

CAIS: Culvert Autonomous Inspection Robotic System

Chuong Phuoc Le¹, Pratik Walunj¹, An Duy Nguyen¹, Yongyi Zhou¹,
Binh Nguyen², Thang Nguyen², Anton Netchaev³, and Hung Manh La¹

Abstract—Culverts, essential components of drainage systems, require regular inspection to ensure their optimal functionality. However, culvert inspections pose numerous challenges, including accessibility, defect localization, and reliance on superficial visual 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 deep learning methodologies, lighting systems, and non-destructive evaluation (NDE) techniques to enable accurate defect localization 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 in various 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, as illustrated in Fig. 1. 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.

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.

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

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

³is 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

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.

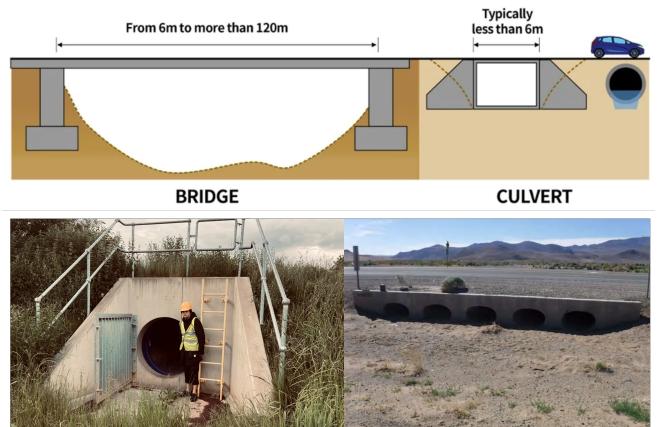


Fig. 1. Culverts and how they differ from bridges. Pictures from the top and middle left are from [1], [2] and some of the culverts where we collected data.

To solve problems **1** and **2**, we have taken a groundbreaking approach by developing an autonomous inspection robot specifically designed for culvert assessments. To solve problem **3**, our solution involves a sophisticated combination of lighting systems, deep learning methodologies, and depth sensors. This synergy allows the robot to operate effectively in GPS-denied and dark environments, enhancing its ability to pinpoint defect locations within the culvert accurately. Recognizing the limitations of visual inspections in providing comprehensive data, we integrate non-destructive evaluation (NDE) methods into our culvert inspection protocol. NDE techniques, widely employed in civil structure inspections like bridges [3]–[8] enable us to measure the subsurface of the culvert, are used to solve 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, dependable data in a secure and cost-efficient manner [4], [9]–[12]. However, the development of culvert inspection robots has lagged behind. Existing research, such as [13]–[17], predominantly focuses on surface-level visual assessments. Furthermore, most current culvert inspection robotic

systems rely on manual operation rather than autonomous functionality. For instance, 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. To address this challenge, we propose a novel framework based on Partially Observable Markov Decision Process (POMDP) to enhance the robot's autonomy and ensure robust performance in culvert inspections.

In this paper, our novel contributions can be summarized as follows:

(a) We introduce the pioneering POMDP framework tailored to resolve uncertainty challenges in autonomous inspections within confined and unstructured environments, such as culverts or tunnels.

(b) Our framework is designed to produce an exhaustive three-dimensional (3D) representation that meticulously outlines visual anomalies, including cracks and spalls, in conjunction with a 3D NDE condition assessment map. This integrated approach facilitates a detailed inspection and analysis of the structural integrity. For illustrative instances of identified cracks and spalls within the assessed structure, refer to Figure 3.

(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. To the best of our knowledge, this constitutes the initial open-source implementation for autonomous culvert inspection, addressing uncertainty concerns, and offering a significant contribution to the research community.

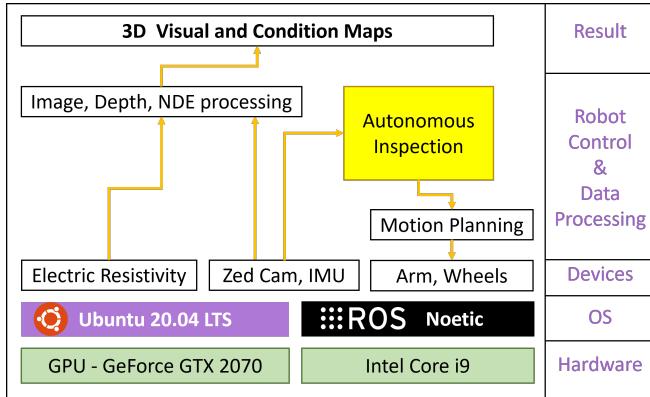


Fig. 2. The flowchart of the data-driven autonomous culvert inspection framework.

