CAIS: Culvert Autonomous Inspection Robotic System

Chuong Phuoc Le[1], Pratik Walunj[1], An Duy Nguyen[1], Yongyi Zhou[1],
Binh Nguyen[2], Thang Nguyen[2], Anton Netchaev[3], and Hung Manh La[1]

Abstract— **Culverts, essential components of drainage systems, require regular inspection to ensure their optimal functionality. However, culvert inspections pose numerous challenges, including accessibility, manpower, defect localization, and reliance on superficial assessments. To address these challenges, we propose a novel Culvert Autonomous Inspection Robotic System (CAIS) equipped with advanced sensing and evaluation capabilities. Our solution integrates RGBD camera, deep learning, lighting systems, and non-destructive evaluation (NDE) techniques to enable an accurate and comprehensive condition assessment. We present a pioneering Partially Observable Markov Decision Process (POMDP) framework to resolve uncertainty in autonomous inspection, especially in confined and unstructured environments like culverts or tunnels. The framework outputs detailed 3D maps highlighting visual defects and NDE condition assessments, demonstrating consistent and reliable performance for indoor and outdoor scenarios. Additionally, we provide an open-source implementation of our framework on GitHub, contributing to the advancement of autonomous inspection technology and fostering collaboration within the research community. Source codes are available*.**

I. Introduction

Culvert inspections play a vital role in ensuring the optimal functionality of drainage systems. Serving as smaller counterparts to bridges, culverts facilitate the passage of pedestrians and vehicles over roads, rails, and waterways. However, the inspection of culverts is fraught with challenges, which can be categorized as follows:

1. Accessibility and Danger: Narrow and confined culverts impede workers' maneuverability, while posing risks of collapse and potential exposure to hazardous chemicals and gases.

2. Manpower and Speed: The extensive length of culverts demands significant manpower, especially when utilizing multiple inspection tools, leading to inefficient resource utilization.

3. Defect Localization: In GPS-denied and poorly lit environments, pinpointing defect locations within culverts is a tough task.

4. Superficial Info: Conventional culvert inspection processes predominantly rely on visual inspections, providing only surface-level information without a comprehensive condition assessment.

To solve problems **1** and **2**, we develop an autonomous inspection robot specifically designed for culvert assessments. To solve problem **3**, our solution involves a combination of lighting systems, deep learning methodologies, and RGBD sensor. This synergy allows the robot to operate effectively in GPS-denied and dark environments, pinpointing defect locations accurately. Recognizing the limitations of visual inspections, we integrate non-destructive evaluation (NDE) methods, widely employed in civil structure inspections [1]–[6] to assess the subsurface of culverts, solving problem **4**.

In recent years, the utilization of robots for comprehensive inspections has surged, owing to their capability to access challenging environments and deliver high-quality data in a secure and cost-efficient manner [2], [7]–[12]. However, the development of culvert inspection robots has lagged behind. Existing research, such as [13]–[17], predominantly focuses on surface-level visual assessments and most current culvert inspection robotic systems rely on manual operation rather than autonomous functionality. For example, the study in [17] primarily explores deploying robots for external visual inspection of culverts using unmanned aerial vehicles (UAVs), limited to shorter culverts and providing superficial data as the UAVs did not conduct thorough internal inspections. Similarly, [16] confines its inspections to surfaces within a known environment, neglecting the challenges posed by unknown environments. Consequently, achieving autonomous inspection remains challenging due to uncertainties in the robot's localization using its onboard sensors.

The challenge of autonomous culvert inspection involves robots exploring unknown environments and searching for defects without prior knowledge.

In terms of exploration, the frontier-based method stands out as one of the earliest strategies. Its fundamental concept involves identifying frontiers as the boundaries between known and unknown spaces. The robot subsequently chooses one of these frontiers as its next destination for movement [18] [19] [20].

Conducting a search in a vast area involves taking actions over various sources of uncertainty in a partially observable environment. Hence, several works have employed the

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. Department of Transportation, Office of the Assistant Secretary for Research and Technology (USDOT/OST-R) under Grant No. 69A3551747126 through INSPIRE University Transportation 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, the USDOT/OST-R, or any other entities.

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

[2]are with the Department of Engineering, Texas A&M University–Corpus Christi, Corpus Christi, TX 78412, USA.

