2026 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 27 - October 1, 2026. Pittsburgh, PA, USA

MS-AIS: Multi-Sensor Autonomous Inspection System

Chuong Phuoc Le[1,2], Pratik Walunj[1], An Duy Nguyen[1], Van Chung 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, poses more significant challenges in managing and preserving these critical assets, such as maneuverability, danger, slow, labor intensive, defect location, and superficial assessment. In this paper, we propose a cost-effective solution for infrastructure inspection through the development of MS-AIS, an autonomous inspection system. Our solution integrates multiple vision cameras, a dense LiDAR, a CNN-based defect characteristics pipeline, lighting system, and non-destructive evaluation (NDE) methods for a comprehensive condition assessment. The system consists of three different autonomous modes that support targeted inspection, visual inspection, and full inspection. Experimental validation in simulated and real-world culverts demonstrates the effectiveness of MS-AIS in enhancing inspection effectiveness and practicality. Our open-source implementation and additional results are provided on GitHub*.**

I. Introduction

Maintaining civil infrastructure is vital for safety and sustainability, yet aging assets such as culverts pose significant failure risks. With infrastructure challenges projected to cost $79 trillion by 2040 [1], cost-effective maintenance is increasingly prioritized over replacement, driving interest in automated inspection. While research on assets like bridges, buildings, and roads is extensive [2]–[10], culverts remain under-researched.

Interviews with state agencies reveal significant limitations in current culvert inspection practices. Mississippi Department of Transportation (MDOT) employs teleoperated robots with limited optical zoom and distance-based localization, while Nevada DOT relies on manual, inspector-led entry. These workflows highlight four critical challenges: (1) high accessibility and safety risks seen in Fig. 1, (2) labor-intensive procedures, (3) imprecise defect localization, and (4) reliance on superficial, low-quality visual data.

The existing literature further underscores these limitations. Furthermore, to the best knowledge of the author, most existing work, summarized in Table I, continues to focus on visual (surface-level) assessments while requiring manual control. For example, Serrano et al. [14] utilize an unmanned aerial vehicle (UAV) to inspect culverts from the outside, restricting it to shorter culverts while outputting limited data. Similarly, Rumaksari et al. [13] utilize a mobile robot for semi-autonomous surface inspection based on user-selected waypoints. Other works [11], [16], [17] address only a single aspect of culvert robotics, focusing on platform design, CNN-based defect identification, and SLAM performance.

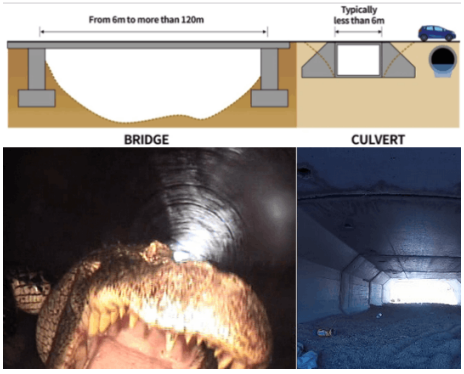


Fig. 1: The top image shows an example of underground culvert. The bottom left shows an alligator that attacked the MDOT inspection robot. The bottom right shows the difficulty of traversing due to sand and mud. The bottom images were collected with/during NDOT inspections.

Autonomous underground inspection can be posed as a multi-object search problem in which a robot must traverse rough environments, find unknown defects, and assess their condition. Unlike object detection and tracking, this is primarily a sequential decision-making problem under uncertainty. Previous works include heuristic, semantic-guided search, and POMDP-based methods [19]–[21], but most focus on single-object search. The multi-object case has been addressed via multi-robot search and object-oriented POMDP formulations [22]–[24].

In this paper, we propose the development of a multi-sensor (MS) autonomous culvert inspection robot. Our contributions can be summarized as follows:

- Development of the MS-AIS robotic platform that integrates vision cameras, LiDAR, RGBD camera and NDE technology for culvert inspection.
- Design of a fully autonomous inspection framework for the MS-AIS platform with three operating modes: (i) Autonomous Inspection System (AIS), an inspection mode that identifies defect areas for data collection, (ii) Full Visual Coverage (FVC), a faster full visual-depth data collection mode, and (iii) Full Multi-sensor

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 under the contract: W912HZ24P0120. 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] are with the USACE Engineer Research and Development Center (ERDC), Information Technology Lab, Vicksburg, MS 39180, USA.

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

Copyright ©2026 IEEE

References	Target	Platform	Sensors	Autonomy	Detection Alg.	Remarks
Ou et al. [11]	N/A	Tracked UGV	Camera	Manual	N/A	Teleoperated camera viewing only; focus on mechanical traversal and adjustable diameter. Limited to 30cm - 45cm pipeline
Miura et al. [12]	N/A	Scott Crawler	Camera	Manual	N/A	Purely teleoperated camera inspection. Laser Range Finder SLAM failed.
Rumaksari et al. [13]	Crack	UGV	Camera	Semi-auto	N/A	Semi-autonomous A*-based robot that moves to user-selected points and records video.
Serrano et al. [14]	Explosives	UAV	PTZ camera	Semi-Auto	N/A	
Meegoda et al. [15]	Crack	UGV	360° camera	Manual	N/A	A manually operated platform records 360° video of the culvert, which inspectors later review to identify defects.
Westcott et al. [16]	Joints	UGV	Camera	Manual	Detectron2	
Dodd et al. [17]	N/A	UGV	Thermal	Manual	N/A	Focused solely on LiDAR–IMU odometry and SLAM evaluation.
Sánchez-Rodríguez et al. [18]	Powerline wires	Vehicle	Mobile Laser scanning system	Manual	CNN	

TABLE I: The notable culvert tunnel inspection methods.

Coverage (FMSC), a full coverage inspection baseline.
- A multi-modal inspection fusion containing SLAM-based 3D reconstruction, defect area characteristics (location, visual, and surface area) for severity evaluation.

The remainder of this paper is organized as follows. Sec. II presents the robot design, Sec. III the SLAM system, Sec. IV the autonomous inspection formulation, Sec. V the experiments and results, and Sec. VI the conclusions and future work.