In summary, our CAIS working operation is meticulously

crafted to ensure a precise and thorough condition assessment through autonomous inspection, culminating in the generation of detailed 3D Visual and Condition Maps. The workflow is depicted in the flowchart shown in Fig. 2.

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



Fig. 3. The difference between a crack and a spall.

II. RELATED WORKS

The challenge of autonomous inspection involves robots exploring unknown environments and searching for objects 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]. In an initial work, [18] introduced the concept of frontiers by focusing solely on the nearest frontier from the current robot location. However, this approach exhibits limitations as the robot may navigate indefinitely in complex search environments, such as a maze. Numerous researchers have enhanced their methods to address these limitations. [19] introduced a hybrid method that combines with sampling-based approach, which reduces the number of candidate frontiers for next-view and leverages inherent voxel grouping in an octree map helps avoid treating frontier voxels as clusters if they exist in the same octant of the tree. Additionally, [20] exploits the OctoMap structure to easily query occupied voxels at various spatial resolutions. Frontier points are identified at the lowest Octomap level. The final frontiers are clustered using mean-shift methods to reduce the number of frontier points considered in further steps.

Conducting a search in a vast area or searching for objects involves taking actions over extended periods amid various sources of uncertainty in a partially observable environment. Hence, several works have employed the POMDP, a principled decision framework for autonomous object searching [21]–[25]. 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 [26] 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. Inspired

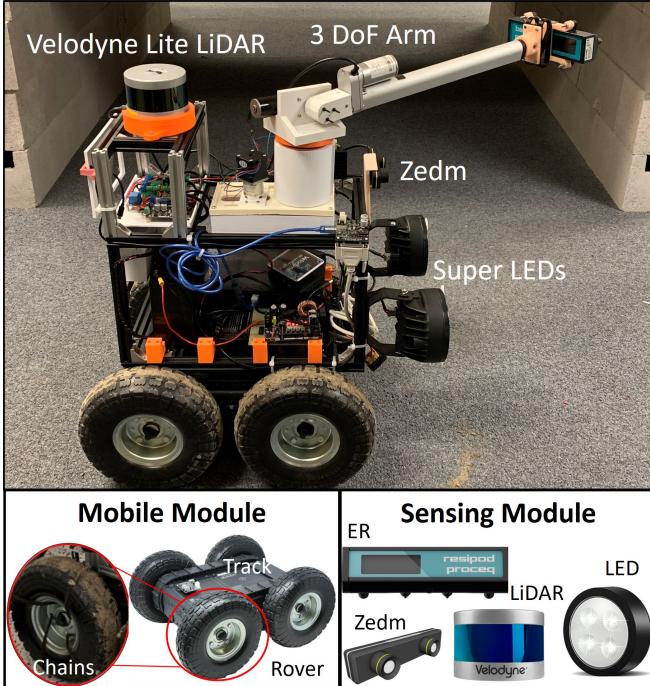


Fig. 4. The overall design of the culvert inspection robot. The robot body is a rover mobile robot. Super-LEDs provide needed light conditions for working in the darkness of culverts. The camera collects visual and depth data, and the Electrical Resistivity (ER) sensor checks concrete quality with physical contact.

by previous research, we propose CAIS with an autonomous inspection framework formulated in a POMDP that addresses navigation, exploration, and detection of defect (crack & spall) areas in confined environments, such as culverts or tunnels.

III. CULVERT INSPECTION ROBOT MECHANICAL SYSTEM

The mechanical system of the robot in Fig. 4 comprises two modules: the Mobile Module and the Sensing Module. The Mobile Module contains a conventional four-wheel-drive rover robot equipped with chains on its wheels to improve traction on challenging terrains, including sand, mud, ice, and obstacles such as debris, branches, stones, and trash. On the other hand, the Sensing Module consists of visual and physical sensors that facilitate inspection and data acquisition. Visual data are captured by a ZED 3D camera, which also provides robot poses. The ER sensor, a contact-based device, is used to examine the quality of concrete

in damaged areas [27]. To deploy the ER sensor, a 3-DOF manipulator in Fig. 4 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.