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

* `https://github.com/aralab-unr/CAIS`

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

POMDP for object search [21]–[26]. The study [21] initially introduced a 3D Multi-Object Search (3D-MOS) formulated in a POMDP in a volumetric observation space. The belief is represented in an octree-belief constructed with multi-resolution voxels. Additionally, a framework SLOOP [23] for partially observable decision-making employs a probabilistic observation model for spatial language which computes the POMDP planner based on Monte Carlo [27] Tree Search. Furthermore, the Correlational Object Search POMDP (COS-POMDP) proposed in [24] introduces a framework for searching small, hard-to-detect objects. It models correlations while maintaining optimal solutions with a minimized state space. [25] initially presents a system for multi-object search (MOS) in a 3D region that is robot-independent and environment-agnostic by taking local point cloud, object detection results, and robot's localization as input, and giving an output as a 6D viewpoint for movement through online planning.

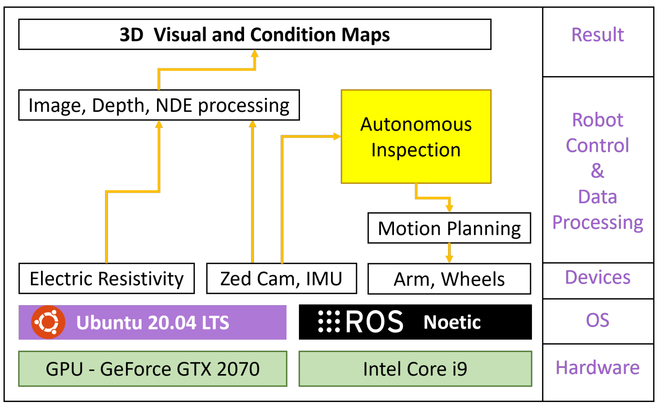


Fig. 1. The flowchart of CAIS.

Inspired by previous researches, we propose CAIS, an autonomous inspection framework formulated in a POMDP that addresses navigation, exploration, and detection of defect areas in confined environments, such as culverts. The workflow is shown in Fig. 1. In this paper, our novel contributions can be summarized as follows:

(a) We introduce the pioneering POMDP framework tailored to resolve uncertainty challenges for autonomous inspections within the confined and unstructured environments.

(b) Our framework is designed to produce a three-dimensional (3D) representation that outlines visual anomalies, including cracks and spalls, with a 3D NDE condition assessment map, facilitating a detailed inspection and analysis of the structural integrity.

(c) We validate the efficacy of our framework through extensive verification in diverse indoor and outdoor scenarios, showcasing its consistent and reliable performance across varied environmental conditions.

(d) We enhance accessibility and foster collaboration by making the source code openly available on GitHub.

The organization of the paper is as follows. Section II discusses and analyzes the mechanical design of the robot. Section III shows our POMDP navigational framework. Section IV includes the experiment, its parameters, and the discussion of the results. Section V gives some concluding remarks and future research directions.

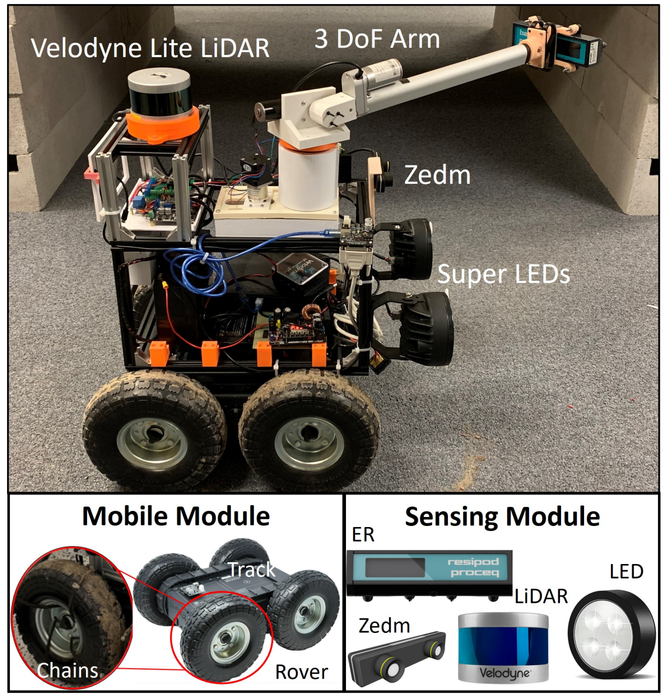


Fig. 2. The overall design of the culvert inspection robot. The robot body is a rover mobile robot. Super-LEDs provide needed light conditions for working in the darkness of culverts. The camera collects visual and depth data, and the Electrical Resistivity (ER) sensor checks concrete quality with physical contact.