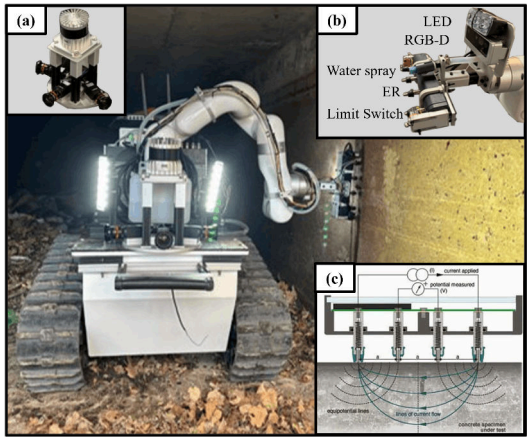


Fig. 2: The complete system design. (a) The perception module; (b) The components of the inspection module end-effector; (c) ER sensor operation.

II. Culvert Inspection Robot Design

The robot design is shown in Fig. 2. We use a tracked platform [25] for traction, and resistance to water, dust, and debris. The onboard computer is an IVX-1000 industrial PC with an Intel® Core™ i9-13900E (13th Gen) CPU, 24 cores, and a 5.2 GHz clock speed, which supports real-time perception and decision making.

The perception module (Fig. 2 (a)) integrates Triton industrial cameras and an Ouster LiDAR. LED illumination enables cameras to capture RGB images to identify surface defects, such as cracks and spalls, which are then fused with LiDAR point clouds for localization and mapping.

As shown in Fig. 2(b), the inspection module features an RGBD camera and an electrical resistivity (ER) sensor [26] on a 0.75m-reach 5-DOF robotic arm. The ER sensor measures deterioration, providing NDE data that complements visual-depth information. To ensure reliable ER measurements, the end-effector includes a water spray, while an ultrasonic sensor and limit switch enable autonomous, precise probe contact. The working principle of the Resipod ER sensor is illustrated in Fig. 2(c).

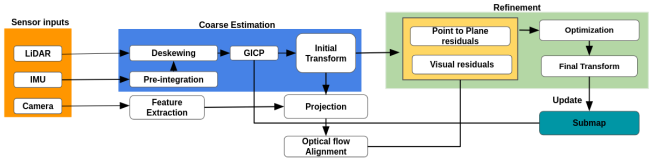


Fig. 3: The overview SLAM system

III. Robotic SLAM System

A. Continuous-Time Kinematic Propagation

The IMU serves as the body frame $\mathcal{B}$, with the initial frame designated as the global frame $\mathcal{G}$. All extrinsic transformations (e.g., $T_{\mathcal{B}}^{\mathcal{L}}$ and $T_{\mathcal{L}}^{\mathcal{C}}$) are pre-calibrated and assumed to be rigid and time-invariant. The overall data processing pipeline of the proposed SLAM framework is illustrated in Fig. 3. To estimate motion and deskew the incoming LiDAR point cloud at timestamp t_k, high-frequency IMU measurements are integrated over the interval $[t_k, t_k + \sum_{n=1}^{M} \Delta t_n]$.

To ensure a physically grounded trajectory during high-dynamic maneuvers, point poses are propagated via a third-order Taylor expansion. By explicitly incorporating linear jerk $\hat{j}$ and angular acceleration $\hat{\alpha}$, this formulation circumvents the fidelity limitations inherent in standard constant-

velocity or constant-acceleration kinematics assumptions:

$$\begin{aligned}
\hat{p}_{n+1} &= \hat{p}_n + \hat{v}_n \Delta t_{n+1} + \frac{1}{2}\hat{R}(\hat{q}_n)\hat{a}_n \Delta t_{n+1}^2 + \frac{1}{6}\hat{j}_{n+1}\Delta t_{n+1}^3 \\
\hat{q}_{n+1} &= \hat{q}_n + \frac{1}{2}(\hat{q}_n \otimes \hat{\omega}_n)\Delta t_{n+1} + \frac{1}{4}(\hat{q}_n \otimes \hat{\alpha}_{n+1})\Delta t_{n+1}^2 \\
\hat{v}_{n+1} &= \hat{v}_n + \hat{R}(\hat{q}_n)\hat{a}_n \Delta t_{n+1} \\
\hat{j}_{n+1} &= \frac{1}{\Delta t_{n+1}}(\hat{R}(\hat{q}_{n+1})\hat{a}_{n+1} - \hat{R}(\hat{q}_n)\hat{a}_n) \\
\hat{\alpha}_{n+1} &= \frac{1}{\Delta t_{n+1}}(\hat{\omega}_{n+1} - \hat{\omega}_n)
\end{aligned} \tag{1}$$

B. Coarse Inertial-Geometric Registration

Following IMU-based point cloud de-skewing, scan-to-map registration is performed via Generalized Iterative Closest Point (GICP) to generate a coarse pose estimate. Let $\hat{\mathcal{P}}_{t_k}^{\mathcal{G}}$ and $\mathcal{S}_{t_k}^{\mathcal{G}}$ denote the deskewed cloud and local submap, respectively. The relative transformation $\Delta\hat{T}_{t_k}^{\mathcal{G}}$ is estimated by minimizing the probabilistic surface distances:

$$\Delta\hat{T}_{t_k}^{\mathcal{G}} = \arg\min_{\Delta\hat{T}_{t_k}} \sum_{i=1}^{N} d_i^T (\Sigma_{\mathcal{S}} + \Delta\hat{T}_{t_k}^{\mathcal{G}} \Sigma_{\mathcal{P}} \Delta\hat{T}_{t_k}^{\mathcal{G}T})^{-1} d_i \tag{2}$$

$$d_i = s_i - \Delta\hat{T}_{t_k}^{\mathcal{G}} p_i, \quad p_i \in \hat{\mathcal{P}}_{t_k}^{\mathcal{G}}, \quad s_i \in \mathcal{S}_{t_k}^{\mathcal{G}} \tag{3}$$

where N denotes the cardinality of the correspondence set, and $\Sigma_{\mathcal{P}}$, $\Sigma_{\mathcal{S}}$ represent the estimated local covariance matrices of the scan and submap points, respectively. The coarse transformation $\Delta\hat{T}_{t_k}^{\mathcal{G}}$ seeds the subsequent joint optimization back-end as a maximum-likelihood prior.

