especially COS-POMDP - while effective, focus on single static object search. The task becomes more complex when searching for multiple objects. The Minnesota Distributed Autonomous Robot Team (MinDART) [23] utilizes a heuristic multi-robot system where homogeneous robots operate independently to locate, retrieve, and return targets to home base. Similarly, [24] addresses the challenge of fault-tolerant parallel search on an infinite line using multiple robots. [25] formulates an Object-Oriented POMDP framework for multi-object search given n objects by modeling each object independently without prior knowledge of their locations while employing a probabilistic observation model and using a planner based on the Monte Carlo Tree Search [26].

Building on the previous methods, it is clear that POMDP is well-suited for underground inspection systems since it excels at modeling sequential decision-making problems in environments where the agent cannot fully observe the state. In this paper, we propose the development of a multi-sensor autonomous culvert inspection robot. Our contributions can be summarized as follows:

(a) We introduced a new multi-sensor robot platform that utilizes multiple RGBD cameras, LiDAR and NDE technology to inspect defects in underground environments, addressing all four challenges outlined in [11].

(b) We introduce a POMDP-based framework designed to address the uncertainty faced by our multi-sensor platform. We also provide two other simple framework mode: full inspection and visual inspection.

(c) Our framework generates a visual data that highlights defects, like cracks and spalls, with a non-destructive evaluation (NDE) condition assessment.

(d) We validate the of our system through extensive testing in both simulated and real life environments, demonstrating consistent and reliable performance.

The organization of the paper is as follows: Section II discusses the mechanical design of the robot. Section III shows our POMDP formulation. Section IV includes the experiment and the result discussion. Section V gives some concluding remarks and future research directions.

II. Underground Inspection Robot Design

A. Mobile System

The 3D design of the robot system is shown in Fig. 2(b). To traverse through the rugged environment, we need a robot with strong tracks for traction and stability like the Bunker Pro. Moreover, it is resistant to water, dust and debris, making it suitable for this inspection task. The only drawback is its large size compare to the Bunker, making unsuitable for smaller underground environments.

B. Perception System

The perception of the robot consists of LED lights for darkness, RGBD cameras - Oak-D Pro - and Velodyne LiDAR as seen in Fig. 2(b), (c). The LEDs ensure adequate lighting to collect RGB images for defect detection. The RGBD cameras detect and localize defects suck as cracks and spalls by combining color and depth information as seen in Fig. 3. The Velodyne supplements the RGBD cameras by providing additional accuracy in alignment and 3D mapping.

C. NDE Deployment System

In Fig. 2(a), a diagram shows how an NDE Resipod ER sensor works while demonstrating the movements of the 5 DoF arm. The ER sensor, a contact-based NDE device, can measure the electrical resistance condition of an area [27]. Due to the correlation between deterioration and resistance, we used it to measure the defect areas for additional information on top of visual. To deploy the ER sensor, a 5-DOF arm - with a 0.8m extension - is designed to hold the sensor as its end-effector. The arm features two linear actuator, for extension movements to press the ER sensor on the defect, along with two joints and a rotational base for added flexibility.

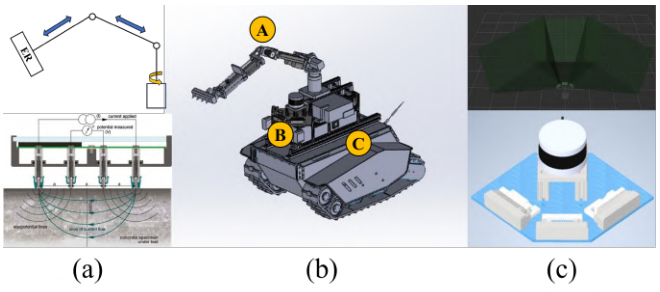

Fig. 2. The complete system design. (a) How ER and 5 DoF Arm works; (b) The overall design A: The NDE deployment system using ER sensor, B: The perception system consisted of RGBD cameras, LiDAR, and LEDs, C: The mobile system main body; (c) The FOV of the perception system: 61° vertical and 175° horizontal FOV.