II. Culvert Inspection Robot Mechanical System

The mechanical system of the robot in Fig. 2 comprises of two modules: the Mobile Module and the Sensing Module. The Mobile Module contains a conventional four wheel-drive robot equipped with chains on its wheels to improve traction on challenging terrains, including sand, mud, ice, and obstacles like debris and branches. On the other hand, the Sensing Module consists of visual and physical sensors that facilitate inspection and data acquisition. Visual data are captured by the ZED camera, which also provides robot poses. The ER sensor, a contact-based device, is used to examine the condition of damaged areas [28]. To deploy the ER sensor, a 3-DOF arm in Fig. 2 is designed to hold the sensor as an end effector. It has a full extension range of 0.71 m. As a proof-of-concept paper, the ER arm will be manually controlled. Moreover, two LEDs are integrated into the robot to ensure adequate lighting in the dark environment of culverts. An Intel NUC computer is responsible for the control and computing tasks, while a 20Ah acid battery provides up to three hours of working time.

III. POMDP-Based Autonomous Navigation

The robot is tasked with searching for unknown defect areas within the unstructured environment with unknown obstacles. We conceptualize autonomous inspection as a POMDP, formulating it as a sequential decision-making problem where the environment state is not fully observable by the agent. Our formulation is presented as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, T, O, R, Def, C, \gamma)$, where $\mathcal{S}, \mathcal{A}, \mathcal{O}$ are the state, action, and observation space, respectively. T, O, R, and γ represent the transition, observation model, and reward functions with a discount γ. S and O are factored into a list of defects $Def = \{def_0, ..., def_n\}$ [26] where each object contains a class $c \in C = \{crack, spall\}$ and a pose. The task is to find a policy $\pi(b_t)$, which maximizes the expectation of future discounted rewards $V^{\pi}(b_t) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R(s_{t+k}, \pi(b_{t+k})) \mid b_t\right]$. The details of the POMDP model are described as follows:

A. State Space:

A state denotes as $s = (s_r, s_d, s_f) \in \mathcal{S}$, where s_r, s_d, and s_f present for the state of the robot, the state of *unknown/known* detection, and the state of the frontier. 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 $s_d = [p_d, b_d]$ containing the position of the estimated defect areas and its bounding boxes, and $s_f = [p_f, \theta_f]$ represents the exploration position of the frontier and the difference in heading between the current robot's position and the searching point.

B. Action Space:

Autonomous inspection generally necessitates three fundamental capabilities: moving, searching, and declaring a *defect* in the grid environment. Formally, the action space encompasses these three types of elementary actions: *MOVE*(s_r, g) moves the robot from the current position to the goal g, where the robot can use the arm with ER sensor for measurement *defects d*. The goal calculation is summarized in Algorithm 1 where l_{arm} is the full extension length of the robot's arm, and α is the arm factor that avoids the defection being out of range for measurement $(0.5 \leq \alpha \leq 1.0)$. *SEARCH*$(s_f$, g) changes the robot's position and heading to explore the environment and search for new *defects* using the frontier points. *DECLARE*(s_r, s_d, o_d) consists of two main tasks: First, it declares whether the belief distribution area is high enough then deploy the ER to measure the belief distribution area with the highest belief. Otherwise, this action changes the current robot pose to confirm detection and updates the belief. *DONE*(s_r) action is used to stop the robot or to mark the completion of a task. For each action taken, the state will be updated as follows:

Algorithm 1 MOVE estimation

Require: g_t : the pose of defect

$$g_t^i = \begin{cases} d_t^i - \alpha * l_{arm} & \text{if } ||d_t^i - s_{r_{q_r}}|| \leq \gamma \text{ in y\_axis}, \\ d_t^i + \alpha * l_{arm} & \text{otherwise.} \end{cases}$$

goTo(g_t^i)

It should be noted that the velocity v_t associated with the MOVE action is determined through the implementation of rudimentary control mechanism. This velocity is subject to variation at each discrete time interval, Δ_t, contingent upon two primary factors: the magnitude of the spatial displacement between the robot's current location and the designated target, and the angular discrepancy between the robot's present orientation and the desired trajectory towards the goal, the latter of which is ascertained through the application of the $atan2$ function. In the *SEARCH* action, the frontier algorithm selects the best frontier points to update the current robot pose. The set of frontiers is reset and changed at every timestamp, ensuring there is no '*undefined*' relationship between the current frontier and the next estimation.