C. Multi-Metric Pose Optimization

1) Visual-Inertial Temporal Residual: Conditioned on the coarse geometric prior, a deskewed global map point $p^{\mathcal{G}}$ is projected onto the previous image I_{k-1} using the camera projection model π:

$$\pi(T_{t_k}^{\mathcal{G}}, p^{\mathcal{G}}) = K(R_{\mathcal{L}}^{\mathcal{C}}(R_{\mathcal{G}}^{\mathcal{L}})p^{\mathcal{G}} - R_{\mathcal{L}}^{\mathcal{C}}(R_{\mathcal{G}}^{\mathcal{L}})^T p_{\mathcal{G}}^{\mathcal{L}} + p_{\mathcal{L}}^{\mathcal{C}}) \tag{4}$$

where K is the intrinsic matrix, and $(R_{\mathcal{L}}^{\mathcal{C}}, p_{\mathcal{L}}^{\mathcal{C}})$ is the LiDAR-camera extrinsic calibration. Valid features are tracked via Shi-Tomasi and optical flow into the current image I_k, yielding the photometric residual r_{fl}:

$$\begin{aligned}
r_{fl}(T_{t_k}^{\mathcal{G}}, p^{\mathcal{G}}) = \|\pi(T_{t_{k-1}}^{\mathcal{G}}, p^{\mathcal{G}}) - s_{k-1} - \\
\pi(T_{t_{k-1}}^{\mathcal{G}} \Delta\hat{T}_{t_k}^{\mathcal{G}}, p^{\mathcal{G}}) + s_k\|^2
\end{aligned} \tag{5}$$

To mitigate inconsistent tracking in challenging environments, optical flow residuals are dynamically thresholded (2–5 pixels) based on the estimated motion aggressiveness, enforcing stricter geometric consistency during smooth trajectories.

2) Geometric Scan-to-Map Residual: To ensure global convergence and refine the coarse estimate, we minimize a point-to-plane geometric residual r_{pl} against the local surface manifold:

$$r_{pl}(T_{t_k}^{\mathcal{G}}, p^{\mathcal{G}}) = \left(n^T (T_{\mathcal{G}_{t_k}}^{\mathcal{L}} T_{t_k}^{\mathcal{G}} p^{\mathcal{L}} - p^{\mathcal{G}})\right)^2 \tag{6}$$

The surface normal n is defined by the target point's three nearest map neighbors $(p_1^{\mathcal{G}}, p_2^{\mathcal{G}}, p_3^{\mathcal{G}})$:

$$n = \frac{(p_2^{\mathcal{G}} - p_1^{\mathcal{G}}) \times (p_3^{\mathcal{G}} - p_1^{\mathcal{G}})}{\|(p_2^{\mathcal{G}} - p_1^{\mathcal{G}}) \times (p_3^{\mathcal{G}} - p_1^{\mathcal{G}})\|} \tag{7}$$

3) Joint Optimization: The state estimation is finalized via a multi-metric joint optimization that minimizes a weighted sum of the photometric and geometric residuals:

$$\min_{T_{t_k}^{\mathcal{G}}} \left\{ \omega_{fl} \sum_{i \in \mathcal{F}} r_{fl}(T_{t_k}^{\mathcal{G}}, p_i^{\mathcal{G}}) + \omega_{pl} \sum_{j \in \mathcal{H}} r_{pl}(T_{t_k}^{\mathcal{G}}, p_j^{\mathcal{G}}) \right\} \tag{8}$$

where scalar weights ω balance the pixel-space photometric constraints ($\mathcal{F}$) and Euclidean-space geometric constraints ($\mathcal{H}$). We solve this nonlinear least-squares problem using the Levenberg-Marquardt algorithm. Linearizing the residuals via first-order Taylor expansion yields a sparse system of normal equations, which is efficiently solved using sparse Cholesky decomposition to facilitate real-time, numerically stable pose updates.

IV. Autonomous Inspection System (AIS)

A. Problem Formulation

We formulate autonomous culvert inspection as a belief-space search over candidate defect areas, defined by states S, actions A, transition models T, observations O, and observation models Ω. At each timestep, the robot takes an action $a \in A$, transitioning from s to s' via $T(s', a, s) = Pr(s'|s, a)$ and receives an observation $o \in O$ via $\Omega(o, s', a) = Pr(o|s', a)$. Under partial observability, we maintain a belief $b_t : S \to [0, 1]$, updated by the recursive Bayesian filter $b'(s') = \eta Pr(o|s', a) \sum_{s \in S} Pr(s'|s, a)\, b(s)$, where η is a normalizing constant.

B. AIS Definition

The robot searches the environment to locate n unknown static defect areas $DEF \subseteq F$, where F is the discrete set of candidate defect locations, and collect close-up data using the *inspection module*. The centerline search points are denoted C. We define AIS as follows:

- S: Our state S includes robot state, $s_r \in S_{robot}$ and defect state $s_d \in F$. The robot's state is defined as $s_r = [p_r, \theta_r]$ for its position and heading in the grid map. The defect states is denoted as $s_d = (s_{d,1}, ..., s_{d,n})$ and $s_{d,i} = [p, \theta]$ containing the pose of the defect area. The belief of defect states is factorized as $b = \prod_{i=1}^{n} b_i$.
- A: The robot has four actions: $Move$, $Declare$, $Coverage$, and $Exit$. $Move$ navigates the robot toward a defect area f, while $Coverage$ explores the environment for potential defects. $Declare$ deploys the *inspection module* via Alg. 1, and $Exit$ terminates the inspection process. The ER arm is deployed only when a defect is within a safe reachable range:

$$A = \begin{cases} true & \|\text{def} - s_r\| \le \alpha l_{arm}, \\ false & \text{otherwise}, \end{cases} \tag{9}$$

where l_{arm} is the arm length and α is an empirically chosen safety margin that avoids unreliable full-reach operation.

- O and Ω are defined in Sec. IV-C
- T: All actions are deterministic, as defects are assumed to be static and robot's motion is highly reliable due to traction (Sec. II) and stable SLAM (Sec. III).

C. Perception Sensor Fusion Model

The robot receives $o_d = \{o_{d,1}, \ldots, o_{d,n}\} \in V$ within sensing region V by fusing three cameras and a LiDAR; if no defects are detected, $o_d = \varnothing$. To obtain o_d, the YOLOv8 [27] instance segmentation is applied to each image I_i to generate masks $\mathcal{M}_i = [m_i, cls_i, pr_i]$ with $m \in \{0,1\}^{H \times W}$ marks defect pixels, cls_i is the class, and pr_i is the confidence. LiDAR points ${}^{\mathcal{L}}\mathbf{P}$ are transformed into the i-th camera frame C_i and projected onto the image plane using intrinsics K_i:

$$ {}^{C_i}\mathbf{P} = {}^{C_i}\mathbf{T}_{\mathcal{L}}\, {}^{\mathcal{L}}\mathbf{P}, \tag{10} $$

$$ \tilde{\mathbf{u}}_i = K_i\, {}^{C_i}\mathbf{P}, \quad \mathbf{u}_i = \begin{bmatrix} \tilde{\mathbf{u}}_{i,x}/\tilde{\mathbf{u}}_{i,z} \\ \tilde{\mathbf{u}}_{i,y}/\tilde{\mathbf{u}}_{i,z} \end{bmatrix}. \tag{11} $$