III. Autonomous Inspection System

The robot is tasked with searching for n unknown static defect areas, def within the unstructured environment and assess its condition using the ER sensor mechanical arm. The set of possible defect area location is discrete and denoted as F. The location of the defect areas $def \in F$ are unknown. The center-line of the environment for searching is represented by C. We conceptualize autonomous inspection as a partially observable Markov decision process (POMDP), formulating it as a sequential decision-making problem where the environment state is not fully observable by the agent.

A. POMDP

A POMDP is a tuple $S, A, T, R, \Omega, O, \gamma$. S is a set of states, A is a set of actions, T is a set of transition models, $R : S \times A \rightarrow \mathbb{R}$ is the reward function, O is a set of observations, Ω is the observation model. At each timestep, an agent takes an action $a \in A$, which transitions state s to s' via $T(s', a, s) = Pr(s'|s, a)$. The agent will receives an observation $o \in O$ via $\Omega(o, s', a) = Pr(o|s', a)$ or $\Omega(o|s')$ depending on the sensor model. The agent will then receives a reward $r = R(s, a)$.

The agent must make decisions (a) under uncertainty of the true state since it does not directly observe the state. Hence, the agent must maintain and update a belief state $b_t : S \rightarrow [0, 1]$ in the true state after taking action a and observing o: $b'(s') = \eta Pr(o|s', a)\Sigma_{s \in S} Pr(s'|s, a)b(s)$ where is a normalizing constant.

To solve the POMDP, we need to find policy π that maps the history of an action which maximizes the expectation of future discounted rewards $V^{\pi}(h) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R\left(s_k, \pi\left(h\right)\right) \mid h_0 = h\right]$ where γ is the discount factor.

B. State Space

Our state denotes as $s = (s_r, s_d)$, where $s_r \in S_{robot}$ is the robot state and $s_d \in F$ is the defect state. The robot's state is defined as $s_r = [p_r, \theta_r]$ for its position and heading in the grid map. The state of the detection denoted as $s_d = [p, \theta]$ containing the position and orientation of the defect area.

C. Action Space

The robot has three fundamental actions: $Move$, $Declare$, $Coverage$, $Exit$. The $Move$ and $Coverage$ actions are simple robot movements from the eight actions around the robot: $forward$, $forward\_left$, $forward\_right$, $left$, $right$, $backward$, $backward\_right$, $backward\_left$. However, the both have different purposes as $Move$ is used to go to def while $coverage$ is to search for defects (more in Section III-I). The $Exit$ action means the robot has finished the inspection process. This only happens when all $def \in F$ have been in the FOV and all defects are $Declare$. $Declare$ is a manual action that deploys the mechanical arm for condition assessment a defect def with the ER sensor. The ER arm will only be deployed if $\exists$ def $\in$ DEF such that $A = true$

$$A = \begin{cases} true & \|\text{def} - s_r\| \leq \alpha * l_{arm} \\ false & otherwise \end{cases} \quad (1)$$

where l_{arm} is the length of the arm and α is the arm factor that prevents the defect from going out of range.

During our interview with multiple DOTs, the inspectors prefer to judge a defect for themselves. In addition, unlike other POMDP search problems [21], [25], the locations of def are not only unknown to the robot but to us as well. Hence, having the $Declare$ action be manual work well in tandem with the reward function $R(s, a)$ (more in III-F).

D. Transition function

For timestamp t, the agent takes an action $a \in \mathcal{A}$, causing the environment state to transition from s to s' $(s, s' \in \mathcal{S})$. In our case, the detection of static defects in leads the probability distribution to transition is determined $Pr(s'|s, a) = 1$. After transitioning states through an action, the agent receives an observation $o \in \mathcal{O}$ from the environment.