Algorithm 2 Update state estimation

Require: s, a_t : The state and action at current frame.
Output: s' : The state estimation
for current action $a_t \in \mathcal{A}$ **do**
  if $a_t \leftarrow$ *MOVE* **then**

$$s' = \begin{cases} s'_r \leftarrow s_r + v_t \Delta t, \ \& \ v_t \text{ is the velocity} \\ s'_d \leftarrow d_t, \\ s'_f \leftarrow \emptyset, \end{cases}$$

  else if $a_t \leftarrow$ *SEARCH* **then**

$$s' = \begin{cases} s'_r \leftarrow s_f, \\ s'_d \leftarrow \emptyset, \\ s'_f \leftarrow \textit{undefined}, \end{cases}$$

  else if $a_t \leftarrow$ *DECLARE* **then**

$$s' = \begin{cases} s'_r \leftarrow s_r, \\ s'_d \leftarrow \begin{cases} o_d, & \text{if high belief} \\ s_d \cap o_d, & \text{otherwise} \end{cases} \\ s'_f \leftarrow \emptyset, \end{cases}$$

  else if $a_t \leftarrow$ *DONE* **then**
   RESET()
  end if
end for

When the robot finishes inspecting a defect area, a function *RESET* is used to reset the state and mark this spot as "*visited*".

C. 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 this case, the observation is the detection of static defects in culverts, and 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.

D. Observation Space & Model:

The robot captures images of the search environment through a mounted camera, and an observation $o \in \mathcal{O}$ is generated. To address the uncertainty in observations, it is crucial to define the probabilistic distribution $Pr(o_i|s', a_t)$ of observations given the previous state and action in the current frame. YOLOv8 model is employed for defect detection, providing results in bounding boxes. Then, an observation is denoted as $o_t = [q_j^d, w_j, h_j, p_j]$, where q_j^d represents the centering position of the detection *j* in pixel coordinates, and w_j, h_j indicate the width and height of the bounding box, with p_j being the probability of the detection from YOLOv8. A *detection function* classifies the observation into two statuses: *UNKNOWN*, *POTENTIAL*.

In the context of our analysis, we define the image input as $I \in \mathbb{R}^{W \times H}$, where W and H denote the width and height of the image, respectively. Utilizing YOLOv8 for defect detection, we obtain the center pixel position of each identified defect area, represented by the coordinates (r, c). The subsequent step involves the localization of the pixel, a process facilitated by aligning the detected pixel location with the spatial mapping array $xyz\_map \in \mathbb{R}^{W \times H}$. This spatial mapping array is derived from the depth map, with each element of xyz_map corresponding to a 3-dimensional coordinate (x, y, z) in physical space, as opposed to the pixel values found in I.

For a pixel $i \in I$, located at the coordinates (r, c), the corresponding spatial position, denoted as q, is directly obtained from $xyz\_map[r, c]$. This direct correspondence facilitates a precise matching process, enabling the accurate localization of defect areas within the three-dimensional space. The relationship can be formally expressed as:

$$q_j^d = (x, y, z) = xyz\_map[r, c], \tag{1}$$

where q_j^d signifies the 3-dimensional spatial position of the j^{th} detected defect within the image, thereby establishing a foundational methodology for our bounding box localization process within the three-dimensional domain.

Algorithm 3 Autonomous Inspection

```
Require: s_t, a_t, b_t : The current state, action, and belief at
  time t
  while t ≤ T_max, and a_t ≠ DONE do
      for o_t ∈ Observations do
          if o_t^i ∉ Def then
              declare def_new
              Def.append(def_new)
          end if
      end for
      if def ∈ Def not visited > 0 then
          g_c ← closestDefect(s_r, def)
          a_t ← MOVE(g_c)
          if MOVE is success & Max(b^i) > β then
              a_t ← DECLARE(s_r, g_c)
          end if
      else if def ∈ Def not visited = 0 then
          gf_max ← MaxWeight(frontier)
          a_t ← SEARCH(sf_max)
      end if
      UPDATE_BELIEF(b, a, o)
  end while
  function UPDATE_BELIEF(b, a, o)
      for o^i ∈ Def do
          ε^i = (1/m) Σ_{k=0}^{m} p_k^i
          b_{t+1}^i = η Pr(o^i | s'^{,i}, a) Σ_s Pr(s' | s, a) b_t^i
      end for
  end function
```