Next, for each mask rc_i, we retain only the LiDAR points ${}^{C_i}\mathbf{P}^{rc_i}$ whose projections fall within the masked pixels:

$$ rc_i = \{(u,v) | m_i(u,v) = 1\}, \tag{12} $$

$$ {}^{C_i}\mathbf{P}^{rc_i} = \{{}^{C_i}\mathbf{P} | \mathbf{u}_i \in rc_i\}; \quad {}^{C_i}\mathbf{P}^{rc_i} \subseteq {}^{C_i}\mathbf{P} \tag{13} $$

Finally, these masked points are transformed into the global frame and appended to the observation set:

$$ {}^{G}\mathbf{P}^{rc_i} = {}^{G}\mathbf{T}_{C_i}\, {}^{C_i}\mathbf{P}^{rc_i}, \tag{14} $$

where ${}^{G}\mathbf{P}^{rc_i} = \{\mathbf{p}\}$ is the mask pointcloud observation of camera i and $\mathbf{p} = [x, y, z, cls, pr]$. Since each ${}^{G}\mathbf{P}^{rc_i}$ is defined per camera, the fused point cloud is $\mathbf{P}_{fuse} = \bigcup_{i=1}^{3} {}^{G}\mathbf{P}^{rc_i}$. When multiple cameras observe the same defect, overlap is determined by ι:

$$ N_{\text{overlap}} = |\{\mathbf{p}_a \in \mathbf{P}_a : \exists\, \mathbf{p}_b \in \mathbf{P}_b,\ \|\mathbf{p}_a - \mathbf{p}_b\| < \iota\}| \tag{15} $$

If $N_{\text{overlap}} > 5$ and $cls_a = cls_b$, the confidence $pr_{\text{fuse}} = 0.5\, pr_a + 0.5\, pr_b$, ensuring consistent multi-cam observations. For each $f_i \in F$, we collect the subset of points

$$ \mathcal{N}(f_i) = \{\, \mathbf{p} \in \mathbf{P}_{\text{fuse}} \ \mid\ \|\mathbf{p}^{xyz} - f_i^{xyz}\| < \delta \,\}, \tag{16} $$

where $\mathbf{p}^{xyz}$ denotes the spatial coordinates of point $\mathbf{p}$ and δ is the width of f. A fused observation $o_{d,i}$ is only defined if $\mathcal{N}(f_i) \neq \emptyset$. In that case, we set $o_{d,i} = \left[\mathbf{p}_\mu,\ cls,\ pr_\mu,\ \urcorner\right]$, where $\mathbf{p}_\mu = \frac{1}{|\mathcal{N}(f_i)|} \sum_{\mathbf{p} \in \mathcal{N}(f_i)} \mathbf{p}^{xyz}$, $\urcorner = |\mathcal{N}(f_i)|$, the class label cls is assumed identical for all, $pr_\mu = \frac{1}{|\mathcal{N}(f_i)|} \sum_{\mathbf{p} \in \mathcal{N}(f_i)} pr$. Finally, the observation probability is defined as $\Omega(o|s,a) = Pr(o|s,a), \quad Pr(o_d|s,a) = pr\, YD$, where YD^i is the detection model and pr is the confidence score. Our detection model $YD(o_d, s_{d,i}, s_r)$ is defined as

$$ = \begin{cases} 1 - \text{TP} & o_{d,i} = \text{null} \\ \phi G(o_{d,i} | \mu, J) & \|o_{d,i}^{\mathbf{p}_\mu} - s_{d,i}\| \leq 0.3\delta \\ \frac{\phi}{\|FOV\|} & \|o_{d,i}^{\mathbf{p}_\mu} - s_{d,i}\| > 0.3\delta, \end{cases} \tag{17} $$

where δ is the width of $f \in F$, $J = \mathbf{I}^{3\times3}\sigma^2$, $G(\cdot)$ denotes the Gaussian distribution, true positive, TP $= 0.88$ (validation). This assigns a high likelihood within 0.3δ, while maintaining a low likelihood beyond this range to preserve weak evidence of potential defects. To account for the accuracy with distance and number of points:

$$ \phi = \frac{e^{\psi \urcorner}}{\log(\|o_{d,i}^{\mathbf{p}_\mu} - s_{d,i}\| + \xi + 1)} \tag{18} $$

where ψ = 0.5 is the scaling factor and ξ = 0.001 ensures numerical stability

Algorithm 1 : InspectionArmDeployment($\mathcal{W}$)

$\mathcal{W} = [\mathbf{p}, \mathbf{q}]$
$\mathbf{P}$ = armInspectionPoints($\mathbf{p}, \mathbf{q}, d_p, d_s, d_i, d_f$) ▷ from Alg. 2
Convert quaternion $\mathbf{q}$ to Euler angles $\mathbf{rpy} = (r, p, y)$
$p = p - 1.57$ *// Adjust pitch to ensure the target X-axis and arm Z-axis are perpendicular*
Convert $\mathbf{rpy}$ back to $\mathbf{q}_{goal}$
PlanArmMotion($\mathbf{P}[-1]$, $\mathbf{q}_{goal}$)
Capture RGBD images
PlanArmMotion($\mathbf{P}[-2]$, $\mathbf{q}_{goal}$)
Spray water for 2 sec and wait for 15 sec
for each $waypoint$ in $\mathbf{P}[:, :-2]$ (reverse order) **do**
  if debounce(lim_switch) is **False then**
    break
  else
    if $ultra < 0.07$ m **then**
      if ER_data is $Valid$ **then**
        Save(ER_data, $\mathcal{W}$)
        return
      else
        PlanArmMotion($waypoint$, $\mathbf{q}_{goal}$)
      end if
    else
      PlanArmMotion($waypoint$, $\mathbf{q}_{goal}$)
    end if
  end if
end for
for attempt = 1 to 5 **do**
  if valid ER reading received **then**
    Save(ER_data, $\mathcal{W}$)
    return
  else
    Save(-1, $\mathcal{W}$)
  end if
end for

D. Inspection Module Deployment

Alg. 1 describes the *inspection module* deployment for a defect area $\mathcal{W} = [\mathbf{p}, \mathbf{q}]$. The arm executes two sequential modes: *visual-depth*, capturing RGBD data, and *condition*, which presses a contact-based ER sensor against the culvert surface to measure resistivity. Ultrasonic sensing, ER readings, and a limit switch jointly verify proper wall contact.