E. Observation Space & Model

The robot receives an observation through a viewing frustum projected from three mounted RGBD camera. The viewing frustum forms the FOV of the robot. The observation space O is factored with defects where $o = (o_r, o_d) \in O$ where o_r is the robot observation and $o_d = (o_{d,0}, o_{d,1}, ..., o_{d,n})$ is the defects observation.

To get o_d, the robot captures images of the search environment through the three mounted RGBD cameras and input it into our defect detection & localization system as shown in Fig. 3. In our system, YOLOv8 model is employed to detect defect for each RGBD camera, providing results in bounding boxes. Then, we take the pixel of the center of each defect bounding box and match it with the depth map to get the position (x, y, z) of $o_{d,i}$. One thing to note is that

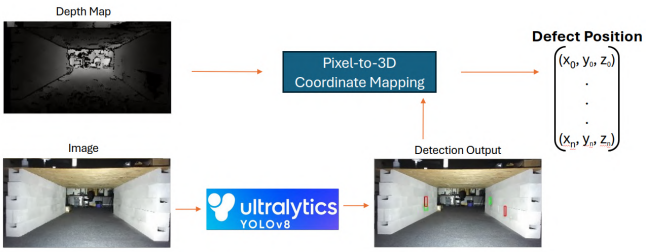

Fig. 3. Our defect detection and localization system.

we filtered out the bounding boxes to make sure we are not getting part of a defect on the picture. An image is define as $I \in \mathbb{R}^{W \times H}$ where W and H denote the width and height of the image, respectively. Each bounding box those vertices $v = [(r_0, c_0), ..., (r_4, c_4)]$. This condition has to be met for the bounding box to be considered a defect:

$$0 + \rho H < r_i < H - \rho H$$
$$0 + \varrho W < c_i < W - \varrho W$$

where ρ and ϱ are the tolerance factors.

Moreover, to ensure consensus between between the different RGBD camera, if there exist where $o_d^z, ..o_d^x \in o_d$, if $\|o_d^z - o_d^x\| \leq \delta$, we replace the two with $o_d^{new} = \frac{o_d^z + o_d^x}{2}$, where δ is the width of $f \in F$.

To address the uncertainty in observations, it is crucial to define the observation probability $\Omega(o, s, a) = Pr(o|s, a)$ of observations given the state and action. $Pr(o^i|s, a) = \epsilon * YD^i$ where YD is the detection model while ϵ is the confidence score.

F. *Reward function*

The reward function $R(s, a) = R(s_r, def, a)$ is given base on the the correctness of declarations determined a human inspector who has since as mentioned in Section III-C. Since it is in real life, we assume that defect's true location is not known, unlike [21], [25], where the environment knows the absolute value of their search target(s). When the $Declare$ action is called, the human verify if there really is a defect. If there is one, the robot will receive $+1000$. If not, the robot will receive -1000. All other actions except $exit$ will receive -2.

When the robot reaches the $exit$, if all def are $declare$, it will receive $1000n$ as the goal is to find all defects and assess their conditions. As long as all defects are identified, a few false positives are acceptable since we do not except our system to be perfect. This helps balance the penalty for false positives, as long as they remain few, while still penalizing false negatives.

G. *Belief Update*

Given the inherent uncertainty in POMDPs about the state of the environment, we represent the belief distribution for each defect state as a discrete set of possible defect arena in the environment F. The belief of defect i can be update as follow:

$$b_{t+1}^i = \eta Pr(o^i|s'^{,i}, a)\Sigma_s Pr(s'|s, a)b_t^i, \tag{2}$$

where η is the normalization factor, $Pr(s'|s, a) = 1$ as the transition probability for defect since it is static, and $Pr(o|s', a)$ as the observation probability. b_t^i is created and added to b_t if there exit $b_t^j \in b_t$ where $\|max(b_t^j) - o_d^r\| > \delta$ where $max()$ returns a defect discrete state $\in F$ of defect j with the highest probability, o_d^r is the new defect observation, and δ is width of $f \in F$.