The observations o^i is consist of n defect-specific observations. If defect is detected by the RGBD sensor using YOLO, then the scenario is considered POTENTIAL, P. If not, it is considered UNKNOWN, U. As such, $Pr(o^i|s, a)$ = $\Sigma_{h^i \in \{U^i, P^i\}} Pr(o^i|h^i, s)Pr(h^i|s)$. In the P scenario, the observation is normally distributed with μ as the true object i position $Pr(o_t|P^i, s) = \eta f(o_t^s|\mu, S)$ where the covariance matrix $S = \mathbf{I}^{3\times3}\sigma^2$ and η is the normalization factor. Moreover, $Pr(P^i|s) = \epsilon^i$ and $Pr(U^i|s) = 1-\epsilon^i$. In scenario U, $Pr(o^i|U^i, s)$ = 1, $Pr(P^i|s) = 1-\epsilon^i$, and $Pr(U^i|s) = \epsilon^i$. The accuracy of the model is denoted as $\epsilon^i = \frac{1}{m}\sum_{k=0}^{m} p_k^i$ where p is probability generated by YOLOv8 and m is the total defect-specific observation of i.

E. *Reward function:*

It receives rewards only if it receives a set of detections from the environment by transitioning from s to $s' \in \mathcal{S}$. *MOVE* and *DECLARE* actions receive a reward R_{max}(+100) depending on the robot state and the current beliefs $b_t(s)$. For example, the updated belief $b_{t+1}(s')$ could be higher than the current one $b_t(s)$ if the *defects* are correctly confirmed by the *DECLARE* action. Otherwise, the robot receives a reward R_{min}(-100).

F. *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 sparse sphere of points (twenty-seven) around the initial observation location. The belief of defects 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, and $Pr(o|s', a)$ as the observation probability.

In conclusion, the inspection algorithm is summarized in Algorithm III-D.

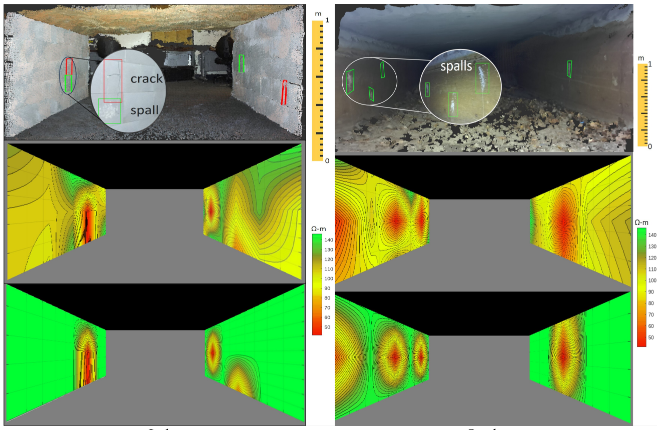
IV. Experiment

The computer we are using is an Intel Mini NUC 11. A flowchart of our framwork and its hardware can be seen in Fig. 1. The Zed camera and our defect detection system utilize the computer's GPU. The images are 960 × 540 for W × H, a_l = 0.71 m, γ = 0.7, β = 0.8, and σ = 0.5. And the model can processed the image at a 65 FPS rate.

The experimental environment for our study is comprised of two distinct settings: an indoor culvert, built for this research using a combination of old and new concrete blocks, and an existing outdoor culvert. These environments were chosen to ensure a comprehensive evaluation of our system under various conditions, encapsulating both controlled and natural settings. The indoor culvert facilitated a controlled assessment of the system's capabilities, while the outdoor culvert provided insights into its performance in a real-world scenario. The culverts are shown in Figure 3.

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, 770 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, superpixel, and Gaussian blur. Then, we split the augmented images into 90%-10% to validate the performance of the Defect Detection System.