Algorithm 2 : armInspectionPoints($\mathbf{p}$, $\mathbf{q}$, d_p, d_s, d_i, d_f)

$\mathbf{v} = [1 - 2(\mathbf{q}_y^2 + \mathbf{q}_z^2)\ 2(\mathbf{q}_x\mathbf{q}_y + \mathbf{q}_w\mathbf{q}_z)\ 2(\mathbf{q}_x\mathbf{q}_z - \mathbf{q}_w\mathbf{q}_y)]^T$
$\mathbf{v} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$
$\mathbf{p}_s = \mathbf{p}_0 + d_s\mathbf{v}$ *// Compute contact point*
$\mathbf{p}_e = \mathbf{p}_0 + d_i\mathbf{v}$ *// Compute spray point*
$D = \|\mathbf{p}_e - \mathbf{p}_s\|$ *// Compute the total distance*
$N = \lfloor D/d_p \rfloor + 1$ *// Compute the number of points*
$\mathbf{t} = \text{linspace}(0, 1, N)$ *// Generate interpolation weights*
$\mathbf{P} = \mathbf{p}_s + \mathbf{t}(\mathbf{p}_e - \mathbf{p}_s)$ *// Generate waypoints*
$\mathbf{p}_f = \mathbf{p} + d_f\mathbf{v}$ *// Compute RGB point*
$\mathbf{P} = \mathbf{P} \cup \mathbf{p}_f$
Return $\mathbf{P}$

Waypoints for the two inspection modes are generated using Alg. 2, where $\mathbf{p}$ and $\mathbf{q}$ define the defect area pose, $\mathbf{v}$ is the surface-normal unit vector computed from $\mathbf{q}$, d_p is the waypoint spacing, d_s is the ER contact distance, d_i is the water-spray standoff distance, and d_f is the visual inspection

distance. Using $\mathbf{v}$, the algorithm computes the ER contact point ($\mathbf{p}_s$), spray point ($\mathbf{p}_e$), and visual inspection point ($\mathbf{p}_f$), then interpolates intermediate waypoints between $\mathbf{p}_s$ and $\mathbf{p}_e$ at spacing d_p for smooth arm motion.

Algorithm 3 Autonomous Inspection Planner

```
1: function PLANNER(s, b_t)
2:     s = (s_r, s_d)
3:     if b_t is empty or all(b^i ∈ b_t) is visited then
4:         if all(c ∈ C) is visited then
5:             return Exit
6:         else
7:             return Coverage
8:         end if
9:     else
10:        OOB ← SAMPLE(b_t)
11:        temp_goal ← nearest_def(s_r, OOB)
12:        if ‖s_r − temp_goal‖ < α · l_arm then
13:            return Declare
14:        else
15:            return Move
16:        end if
17:    end if
18: end function
19: function SAMPLE(b) [24]
20:    for d ∈ DEF do
21:        ŝ_d ∼ b
22:    end for
23:    return ∪ŝ_d
24: end function
25: function COVERAGE(s_r)
26:    temp_goal ← closest_unvisited(c ∈ C)
27:    neighbors ← neighbor(s_r)
28:    h ← ‖neighbors − temp_goal‖
29:    g ← ‖neighbors − s_r‖
30:    f ← g + h
31:    neighbor_goal ← argmin(f)
32:    goTo(neighbor_goal)
33: end function
34: function MOVE(s_r, def_goal)
35:    p, q ⊆ def_goal
36:    θ = quaternion_to_yaw(q)
37:    goal = [x, y]^T + dist[cos(θ), sin(θ)]^T  x, y ∈ p
38:    goTo(goal)
39: end function
```

After the waypoints are generated in Line 2 of Alg. 1, the arm sequentially executes them using **PlanArmMotion**$(\mathbf{p}, \mathbf{q})$ denotes a generic arm motion-planning call; our implementation uses MoveIt [28], but the algorithm is planner-agnostic. Lines 3–7 correspond to the *visual-depth* inspection, moving to the waypoint $\mathbf{P}[-1]$ to capture RGBD images data, which is processed via the geometric estimation pipeline (Sec. IV-F) to quantify defect surface area. Lines 8–9 moves to the waypoint $\mathbf{P}[-2]$ and activate the water spray to wet the surface for reliable ER contact measurements. Lines 10–25 describe the *condition* inspection. The arm executes the remaining waypoints in reverse, while using limit-switch, ultrasonic, and ER feedback to verify surface contact. The limit switch reports $lim_switch = \mathbf{False}$ when pressed ($\mathbf{True}$ otherwise), and contact is confirmed via debouncing with five consecutive $\mathbf{False}$ samples. Lines 26–33 validate and log ER measurements. Without confirmed contact, the ER sensor maintains its previous value. For robustness, we compute $\mu_{ER} = \frac{1}{N}\sum_{i=1}^{N} ER_i$ and $\sigma_{ER} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(ER_i - \mu_{ER})^2}$ from N consecutive samples, set $ER_data = \mu_{ER}$, and mark it *Valid* only if $\sigma_{ER} > 0$.

E. Planner

To solve the formulation and select high-level actions, a planner was developed as outlined in Alg. 3. PLANNER$(\cdot)$ outputs one of four actions mentioned in Section IV-B. *Coverage* calls COVERAGE$(\cdot)$ to explore along C, and *Move* calls MOVE$(\cdot)$ to navigate to a defect goal, generated from the estimated defect pose in line 37 to satisfy the condition for Eq. 9. When robot reaches the goal, the planner returns *Declare* (Alg. 1). The planner is embedded in the AIS closed-loop execution (Alg. 4): at each timestep t, the robot executes the selected action a_t, acquires a fused observation o_t (Section IV-C), updates the belief b^{t+1} (Section IV-A).