H. *Detector*

We utilize YOLOv8, one of the new and popular cutting edge real-time object detector. After careful consideration, our detection model, $YD(o^i, s_d^i, s_r)$

$$= \begin{cases} 1 - TP & o^i = \text{null} \\ \phi f(o^i|\mu, S) & \|o^i - s_d^i\| \leq 1.5\delta \\ \frac{\phi}{\|FOV\|} & \|o^i - s_d^i\| > 1.5\delta \end{cases} \tag{3}$$

where δ is the width of $f \in F$, $S = \mathbf{I}^{3\times3}\sigma^2$, f denotes Gaussian distribution, and TP=0.81 is the true positive rate from validating. We determined that for our purposes, the RGBD data needs to be highly accurate. However, most RGBD camera, while accurate at close distances, is not good at longer distances and our Oak-D Pro is no exception. Therefore, we set a maximum distance range of max_r meters. Within this range, the sensor's performance is acceptable, but it performs significantly better at distances between its minimum range min_r, 0.7 meters and a threshold distance, $d_{threshold}$ = 2 meters. To account for the varying accuracy with distance

$$\phi = \begin{cases} e^{\omega(\|o_i - s_r\| - d_{\text{threshold}})} & \|o_i - s_r\| \geq d_{\text{threshold}} \\ 1 & \|o_i - s_r\| < d_{\text{threshold}} \end{cases}$$

where $\omega = -\frac{\ln(2)}{2}$ to reflects the improved accuracy of the sensor as the distance decreases, emphasizing that closer ranges yield better results. The mechanical design of the robot ensures that $\|o_i - s_r\| \geq min\_r$.

Algorithm 1 Autonomous Inspection Planner

function $Planner(s_r, b_t)$
  if b_t is empty or all($b^i \in b_t$) is visited **then**
    if all($c \in C$) is visited **then**
      action = $Exit$
    else
      action = $Coverage$
    end if
  else
    $OOB = Sample(b_t)$ from [28]
    $temp\_goal$ = nearest_def(s_r, OOB)
    if $\|s_r$ - $temp\_goal$ $\| < \alpha * a\_l$ **then**
      action = $Declare$
    else
      action = $Move$
    end if
  end if
end function
function $Coverage(s_r)$
  $temp\_goal \leftarrow closest\_unvisit(c \in C)$
  $neighbors \leftarrow neighbor(s\_r)$
  h = $\|$neighbors$-$temp_goal$\|$
  g = $\|$neighbors$-s_r$ $\|$
  f = g + h
  $neighbor\_goal$ = min(f)
  goTo($neighbor\_goal$)
end function
function $Move(s_r, temp\_goal)$
  $neighbors \leftarrow neighbor(s\_r)$
  h = $\|$neighbors$-$temp_goal$\|$
  g = $\|$neighbors$-s_r$ $\|$
  f = g + h
  $neighbor\_goal$ = min(f)
  goTo($neighbor\_goal$)
end function