Furthermore, two super LEDs are integrated into the robot to ensure adequate lighting conditions in the dark environments of culverts. The Intel NUC computer is responsible for the control and computing tasks, while a 20Ah acid battery provides up to three hours of working time. Overall, integrating the Mobile and Sensing Modules allows the robot to navigate and operate in harsh environments while simultaneously conducting inspections and gathering valuable data.

IV. POMDP-BASED AUTONOMOUS NAVIGATION

The robot is assigned the task of searching for crack or defect areas with unknown size and location within a search environment (culvert or tunnel) that is unstructured and contains unknown obstacles. The search environment is represented in a 3D grid map. 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. In general, the model formulation is presented as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, T, O, R, \gamma)$, where $\mathcal{S}, \mathcal{A}, \mathcal{O}$ are the state, action, and observation space, respectively. T, O, R , and γ represent the transition, detection, and reward functions with a discount γ . The task is to find a policy $\pi(b_t)$, which maximizes the expectation of future discounted rewards $V^\pi(b_t) = \mathbb{E} [\sum_{k=0}^{\infty} \gamma^k R(s_{t+k}, \pi(b_{t+k})) | b_t]$. 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 *crack/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 *cracks/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 deflection 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 *cracks/defects* using the

frontier points. $DECLARE(s_r, s_d, o_d)$ consists of two main tasks: First, deploy the ER to measure the detection if there is high 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: d_t : Set of the detection at current frame
Output: g_t : Set of the goal for inspection

```

for  $i = [1, \dots, n]$ ;  $n$  is the number of detection  $\in d$  do
     $g_t^i = \begin{cases} d_t^i - \alpha * l_{arm} & \text{if } \|d_t^i - s_{r_{qr}}\| \leq \gamma \text{ in y-axis,} \\ d_t^i + \alpha * l_{arm} & \text{otherwise.} \end{cases}$ 
end for
```

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

It should be noted that the velocity, denoted as v_t , associated with the MOVE action is determined through the implementation of a 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. When the robot finishes inspecting a defect

area, a function *RESET* is used to reset the historical state and mark this spot as "*visited*".

C. Transition function:

For a 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 cracks/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. Besides, YOLOv8 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 the YOLOv8. A *detection function* classifies the observation into three statuses: *UNKNOWN*, *POTENTIAL*, and *HIGH*.

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 correlating 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:

$$q_j^d = (x, y, z) = xyz_map[r, c], \quad (1)$$

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.

When $d(o_t) = UNKNOWN$ denotes the absence of detected cracks/defects or free space at a specific location in the grid map, then $Pr(o_t|s', a_t)$ is a uniform distribution. Based on the probability results of YOLOv8 for cracks/defects, we classify them into two types: *POTENTIAL* and *HIGH*, each having different probabilistic distributions $Pr(o_t|s', a_t) = Pr(d(o_t) = POTENTIAL|s', a_t) = \alpha'$, and $Pr(o_t|s', a_t) =$

$Pr(d(o_t) = HIGH|s', a_t) = \beta$. The effectiveness of the observation model is independently governed by the hyper-parameters α' and β , and their sum is not necessarily equal to one for updating the belief. The methodology presented herein was developed drawing inspiration from the work described in [21]. Further details will be discussed in the next section.

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 *cracks/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, beliefs $b \in [0, 1]$ are employed to represent the probability distribution of the current state. The belief of detection (*cracks/defects*) can be update as follow:

$$b_{t+1} = \sum_{w_j} \sum_{h_j} \frac{P_t^i}{Norm_t} Pr(o_f|s', a) \sum_s Pr(s'|s, a) b_t, \quad (2)$$

Assuming each pixel stores an unnormalized belief value of *cracks/defects* P_t^i , where i is index of the detection pixels. For fast computation, instead of computing every pixel with different weights corresponding to HIGH or POTENTIAL, respectively. The normalized belief is computed as the sum of unnormalized belief values over the detection area, and $Norm_t$ represents the normalization value for the entire image. In the case where the detection is deemed POTENTIAL but the residual belief $|b_{t+1} - b_t| > \theta$, it means that if the detection indicates a high belief in a crack or spall, then the status should change from POTENTIAL to HIGH, and the values of pixels and norms should be updated as follows:

$$P_t^{i'} = Pr(d(o_t) = HIGH|s_t, a_t) P_t^i, \quad (3)$$

$$Norm_{t'} = Norm_t + P_t^{i'} - P_t^i. \quad (4)$$

In conclusion, the inspection algorithm is summarized in Algorithm 3.