Algorithm 4 : Autonomous Inspection System (AIS)

```
1: b^1 ← Initial Belief
2: s^1 ← Initial State
3: t = 1
4: while t < T_max do
5:     a^t ← PLANNER(s^t, b^t)                ▷ from Alg. 3
6:     if a^t is Exit then
7:         break
8:     end if
9:     o^t ← execute action a^t, receive observation
10:    b^{t+1} ← η Pr(o^t|s^t, a^t) Σ_{s∈S} Pr(s^{t+1}|s^t, a^t) b^t
11:    t ← t + 1
12: end while
```

F. Defect Geometry Pipeline

Defect surface area estimation is performed by first applying YOLO-based instance segmentation to the RGB image to obtain a binary defect mask $M(u, v)$. Let $d(u, v)$ denote the corresponding depth image. The valid depth pixels inside the mask are then back-projected into a 3D defect point set $\mathcal{P}$ using the camera intrinsic matrix K: $\mathbf{p}(u, v) = d(u, v)\, K^{-1}[u,\ v,\ 1]^T$. A best-fit plane is estimated from $\mathcal{P}$, with centroid $\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{p}_i$, and plane equation $\mathbf{n}^T(\mathbf{p} - \boldsymbol{\mu}) = 0$, where $\mathbf{n}$ is the plane normal. The 3D defect points are then projected onto the plane using two in-plane orthonormal basis vectors $\mathbf{e}_1, \mathbf{e}_2$:

$$x_i = \mathbf{e}_1^T(\mathbf{p}_i - \boldsymbol{\mu}), \qquad y_i = \mathbf{e}_2^T(\mathbf{p}_i - \boldsymbol{\mu}).$$

This yields a 2D point set $\mathcal{Q} = \{[x_i, y_i]^T\}$, from which a boundary polygon B is formed and the defect area is computed as $A = \mathrm{Area}(B)$.

V. EXPERIMENT

A. SLAM Performance

We evaluate our SLAM system on the public dataset [29], against the state-of-the-art LiDAR-Inertial Odometry (LIO) baselines: LIO-SAM [30], DLIO [31], and FAST-LIO2 [32]. Performance is quantified using the Absolute Trajectory Error (ATE), reported as Mean error and Root Mean Square Error (RMSE) to capture overall drift and worst-case deviations. As summarized in Table II, our framework achieves the most consistent stability across the evaluated sequences. Although baseline methods provide adequate accuracy in constrained, feature-rich outdoor areas (e.g., the *gate* sequences), their performance degrades significantly when pushed to environmental extremes. For instance, in the expansive outdoor *street_02* sequence, conventional geometric registration algorithms suffer from scale ambiguity, causing the estimated trajectory to accumulate substantial drift (up to 4.58 m RMSE). Conversely, our framework exhibits equivalent resilience in feature-deprived, perceptually degraded indoor environments. In the narrow, structurally repetitive *hall* sequences, which emulate the geometric degeneracy

Dataset	gate_01		gate_02		gate_03		street_01		street_02		street_03		hall_02		hall_03	
	Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE
LIO-SAM	0.179	0.238	0.270	0.330	0.102	**0.112**	1.732	1.970	3.707	4.580	0.316	0.525	0.329	0.355	0.387	0.416
DLIO	0.091	0.116	0.237	0.300	0.108	0.123	0.344	0.373	3.100	3.971	0.129	0.140	0.322	0.354	0.371	0.404
FAST-LIO2	0.163	0.176	0.274	0.325	0.208	0.218	**0.278**	**0.302**	2.692	2.858	0.170	0.181	0.505	0.530	0.588	0.635
OURS	**0.090**	**0.111**	**0.233**	**0.295**	**0.099**	0.114	0.279	0.308	**2.411**	**2.618**	**0.126**	**0.136**	**0.290**	**0.308**	**0.353**	**0.404**

TABLE II: **Absolute trajectory error (m)**. The best results are marked in bold.

characteristic of underground tunnels and culverts, representative methods such as FAST-LIO2 degrade severely, yielding a tracking error of up to 0.635 m RMSE in *hall_03*. In contrast, our multi-metric optimization paradigm effectively constrains drift across both unconstrained outdoor and highly constrained indoor settings, achieving superior trajectory accuracy across the majority of benchmark sequences. This domain-agnostic, drift-resilient state estimation guarantees the high-precision localization fidelity required for downstream tasks such as autonomous inspection and structural defect mapping.

While the public benchmarks quantitatively validate our localization accuracy, our proprietary culvert dataset – collected in a completely GPS-denied environment lacking reliable ground-truth positioning – is evaluated qualitatively via dense 3D point cloud colorization. Because returning to an exact initial pose for closed-loop evaluation was impractical in this operational deployment, the structural and texturing fidelity of the generated map serves as a direct proxy for odometric consistency. Precise extrinsic calibration combined with stable trajectory estimation yields seamless color projections; any uncompensated registration errors or spatial drift would manifest as distinct reprojection artifacts, such as color bleeding or ghosting. As illustrated in Figs. 7(b) and 8(b), the high-fidelity colorized reconstructions demonstrate precise textural correspondence with the raw camera frames shown in Figs. 7(c–d) and 8(d–e).

B. Autonomous System

Experimental validation was conducted across simulated environments and field tests. The simulated domain, implemented in Gazebo, includes: (i) 5DF ($1.2 \times 4.0 \times 0.7$ m), with five defects over $n = 8$ defect areas, and (ii) 10DF ($1.2 \times 8 \times 0.7$ m), with ten defects over $n = 18$. Field evaluation was performed in two distinct settings: an artificial ($2.4 \times 3.4 \times 1.0$ m) culvert constructed of heterogeneous concrete blocks ($n = 9$), and a natural culvert ($2.3 \times 3.0 \times 1.0$ m) to assess performance in unstructured settings ($n = 6$). This provides both controlled and natural test settings. The system utilized an image resolution of 2.8 MP. Parameters were set as: $d_p = 0.04$ m, $d_s = 0.19$ m, $d_i = 0.39$ m, $d_f = 0.45$ m, $l_{arm} = 0.75$ m, $\delta = 0.348$ m, $\alpha = 0.9$, and $\iota = 0.01$ m. F is generated via a 3D uniform sampling of the culvert volume, and C is the centerline within F.

Given the unique challenges of robotic culvert inspection, we conduct a comparative analysis of four approaches: AIS, FVC, FMSC, MOS, a multi-object search POMDP inspired by [23] from [33]. To compare planning strategies on a common basis, we score all methods with a task reward and its discounted cumulative form (DCR). A correct $Declare$ at an unvisited defect area yields $R_{\max} = 1000$, an incorrect one $R_{\min} = -1000$, and every other non-$Exit$ action incurs a step cost of -2; DCR applies a discount $\gamma = 0.99$. FVC, a speed benchmark, executes the *visual-depth* mode at every $f \in F$; FMSC adds the *condition* mode as a practical worst-case baseline. Since both inspect every f, their DCR is inherently very low, so we omit it and use them mainly as references for completeness and execution time. MOS serves as a comparative simulation baseline; however, its 2D formulation lacks the dimensionality required for 3D field tests. Its *find* action is mapped to our $Declare$ action. Each method was run 20 times per environment, and key metrics, including duration, discounted cumulative reward (DCR), and area-normalized reward (ANR), were recorded. ANR for all runs, l, is defined as $\text{ANR} = \frac{\sum_{i=1}^{l} \text{ACR}}{n R_{\max}}$, where ACR is the average cumulative reward. Unlike raw DCR, ANR normalizes performance across domains with different numbers of defect areas n.