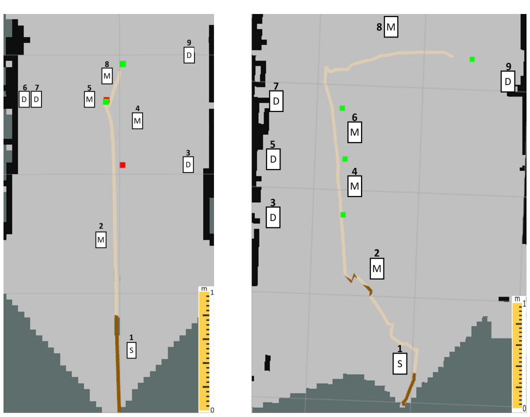


Indoor Outdoor

Fig. 3. 3D map of the indoor culvert generated by RTAB-Map with crack and spall labels and ER condition map (bottom 2) outputted by manual and and our proposed method. The condition metric ER unit is Ωm, where 120 < ER means good, 80 < ER < 120 means fair, and ER < 80 means poor. While manual inspection does have more information about the condition of the culvert, the extra info is **not relevant** since inspectors are only concern with the poor regions.

B. *Culvert inspection results*

Given the unique challenges and specifications inherent to culvert inspection endeavors, it is This study opts to undertake a comparative analysis of our autonomous inspection system against two methods: the pure exploration approach and the manual inspection technique. The exploration approach navigates the environment until all frontiers are explored while the manual approach employs the utilization of an ER sensor, applying at intervals of 0.3048 meters across the culvert. It is imperative to emphasize that both methods uses the same 3D mapping and assessment protocol as our approach.



Indoor Outdoor

Fig. 4. Trajectory and action of robot for indoor and outdoor culverts. The robot takes actions *MOVE* (M), *SEARCH* (S), *DECLARE* (D) respectively.

We present an in-depth evaluation of CAIS, focusing specifically on comparative analysis regarding the efficiency of the inspection process with discounted cumulative reward, and the dimensional accuracy of the generated 3D maps.

Table I showned the comparison between other inspection methods - manual control and exploration approaches - and CAIS. When the robot is manually operated, it achieves a comprehensive inspection of the culvert with an average duration of 556 seconds for indoor settings and 645 seconds for outdoor environments. Conversely, the exploration method, which is solely visual data acquisition, is much faster, completing the mission in approximately 49 seconds for indoor and 67 seconds for outdoor scenarios.

Although our inspection strategy outpaces the manual approach, requiring only 167 seconds for indoor and 237 seconds for outdoor inspections, it is still slower than the exploration approach. Nonetheless, our method generated a partial ER condition map that contains crucial assessment details absent in the exploration method, whilst eschewing the superfluous data characteristic of the manual approach. As shown in Figure 3, it is apparent that an examination of every segment is unnecessary, particularly for non-defective areas, which are in good condition.

TABLE I

A COMPARISON BY APPLYING DIFFERENT APPROACHES

Methods	Time(s) Indoor/Outdoor	Visual	Condition (ER)
Manual	556/645	Yes	Yes (defects & non-defect)
Exploration	49/67	Yes	N/A
Our approach	167/237	Yes	Partial (only defects)

TABLE II

AN EVALUATION OF 3D MAP DIMENSION ERROR IN METERS

	Length	Entrance Width	End Width	Entrance Height	End Height
Indoor	0.029	0.089	0.078	0.066	0.054
Outdoor	0.033	0.093	0.092	0.076	0.059

TABLE III

AVERAGE DISCOUNTED CUMULATIVE REWARD (DCR)

	Indoor Culvert	Outdoor Culvert
DCR	378	342
Total defects	4 (2 spalls & 2 cracks)	4 (spalls)

In summary, our approach facilitates a fast culvert inspection while delivering essential information. The manual method required more time and often yields additional, less pertinent, data. Conversely, our method, though slower than the exploration approach, provides comprehensive insights, unlike the latter, which typically offers only surface-level information.

Due to the challenges of obtaining ground truth for a 3D culvert map, we did dimensional size comparisons of our estimated 3D map and the real one like in [29], as shown in Table II. The trajectory of the robot is depicted in Fig. 4 for indoor and outdoor experiments.

Initially,the robot performs a *SEARCH* action at the beginning until an observation triggers by a detection, switching to *MOVE/DECLARE*. The current robot's pose and the inspection goal are calculated in Algorithm I. Additionally, the belief and pose of the agent are updated at every timestamp. The reward R_{max}(+100) is achieved only if the *defects* are declared. Additionally, the agent receives a positive/negative reward of +5 for *SEARCH* $\rightarrow$ *MOVE*, *MOVE* $\rightarrow$ $DECLARE$, and -5 for $MOVE$ $\rightarrow$ *SEARCH* actions. The total discounted accumulated reward is shown in Table III. The optimal accumulated rewards should be bigger than $n * R_{max} - R_{max}$ (n is the total number of *defects*). Our algorithm demonstrates good result since $DCR{>}n * R_{max} - R_{max}$, indicating that the algorithm achieves optimal results. DCR is not applicable for exploration and manual approaches.