V. 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. 2. The Zed camera and our defect detection system utilize the computer's GPU. The images are 960×540 for $W \times H$, $a_l = 0.71$ m, $\gamma = 0.7$, and $\alpha = 0.9$.

The experimental environment for our study was carefully constructed to simulate realistic conditions encountered in culvert inspection scenarios. This comprised two distinct settings: an indoor culvert, which was purpose-built for this research using a combination of old and new concrete blocks, and an existing outdoor culvert. These environments were

meticulously chosen to ensure a comprehensive evaluation of our system under varied 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 is shown in Figure 5 and 6.

Algorithm 3 Autonomous Inspection

```

Require:  $s_t, a_t, b_t$  : The current state, action, and belief at time  $t$ 
Output:  $s'_t, a_{t+1}, b_{t+1}$  : The updated state, and action, and belief
while  $t \leq T_{max}$ , and  $a_t \neq DONE$  do
     $d_{o_t} \leftarrow$  executes  $a_t$  and receives detection function in observation space
    Plan POMDP to update the belief, action, and state due to  $d_{o_t}$ 
    if  $d_{o_t} = POTENTIAL$  then
         $b_{t+1} \leftarrow UpdateBelief(b_t, a_t, o_t)$ , Eq. (2)
        if  $|b_{t+1} - b_t| < \theta$  then
             $a_{t+1} \leftarrow a_t$ 
        else
            for  $i \in I$  do
                 $P_t^{i'} \leftarrow Pr(d(o_t) = HIGH|s_t, a_t) P_t^i$ , Eq. (3)
                 $Norm_t' \leftarrow Norm_t + P_t^{i'} - P_t^i$ , Eq. (4)
            end for
             $b_{t+1} \leftarrow UpdateBelief(P_t^{i'}, Norm_t')$ , Eq. (2)
             $a_{t+1} \leftarrow MOVE(s_{r_t}, g_t)$ 
        end if
    else if  $d_{o_t} = HIGH$  then
         $b_{t+1} \leftarrow UpdateBelief(b_t, a_t, o_t)$ , Eq. (2)
         $g_t \leftarrow Inspecting\ estimation()$ 
         $dist = ||g_t - s_{r_t}||$ ,
        if  $mindist > \gamma$  then
             $a_{t+1} \leftarrow a_t$ 
        else
             $a_{t+1} \leftarrow DECLARE(s_{r_t}, s_{d_t}, o_t)$ 
        end if
    else if  $d_{o_t} = UNKNOWN$  then
         $b_{t+1} \leftarrow UniformDistribution()$ 
         $a_{t+1} \leftarrow SEARCH(s_{f_t})$ 
    end if
     $s_{t+1} \leftarrow Update\ state\ estimation()$ 
     $t \leftarrow t + 1$ 
end while

```

A. Dataset

Five datasets are collected and used in this paper to train, validate, and test the performance of the Real-Time Culvert Defect Detection System. The dataset was collected from four culverts within Nevada: three from Carson City with the Nevada Department of Transportation (DOT) and one from Reno. All culvert data were collected in the darkness with the robot using its light. The fifth culvert dataset was

obtained manually from the Internet. As a result, 770 high-quality images were obtained.

Since 770 is low, 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.