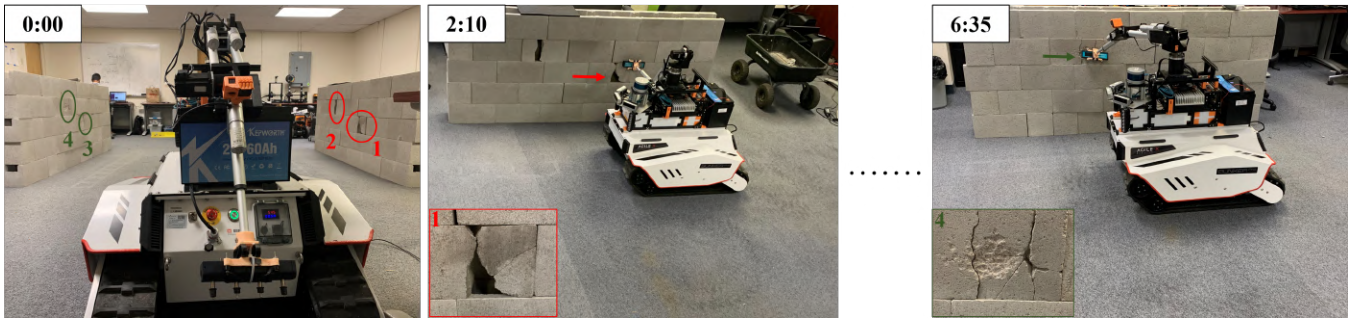

Fig. 4. The real life experiment for our POMDP framework. The first image shows the robot at timestep 0. Red and Green circles are spalls and cracks. The robot search, move to, then declare the first spall defect in the second image at timestep 5. The inspection process continues until the robot finds the last defect in the third image at timestep 13. We removed the roof for better visualization.

I. Planner

To solve our POMDP formulation, we created an online POMDP planner to plan an action as seen in Algorithm 1. We determine where the defects are base on the belief update and the sample belief from [28]. To reach the goal to $Declare$ we used heuristics to find best neighbor of s_r to go to.

IV. Experiment

The experimental environment for our study is comprised of two distinct settings: a simulated tunnel environment with different number of cracks in Gazebo, and real life culvert, built for this research using a combination of old and new concrete blocks. These environments were chosen to ensure a comprehensive evaluation of our system under various conditions, encapsulating both controlled and natural settings. The real life culvert provided insights into its performance in a real-world scenario.

F is generated by taking the inner areas the tunnel CAD model via triangle mesh, while C generates the center line in F. In the case of low triangle mesh, we generated a set uniform areas base on the inner length, width, height, and scale.Given the unique challenges and specifications inherent to culvert inspection endeavors, the study opts to undertake a comparative analysis of our three autonomous inspection system approaches: the full inspection, the visual inspection, and the our POMDP-based inspection. The visual inspection navigates the environment through coverage until all $f \in F$ is visited while the full inspection approach employs the utilization of an ER sensor, applying to every point in F. All methods uses the same defect detection, localization and assessment protocol.

A. Dataset

The dataset were collected with the Nevada Department of Transportation (DOT) from four culverts within Nevada and from online to train, validate, and test the performance of the Real-Time Culvert Defect Detection System. As a result, 1862 high-quality images were obtained. To increase our dataset, we augmented each image ten times using various data augmentation techniques, including vertical and horizontal flips, random rotations, translation, shear, brightness, super-pixel, and Gaussian blur. Then, we split the augmented images into 90%-10% to validate the performance of the Defect Detection System.

B. Simulation

The simulated environment is used to evaluate the efficiency of the autonomous navigation system. It consists of a 27 x 3 grid for robot and $\|F\|$ = 252 possible defect location. We conduct three different experiment with the simulated environment with varying number of defects: 2_def, 3_def, and 5_def. For each domain, we ran 50 trials to collect data. The robot took an average of 28, 30, and 35 actions for 2_def, 3_def, and 5_def. The reason the number of actions are so close despite different number defects is while the robot $Move$ to and $Declare$ defects actions might be different, the $Coverage$ action of making sure all $f \in F$ is visited are very similar.

TABLE I

A Comparison for different defect number in sim

Methods	**Time(sec) 2/3/5 defects**	**Defects Found&FP**	**Condition**
Full Inspect	491/491/491	2/3/5 0/0/1	N/A
Visual	114/114/114	2/3/5 0/0/1	N/A
Our approach	112/114/225	2/3/5 0/0/1	N/A

TABLE II

A Comparison in real life

Methods	**Time(sec)**	**Defects Found&FP**	**Condition**
Full Inspect	834	4 & 2	Yes
Visual	132	4 & 2	No
Our approach	398	4 & 2	Defects

TABLE III

Average Discounted Cumulative Reward (DCR)

	Simulations (2/3/5 defects)	**Real life**
DCR	2419/3988/6903	7196
Total defects	2/3/5	4

Table III illustrate results for each experiment.While the number of actions is similar, the actual time in simulation differs, as the robot moves diagonally more frequently in the 5_def environment compared to the others. Our approach

is as fast as the visual inspection process when dealing with a low number of defects. Although the detection of more defects slows the process, it remains faster than the full inspection method. It also only has 1 false positive detection and no false negative detection. Additionally, NDE condition assessment using the ER sensor is not applicable in simulations.