V. Conclusion and Future Work

The paper introduces CAIS as a pioneering solution for culvert inspections. CAIS employs a POMDP inspection system for efficient traversal, ensuring thorough coverage, and generates a detailed 3D map of the culvert, highlighting defects such as spalls and cracks. It demonstrate the reliability, solving all four problems presented in Section I. While CAIS represents a significant advancement, further improvements is possible. These include integrating a high Degree of Freedom (DoF) manipulator autonomous arm, developing an auto-virtual boundary fence for exploration efficiency, utilizing multiple vision sensors for enhanced 3D map quality and localization accuracy, and more quantitative analysis by testing on more culverts in the city.

References

[1] H. M. La, N. Gucunski, K. Dana, and S.-H. Kee, "Development of an autonomous bridge deck inspection robotic system," *J. of Field Robotics*, vol. 34, no. 8, pp. 1489–1504, 2017.

[2] 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.

[3] N. Gucunski, S. Kee, H. La, B. Basily, and A. Maher, "Delamination and concrete quality assessment of concrete bridge decks using a fully autonomous rabit platform," *Structural Monitoring and Maintenance*, vol. 2, no. 1, pp. 19–34, 2015.

[4] H. Ahmed, H. M. La, and N. Gucunski, "Review of non-destructive civil infrastructure evaluation for bridges: State-of-the-art robotic platforms, sensors and algorithms," *Sensors*, vol. 20, no. 14, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/14/3954

[5] L. Van Nguyen, S. Gibb, H. X. Pham, and H. M. La, "A mobile robot for automated civil infrastructure inspection and evaluation," in *2018 IEEE Intern. Symp. on Safety, Security, and Rescue Robotics (SSRR)*, 2018, pp. 1–6.

[6] S. Gibb, H. M. La, T. Le, L. Nguyen, R. Schmid, and H. Pham, "Nondestructive evaluation sensor fusion with autonomous robotic system for civil infrastructure inspection," *J. of Field Robotics*, vol. 35, no. 6, pp. 988–1004, 2018.

[7] 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.

[8] 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.

[9] 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.

[10] 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

[11] 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.

[12] 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.

[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] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE Intern. Symp. on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation'*, 1997, pp. 146–151.

[19] A. Dai, S. Papatheodorou, N. Funk, D. Tzoumanikas, and S. Leutenegger, "Fast frontier-based information-driven autonomous exploration with an mav," in *2020 IEEE Intern. Conf. on Robotics and Automation (ICRA)*, 2020, pp. 9570–9576.

[20] A. Batinovic, T. Petrovic, A. Ivanovic, F. Petric, and S. Bogdan, "A multi-resolution frontier-based planner for autonomous 3d exploration," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4528–4535, 2021.

[21] K. Zheng, Y. Sung, G. Konidaris, and S. Tellex, "Multi-resolution pomdp planning for multi-object search in 3d," in *2021 IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems (IROS)*, 2021, pp. 2022–2029.

[22] T. Nguyen, V. Hrosinkov, E. Rosen, and S. Tellex, "Language-conditioned observation models for visual object search," in *2023 IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems (IROS)*, 2023, pp. 10 894–10 901.

[23] K. Zheng, D. Bayazit, R. Mathew, E. Pavlick, and S. Tellex, "Spatial language understanding for object search in partially observed city-scale environments," in *2021 30th IEEE Intern. Conf. on Robot Human Interactive Communication (RO-MAN)*, 2021, pp. 315–322.

[24] 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.

[25] K. Zheng, A. Paul, and S. Tellex, "A system for generalized 3d multi-object search," in *2023 IEEE Intern. Conf. on Robotics and Automation (ICRA)*, 2023, pp. 1638–1644.

[26] 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.

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

[28] 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.

[29] S. Bunkley, C. Ellison, G. Glaspell, J. Klein, and A. Netchaev, "Robotic localization in earth dam outlet works," in *2024 IEEE/SICE Intern. Symp. on System Integration (SII)*, 2024, pp. 816–820.