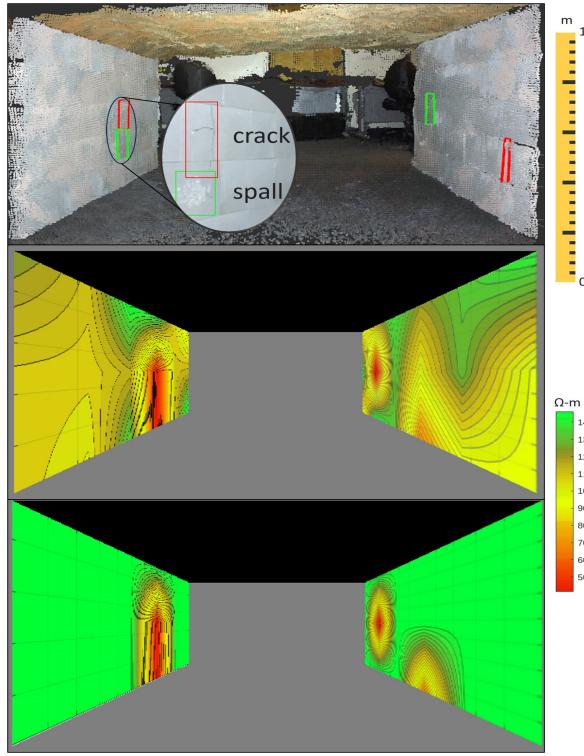


Fig. 5. 3D map of the indoor culvert generated by RTAB-Map with crack and spall labels and ER condition map (bottom 2) outputted by manual and our proposed method. The condition metric ER unit is Ωm , where $120 < \text{ER}$ means good, $80 < \text{ER} < 120$ means fair, and $\text{ER} < 80$ means poor. 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 not feasible to conduct a direct comparison of our methodology with other search strategies encapsulated within the POMDP framework. Consequently, this study opts to undertake a comparative analysis of our autonomous inspection system against two divergent methodologies: the pure exploration approach and the manual inspection technique. The exploration strategy systematically navigates the environment until all potential frontiers have been exhaustively explored. In contrast, the manual approach employs the utilization of an ER sensor, systematically applied at intervals of 0.3048 meters across the inspection area. It is imperative to emphasize that both the exploration and manual methodologies incorporate an identical 3D mapping and assessment protocol as employed by our autonomous system.

This research paper presents an in-depth evaluation of our autonomous inspection system, focusing specifically on a comparative analysis regarding the temporal efficiency of the inspection process, discounted cumulative reward, and the dimensional accuracy of the generated 3D maps. This comprehensive evaluation is intended to illuminate the operational efficacy, time efficiency, and the detail-oriented nature of the 3D mapping capabilities of our autonomous system within the context of culvert inspection tasks, thereby providing insights into its comparative advantages and identifying potential avenues for further refinement.

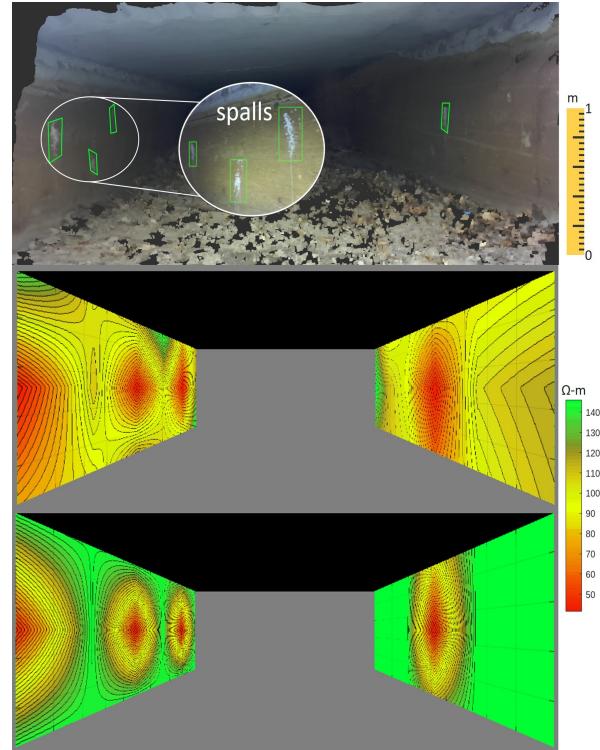


Fig. 6. This shows the same results as Fig. 5 but for the outdoor culvert. It is important to note that the presence of a step within the culvert posed a significant limitation to our inspection capabilities, as navigating steps falls outside the operational scope of our four-wheeled robotic platform. This resulted the inspection of part of the outdoor culvert.