C. Real World

We evaluate the real world using a Bunker Pro mobile robot as seen in Section II. The robot inspect a 2.4 m x 3.4 m x 1.1 m (L x W x H) culvert. We decomposed the workspace of the robot into a 15 x 3 grid while $\|F\| = 35$ with arm reach a_l of 1.2 meters.

Fig. 6 depicts the sequence of actions performed by the robot. The robot search and detects the defects using the RGBD sensor, then go directly to it to take the ER reading. Over the 10 trial runs, the robot took an average of around 12.4 actions to find and declare all objects with an average of 398 seconds and an average reward of 7196.

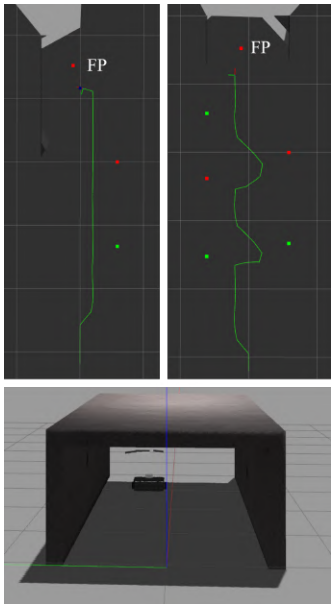

Fig. 5. The trajectory of 2_def and 5_def. Red is spall, green is crack. There's also a FP at the end of the tunnel. Each square is $1m^2$.

The limited number of actions with the long duration in our approach is primarily due to the time it takes for the arm to move, which can range from 60 to 120 seconds depending on its starting position. The optimal accumulated reward should exceed $n \times R_{\text{max}} - R_{\text{max}}$, where n is the total number of defects and $R_{\text{max}} = 1000$. In both simulations and real-world tests, our autonomous system achieves $DCR > n \times R_{\text{max}} - R_{\text{max}}$, indicating optimal performance. It should be noted that DCR is not applicable to our alternative approaches. These results demonstrate the efficiency of our system in autonomously operating within real-world environments.

D. Culvert inspection results

Outside of autonomous navigation, our approach outputs visual and condition assessment of defects as shown in Fig. 6. Our visual approach gives us detection of defects like spalls and cracks. Our full inspect approach provides visual and a comprehensive condition map of the whole environment. Our POMDP-based approach provides visual and a condition map of defects, which is the best all-around since it provides the visual assessment of defects while taking less than half the time of full inspect approach. However, we presents the other two as alternative approaches depending on the inspector's requirements.