Our defect dataset, collected with the Nevada DOT from four culverts and online sources, contains 1,862 augmented images split 80%-10%-10% for training, validation, and test, achieving a TP of 0.88.

1) Simulation: We evaluated navigation efficiency in Gazebo using 5DF ($\|F\| = 52$) and 10DF ($\|F\| = 100$) environments. MOS was configured with $\gamma^{depth} = 25$, exploration constant $c = 2000$.

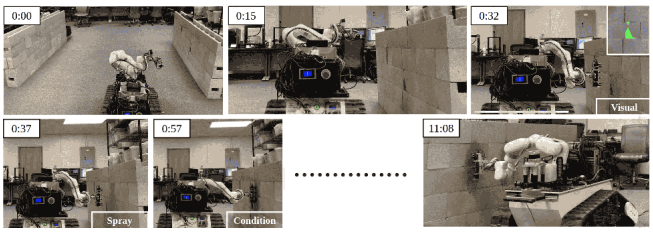


Fig. 4: In the artificial culvert, the robot begins at 0:15 by going to the defect area, captures RGB-D data at 0:32, sprays water at 0:37, and deploys the ER sensor; this repeats until 11:08. The roof was removed for visualization.

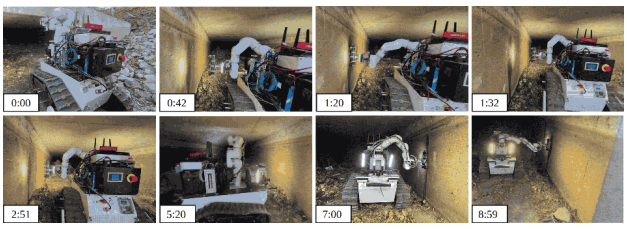


Fig. 5: The natural culvert experiment.

As shown in Table III, AIS is consistently the most efficient method, requiring the shortest duration in all settings and achieving the highest DCR and ANR with the lowest standard deviations. Notably, it exceeds MOS despite not

Method	5DF			10DF			Artificial		Natural	
	Time(s)	DCR	ANR	Time(s)	DCR	ANR	Time(s)	DCR	Time(s)	DCR
MOS	251(4.88)	5759(1.07)	0.986	2678(335)	7651(15.85)	0.964	N/A	N/A	N/A	N/A
FMSC	N/A	N/A	N/A	N/A	N/A	N/A	1456(4.44)	N/A	1477(6.55)	N/A
FVC	327(1.94)	N/A	N/A	647(2.17)	N/A	N/A	**462**(4.8)	N/A	**478**(5.64)	N/A
AIS	**195**(1.82)	**7291**(0.78)	**0.998**	**417**(3.17)	**14935**(0.88)	**0.998**	668(2.28)	5613(0.35)	561(3.25)	8236(0.56)

TABLE III: **Quantitative Results**. DCR is not reported for FMSC and FVC because both $Declare$ every $f \in F$, yielding large negative rewards. In the 5DF/10DF simulated environment, the *inspection module* is not modeled, so FMSC is runtime-equivalent to FVC and marked N/A. Results are averaged over 20 runs, with standard deviations in parentheses; bold indicates the best result. AIS achieves the best or second-best performance across all metrics.

optimizing the reward directly; while MOS performs well in small environments, its MCTS planning cost grows sharply with size, and in 10DF its tendency to revisit regions makes it slower than even the FVC benchmark.

2) Field Test: Real life evaluations were conducted in an artificial culvert ($\|F\| = 24$) and a natural culvert ($\|F\| = 24$). Although the natural culvert is longer, heavy debris (Fig. 8(a)) limited the robot to 2.3 m of traversal inside.

Figs. 4 and 5 show the indoor and outdoor inspection action sequences. The robot starts with a uniform defect belief, updates it from observations, and selects $Move$ and $Declare$ to inspect the defect areas. As shown in Table III, AIS is slower than FVC in the field because it executes both *visual-depth* ($\sim$ 15s) and *condition* (40–70s) inspection modes, whereas FVC executes only the shorter visual-depth mode. FMSC is the slowest because it performed both inspection modes in all $f \in F$.

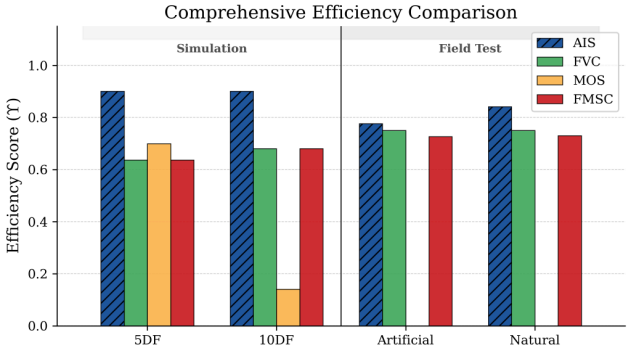


Fig. 6: Normalize efficiency score comparison graph.

C. Efficiency Analysis

To evaluate performance across heterogeneous environments, we evaluate each autonomous mode using a normalized efficiency score $\Upsilon = [w_t(\frac{T_{best}}{T_{method}}) + w_d(\frac{V+C}{2}) + w_c\kappa] \in [0, 1]$ as a weighted balance between time efficiency, diagnostic fidelity, and coverage. $V, C \in \{0, 1\}$ are binary indicators for visual-depth and condition assessment. Weights are domain-specific: in the simulation, where the *inspection module* is not modeled, $(w_t, w_d, w_c) = (0.9, 0.0, 0.1)$; in the field test $(w_t, w_d, w_c) = (0.4, 0.5, 0.1)$.

As seen in Fig. 6, AIS consistently achieves the highest score. In simulation, AIS maximizes Υ by leveraging belief-pruning to skip redundant defect areas, whereas MOS scales poorly as environment complexity increases. In field tests, AIS yields the highest Υ from its selective trigger logic; by prioritizing diagnostic fidelity (V, C) at high-probability defect locations, it avoids the massive temporal overhead of FMSC while still providing the high-fidelity data that FVC lacks.