In Table I, we conduct a comparative analysis of different inspection methodologies - namely, manual control and exploratory (visual) approaches - juxtaposed against our proposed method. When the robot is manually operated, it achieves a comprehensive inspection of the culvert, necessitating an average duration of 556 seconds for indoor settings and 645 seconds for outdoor environments. Conversely, the exploratory method, which relies solely on visual data acquisition, significantly expedites the inspection process, completing the mission in approximately 49 seconds for indoor and 67 seconds for outdoor scenarios.

Although our inspection strategy markedly outpaces the manual approach, requiring only 167 seconds for indoor and 237 seconds for outdoor inspections, it is still over threefold slower than the exploratory approach. Nonetheless, our method furnishes a partial ER condition map that

encapsulates crucial assessment details absent in the purely exploratory method, whilst eschewing the superfluous data characteristic of the manual approach. As delineated in Figures 5 and 6, it becomes apparent that a meticulous examination of every segment is unwarranted, particularly for blocks devoid of defects, which are otherwise in satisfactory condition. The focal point for inspectors lies in identifying defect areas and evaluating the severity of their condition through resistivity values, which bear implications on the structural integrity of the culvert.

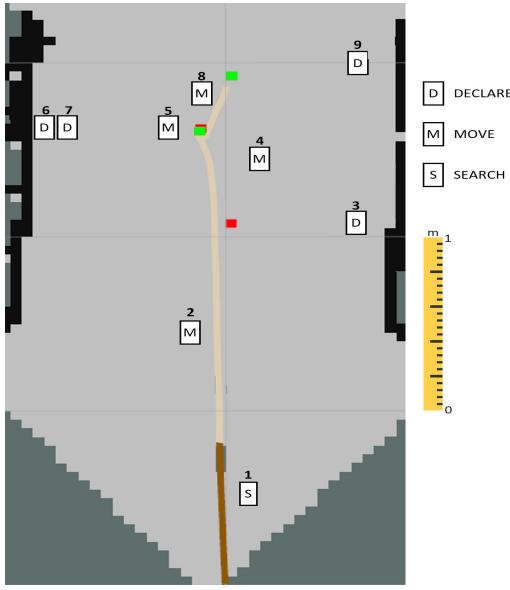


Fig. 7. Trajectory and action of robot for indoor culvert.

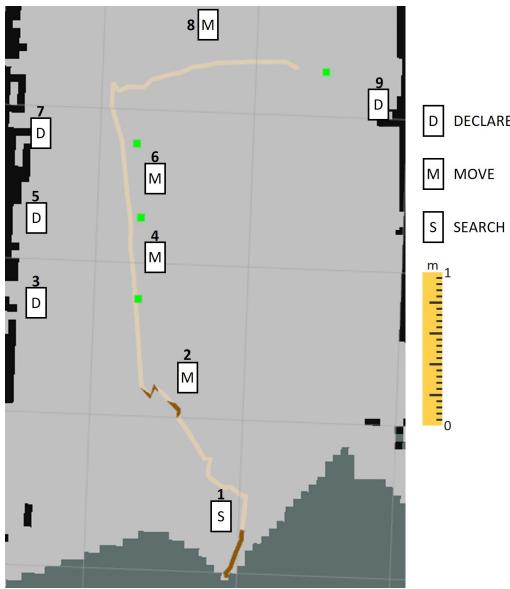


Fig. 8. Trajectory and action of robot for outdoor culvert.

In summary, our approach facilitates a faster culvert inspection compared to the manual method while delivering essential information. While the manual method may require more time, it often yields additional, albeit less pertinent,

data. Conversely, our method, though slower than the visual inspection approach, provides comprehensive insights, unlike the latter, which typically offers only surface-level information.

As shown in Fig. 2, our goal is to construct a 3D condition map highlighting defect location. However, obtaining ground truth for a 3D culvert map poses challenges. Therefore, we compared the dimensional size of our estimated 3D map with an actual one like in [28], as shown in Table II. Moreover, the trajectory of the agent is depicted in Fig. 7 and 8 for indoor and outdoor experiments.