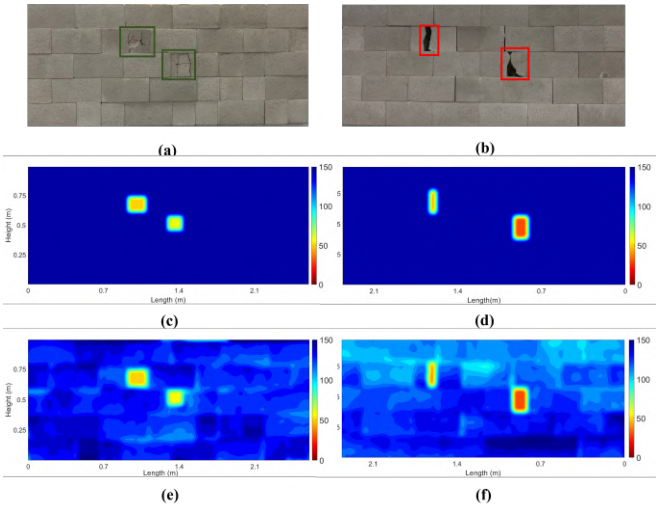

Fig. 6. The visual data (top) with crack and spall labels and ER condition map (bottom 2) of the real life environment. (a) (c) and (e) is the left culvert wall where entrance is left. (b), (d) and (f) is the right wall with the entrance on the right. The visual approach outputs only the top row. The full inspect approach outputs the top and bottom row. Our approach outputs top and mid row. The condition metric ER unit is Ωm, where 120 $<$ ER means good, 80 $<$ ER $<$ 120 means fair, and ER $<$ 80 means poor. While the full inspect approach condition map provides more information, the additional data is **not necessary** since inspectors are only concern with the poor regions.

V. Conclusion and Future Work

This paper introduces a multi-sensor autonomous inspection platform designed for efficient traversal in underground environments. It employs a POMDP-based inspection system to ensure comprehensive coverage and offers both visual and condition assessments for defects such as spalls and cracks. Additionally, it provides alternative inspection methods for inspectors, including visual and full inspections. As a proof of concept, this paper highlights the potential of the underground autonomous inspection system but acknowledges several limitations that warrant future improvements: (1) the navigation system needs further testing in diverse environments to demonstrate its robustness; (2) the ER arm is manually controlled; (3) It is limited to a single NDE condition assessment using the ER sensor; (4) The state space of defect F is uniform, having a large defect in multiple $f \in F$ can cause problem to the belief update.

References

[1] B. R. Kumar, "Trends in infrastructure industry," in *Project Finance: Structuring, Valuation and Risk Management for Major Projects*. Springer, 2022, pp. 1–29.

[2] H. Ahmed, S. T. Nguyen, D. La, C. P. Le, and H. M. La, "Multi-directional bicycle robot for bridge inspection with steel defect detection system," in *2022 IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems (IROS)*, 2022, pp. 4617–4624.

[3] H.-D. Bui, S. Nguyen, U.-H. Billah, C. Le, A. Tavakkoli, and H. M. La, "Control framework for a hybrid-steel bridge inspection robot," in *2020 IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems (IROS)*, 2020, pp. 2585–2591.

[4] T. Le, S. Gibb, N. Pham, H. M. La, L. Falk, and T. Berendsen, "Autonomous robotic system using non-destructive evaluation methods for bridge deck inspection," in *2017 IEEE Intern. Conf. on Robotics and Automation (ICRA)*, 2017, pp. 3672–3677.

[5] C. P. Le, A. Q. Pham, H. M. La, and D. Feil-Seifer, "A multi-robotic system for environmental dirt cleaning," in *2020 IEEE/SICE Intern. Symp. on System Integration (SII)*, 2020, pp. 1294–1299.

[6] H. Ahmed, C. P. Le, and H. M. La, "Pixel-level classification for bridge deck rebar detection and localization using multi-stage deep encoder-decoder network," *Developments in the Built Environment*, vol. 14, p. 100132, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666165923000145

[7] T. Yasmin, C. Le, and H. M. La, "Deep architecture based spalling severity detection system using encoder-decoder networks," in *International Symposium on Visual Computing*. Springer, 2022, pp. 332–343.

[8] C. P. Le, C. Ellison, S. Bunkley, H. La, and A. Netchaev, "A real-time multi-camera auto-adjustment framework for infrastructure inspections," in *2024 IEEE/SICE Intern. Symp. on System Integration (SII)*, 2024, pp. 681–686.

[9] H. Ahmed, C. P. Le, and H. M. La, "Pixel-level classification for bridge deck rebar detection and localization using multi-stage deep encoder-decoder network," *Developments in the Built Environment*, vol. 14, p. 100132, 2023.

[10] A. D. Nguyen, C. Le, P. Walunj, T. N. Do, A. Netchaev, and H. La, "A multi-model fusion of lidar-inertial odometry via localization and mapping," in *2024 IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems (IROS)*, 2024.