This analysis reveals that, while FVC is a "speed benchmark," it is specialized for rapid inspection where duration is the primary constraint. However, for regulatory audits and asset management—where high-fidelity evidence of material degradation is needed—AIS represents the Pareto-optimal solution.

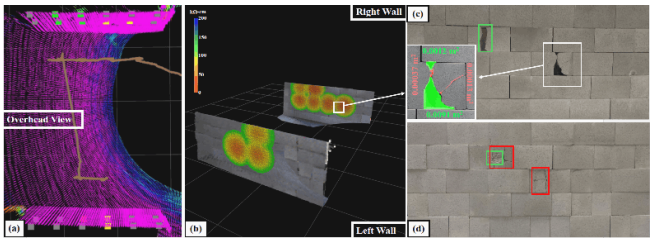


Fig. 7: (a) Robot trajectory. (b) 3D map of the artificial culvert with ER-based condition levels: 0–60 (Poor), 60–90 (Fair), 90–120 (Good), and $\geq$ 120 (Excellent).(c) Right wall (d) left wall of the culvert, showing crack and spall defects.

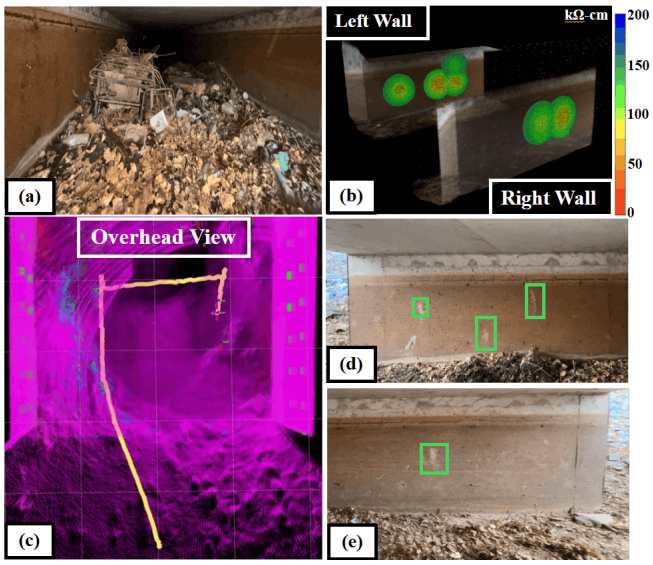


Fig. 8: (a) Culvert interior. (b) 3D map of the natural culvert with ER-based condition levels: 0–60 (Poor), 60–90 (Fair), 90–120 (Good), and $\geq$ 120 (Excellent). (c) Robot trajectory. (d) Left and (e) right culvert walls showing spall defects.

D. Structural Health Assessment

Beyond autonomous navigation, MS-AIS provides visual and condition assessment of defects (Figs. 7 and 8). The *visual-depth* mode of the *inspection module* enables precise quantification of defect surface area within the localized 3D map, providing a metric for structural health monitoring.

As shown in Fig. 7, the indoor culvert contained intentionally degraded sections where severe cracks and spalls aligned with low ER readings, confirming true material deterioration. In contrast, Fig. 8 shows that visually degraded outdoor regions, such as surface spalls, often exhibited fair ER values, indicating mainly superficial damage. Together, these results demonstrate the complementary value of visual and condition inspection: ER confirms structural deterioration when present and helps distinguish cosmetic defects from deeper damage, consistent with prior works [6], [34].

VI. Conclusion and Future Work

This paper introduces a multi-sensor autonomous inspection platform designed for efficient inspection in underground environments. It employs three different inspection modes to ensure robust and comprehensive evaluation. This paper highlights the potential of the culvert 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) it is limited to a single NDE (ER) sensor, necessitating the future integration of a multi-modal NDE—including ground penetrating radar and impact echo; (3) the framework has no obstacle avoidance capability.

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

[3] 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.

[4] 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.

[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] 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.

[7] A. D. Nguyen, C. P. Le, P. Walunj, A. Netchaev, T. N. Do, and H. M. 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, pp. 3334–3339.

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

[10] B. A. Christie, O. Ennasr, and G. P. Glaspell, "Autonomous navigation and mapping in a simulated environment," 2021.

[11] 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.

[12] 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.

[13] 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.

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

[15] J. N. Meegoda, J. A. Kewalramani, and A. Saravanan, "Adapting 360-degree cameras for culvert inspection: case study," *J. of Pipeline Systems Engineering and Practice*, vol. 10, no. 1, p. 05018005, 2019.

[16] J. Westcott and A. Bowman, "Predictive detection of soil cavities in embankments due to culvert defects using centrifuge modeling and image analysis," in *Geo-Extreme 2025*, pp. 155–163.

[17] B. M. Dodd, O. Ennasr, A. Naser, C. C. Ellison, J. D. Ray, G. P. Glaspell, and A. D. Netchaev, "Mapping and localization within a mock sewer system," 2023.

[18] A. Sánchez-Rodríguez, M. Soilán, M. Cabaleiro, and P. Arias, "Automated inspection of railway tunnels' power line using lidar point clouds," *Remote Sensing*, vol. 11, no. 21, p. 2567, 2019.

[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 Intern. Conf. 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] 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 Comp.*, 2002, pp. 83–91.

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

[24] D. Silver and J. Veness, "Monte-carlo planning in large pomdps," *Advances in neural information processing systems*, vol. 23, 2010.

[25] A. Robotics, "Bunker Pro Tracked Mobile Robot," https://global.agilex.ai/products/bunker.

[26] 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.

[27] G. Jocher, J. Qiu, and A. Chaurasia, "Ultralytics YOLO," Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[28] I. A. Sucan and S. Chitta, "MoveIt," https://moveit.ros.org/.

[29] J. Yin, A. Li, T. Li, W. Yu, and D. Zou, "M2dgr: A multi-sensor and multi-scenario slam dataset for ground robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2266–2273, 2021.

[30] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ Intern. Conf. on Intelligent Robots and Sys. (IROS)*. IEEE, 2020, pp. 5135–5142.

[31] K. Chen, B. T. Lopez, A.-a. Agha-mohammadi, and A. Mehta, "Direct lidar odometry: Fast localization with dense point clouds," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2000–2007, 2022.

[32] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[33] K. Zheng and S. Tellex, "pomdp_py: A framework to build and solve pomdp problems," in *ICAPS 2020 Workshop on Planning and Robotics (PlanRob)*, 2020.

[34] 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.