TABLE I
A COMPARISON BY APPLYING DIFFERENT APPROACHES

Methods	Time(s)	In-door/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	427	411
Total defects	4 (2 spalls & 2 cracks)	4 (spalls)

Initially, the agent's belief is uniformly distributed, and it performs a *SEARCH* action at the beginning. The agent's pose is continually updated by the frontier points until an observation triggers a detection, switching the current action to *MOVE/DECLARE*. The green and red points represent the current robot's pose and the inspection goal, respectively, which are calculated in Algorithm I. Additionally, the belief and pose of the agent are updated at every timestamp after an action is taken. The maximum reward $R_{max}(+100)$ is achieved only if the cracks/defects are declared as well. Additionally, the agent receives a positive/negative reward of +5 for *SEARCH* → *MOVE*, *MOVE* → *DECLARE*, and -5 for *MOVE* → *SEARCH* actions, respectively. The total discounted accumulated reward is shown in Table III. Let's assume that the optimal accumulated received rewards are equal to $DCR = n * R_{max}$ (n is the number of cracks/defects) if the cracks/defects are well detected and declared. Then, the inspection algorithm is considered degenerated if $DCR \gg n * R_{max}$ or $DCR \ll n * R_{max}$. Our algorithm demonstrates decent results when DCR's value is very close to the optimal value, indicating that the algorithm

achieves optimal steps in inspecting *cracks/defects*. DCR is not applicable for exploration and manual approaches.

VI. CONCLUSION AND FUTURE WORK

The paper introduces the Culvert Autonomous Inspection Robotic System (CAIS), as a pioneering solution for culvert inspections, offering a blend of innovative navigation capabilities, comprehensive mapping techniques, and significant insights into structural integrity. CAIS employs a POMDP navigation system for efficient traversal, ensuring thorough coverage. Utilizing this system, CAIS generates a detailed 3D map of the culvert, highlighting defects such as spalls and cracks. Results demonstrate the reliability of our algorithm, evidenced by the translation error of the 3D map compared to the actual one, and the optimization of coverage while minimizing exploration time, as indicated by achieved rewards similar to the optimal. While CAIS represents a significant advancement, further improvements and expansions are possible. These include integrating a high Degree of Freedom (DoF) manipulator autonomous arm, developing an auto-virtual boundary fence for exploration efficiency, a better mobile platform for culvert steps and utilizing multiple vision sensors for enhanced 3D map quality and localization accuracy. In conclusion, CAIS presents an innovative solution for culvert inspections, combining advanced navigation capabilities, comprehensive mapping techniques, and valuable insights into structural integrity.

REFERENCES

- [1] Wikipedia, the free encyclopedia, “Culverts,” 2022. [Online]. Available: <https://en.wikipedia.org/wiki/Culvert>
- [2] Midas Bridge, “Culvert - solutions,” 2022. [Online]. Available: <https://www.midasbridge.com/en/solutions/culverts>
- [3] H. M. La, N. Gucunski, K. Dana, and S.-H. Kee, “Development of an autonomous bridge deck inspection robotic system,” *Journal of Field Robotics*, vol. 34, no. 8, pp. 1489–1504, 2017.
- [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 International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3672–3677.
- [5] N. Gucunski, S. Kee, H. La, B. Basil, and A. Maher, “Delamination and concrete quality assessment of concrete bridge decks using a fully autonomous robot platform,” *Structural Monitoring and Maintenance*, vol. 2, no. 1, pp. 19–34, 2015.
- [6] 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>
- [7] 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 International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2018, pp. 1–6.
- [8] 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,” *Journal of Field Robotics*, vol. 35, no. 6, pp. 988–1004, 2018.
- [9] 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 International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 4617–4624.
- [10] 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 International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 2585–2591.
- [11] 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 International Symposium on System Integration (SII)*, 2020, pp. 1294–1299.
- [12] C. Phuoc Le, C. Ellison, S. Bunkley, H. La, and A. Netchaeve, “A real- time multi-camera auto-adjustment framework for infrastructure inspections,” in *2024 IEEE/SICE International Symposium 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 International Conference 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,” *Journal 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 International Symposium 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 International Conference 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 International Conference 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 International Conference 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 International Conference 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 International Conference 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 International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1638–1644.
- [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 International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 2672–2677.
- [28] S. Bunkley, C. Ellison, G. Glaspell, J. Klein, and A. Netchaeve, “Robotic localization in earth dam outlet works,” in *2024 IEEE/SICE International Symposium on System Integration (SII)*, 2024, pp. 816–820.