[11] C. P. Le, P. Walunj, A. D. Nguyen, Y. Zhou, B. Nguyen, T. Nguyen, A. Netchaev, and H. M. La, "Cais: Culvert autonomous inspection robotic system," in *2024 IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems (IROS)*, 2024.

[12] Wikipedia, the free encyclopedia, "Culverts," 2022. [Online]. Available: https://en.wikipedia.org/wiki/Culvert

[13] C.-W. Ou, C.-J. Chao, F.-S. Chang, S.-M. Wang, J.-N. Lee, R.-D. Hung, B. Chiu, K.-Y. Cho, and L.-T. Hwang, "Design of an adjustable pipeline inspection robot with three belt driven mechanical modules," in *2017 IEEE Intern. Conf. on Mechatronics and Automation (ICMA)*. IEEE, 2017, pp. 1989–1994.

[14] "The weasel 1: Full scope robotic inspection crawler," https://www.forbestusa.com/blogs/resources/the-weasel-1-full-scope-robotic-inspection-crawler.

[15] H. Miura, A. Watanabe, M. Okugawa, and T. Miura, "Verification and evaluation of robotic inspection of the inside of culvert pipes," *J. of Robotics and Mechatronics*, vol. 31, no. 6, pp. 794–802, 2019.

[16] A. Rumaksari, A. G. Sooai, G. S. Abimanyu, G. Dewantoro, H. K. Wardana, B. Murtianta, and L. B. Setyawan, "Real world design and implementation of pathfinding sewer inspection robot using a-star algorithm," *Jurnal Mantik*, vol. 7, no. 1, pp. 202–215, 2023.

[17] N. E. Serrano, "Autonomous quadrotor unmanned aerial vehicle for culvert inspection," Ph.D. dissertation, 2011.

[18] K. Zheng, "Generalized object search," *arXiv preprint arXiv:2301.10121*, 2023.

[19] R. Izquierdo-Cordova, E. F. Morales, L. E. Sucar, and R. Murrieta-Cid, "Searching objects in known environments: Empowering simple heuristic strategies," in *RoboCup 2016: Robot World Cup XX 20*. Springer, 2017, pp. 380–391.

[20] Z. Zeng, A. Röfer, and O. C. Jenkins, "Semantic linking maps for active visual object search," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 1984–1990.

[21] K. Zheng, R. Chitnis, Y. Sung, G. Konidaris, and S. Tellex, "Towards optimal correlational object search," in *2022 Intern. Conf. on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7313–7319.

[22] G. Jocher, "Yolov5 by ultralytics," 2020. [Online]. Available: https://github.com/ultralytics/yolov5

[23] P. E. Rybski, A. C. Larson, H. Metcalf, D. Skyllingstad, H. Veeraraghavan, and M. L. Gini, "Mindart: A multi-robot search and retrieval system." in *AAAI Mobile Robot Competition*, 2002, pp. 83–91.

[24] J. Czyzowicz, K. Georgiou, E. Kranakis, D. Krizanc, L. Narayanan, J. Opatrny, and S. Shende, "Search on a line by byzantine robots," *International Journal of Foundations of Computer Science*, vol. 32, no. 04, pp. 369–387, 2021.

[25] A. Wandzel, Y. Oh, M. Fishman, N. Kumar, L. L. Wong, and S. Tellex, "Multi-object search using object-oriented pomdps," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 7194–7200.

[26] D. Silver and J. Veness, "Monte-carlo planning in large pomdps, in 'advances in neural information processing systems (nips)'," 2010.

[27] S. Gibb, T. Le, H. M. La, R. Schmid, and T. Berendsen, "A multi-functional inspection robot for civil infrastructure evaluation and maintenance," in *2017 IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems (IROS)*, 2017, pp. 2672–2677.

[28] D. Silver and J. Veness, "Monte-carlo planning in large pomdps," in *Advances in Neural Information Processing Systems*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., vol. 23. Curran Associates, Inc., 2010. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2010/file/edfbe1afcf9246bb0d40eb4d8027d90f-Paper.pdf