

Nondestructive evaluation sensor fusion with autonomous robotic system for civil infrastructure inspection

Spencer Gibb  | Hung Manh La  | Tuan Le | Luan Nguyen | Ryan Schmid | Huy Pham

Advanced Robotics and Automation Lab,
Department of Computer Science and
Engineering, University of Nevada, Reno,
Nevada

Correspondence

Hung La, Advanced Robotics and Automation
Lab and Department of Computer Science and
Engineering, University of Nevada, Reno, NV
89557.

Email: hla@unr.edu

Funding information

National Science Foundation, Grant/Award
Number: NSF-IIP-I-Corps #1639092; National
Aeronautics and Space Administration,
Grant/Award Number: NNX10AN23H; U.S.
Department of Transportation; Missouri Univer-
sity of Science and Technology

Abstract

Civil infrastructure inspection is crucial to maintaining the quality of that infrastructure, which has a great impact on the economy. Performing this inspection is costly work that requires workers to be trained on how to use varying technologies, which can be error prone when performed manually and can result in damage to the infrastructure in some cases. For this reason, nondestructive evaluation (NDE) sensors are preferred for civil infrastructure inspection as they can perform the necessary inspection without damaging the infrastructure. In this paper, we develop a fully autonomous robotic system capable of real-time data collection and quasi-real-time data processing. The robotic system is equipped with several NDE sensors that allow for a sensor fusion method to be developed that successfully minimizes inspection time while performing adequate inspection of areas that require more in-depth data to be collected. A detailed discussion of the inspection framework developed for this robotic system, and the dual navigation modes for both indoor and outdoor autonomous navigation is presented. The developed robotic system is deployed to inspect several infrastructures (e.g., parking garages, bridges) at and near by the University of Nevada, Reno campus.

KEYWORDS

concrete inspection, field robots, non-destructive inspection

1 | INTRODUCTION

The three infrastructures in the United States that are the most critical to the economic wellbeing of the country are aviation, ports, and roads (ASCE, 2016). Consequently, maintaining civil infrastructure is a necessary part of ensuring healthy economic and social growth in modern day society (La, Dinh, Pham, Ha, & Pham, 2018). It is estimated that deficiencies will cost the United States \$3.9 trillion by the year 2025 (ASCE, 2016), and that the number of concrete highway bridges in the United States with deteriorating surfaces is over 180,000 (FHWA, 2015). Since all civil infrastructure is susceptible to eventual deterioration, and total deficiency if left unmaintained, it is necessary to prevent this deterioration when possible. Unfortunately, by the time that civil infrastructure is visibly damaged as in Figure 1, it can be too late or too costly to fix or repair the damaged infrastructure because of how long the damage has been present. This further illustrates the need for preventative maintenance of civil infrastructure.

Preventing these expenses and deterioration is possible through building new civil infrastructure to eliminate the cost of repairing the previous infrastructure, and through maintaining current civil

infrastructure. It is less costly to properly maintain current civil infrastructure than it is to build new infrastructure, but there is still a need for low-cost inspection and maintenance programs. For this reason, the Federal Highway Administration (FHWA) initiated the Long-Term Bridge Performance (LTBP) program to utilize nondestructive evaluation (NDE) technologies for bridge deck condition assessment (Gucunski, 2015). Many civil infrastructure maintenance programs have since been developed (Frangopol, Lin, & Estes, 1997; Furuta, Kameda, Fukuda, & Frangopol, 2004; Kang & Wen, 2000).

Even though there are a wide variety of uses for NDE sensors through the current infrastructure inspection and maintenance programs, the inspection process is still time consuming and requires workers who are well trained (Solla, Asorey-Cacheda, Nez-Nieto, & Riveiro, 2015; Tajdini & Rappaport, 2013; Wang, Li, Jiang, & Chen, 2013). In Figure 2, an inspection is being performed where multiple inspectors are required to perform the inspection of a single bridge, which results in this kind of manual inspection being cost inefficient. This inefficiency is present in any type of NDE sensor data collection or inspection that requires manual collection, which can be extended to inspection of civil infrastructure other than bridges, like inspection

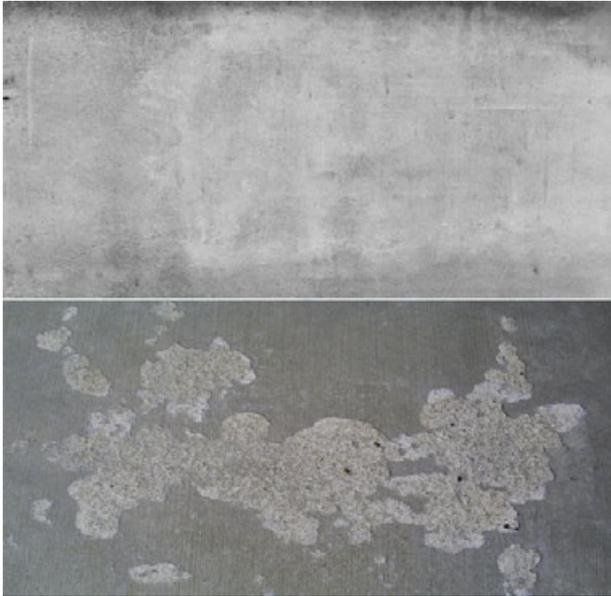


FIGURE 1 An image of concrete that has no spalling (top), and an image of spalling concrete (bottom) where corroded subsurface rebar lead to cracking in the concrete and weakness in the bridge

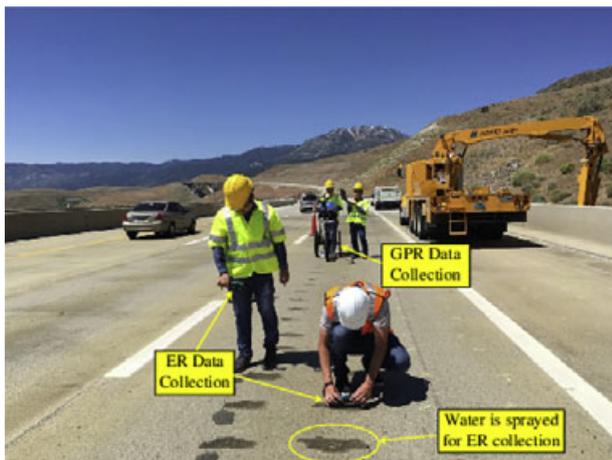


FIGURE 2 Advanced Robotics and Automation (ARA) Lab team members operating NDE sensors for bridge deck inspection on the Pleasant Valley Bridge on Highway 580 from Reno, NV toward Carson City, NV, July 2016

of parking structures and other structures containing reinforced concrete with rebar in it. In addition, manual data collection with NDE sensors can lead to errors in the collection process while workers collect data from different areas and record the location and data. These issues with the manual inspection of civil infrastructure lead to an increase in the need for a fully autonomous robotic system for civil infrastructure inspection (Pham & La, 2016; Pham et al., 2016). Such a robotic system could reduce the required training for inspection workers, reducing safety requirements, and reducing the cost of civil infrastructure inspection. There have been some research efforts related to the development of automated bridge inspection systems such as Robotic Bridge Inspection Tool (RABIT) (Gucunski et al., 2013; La et al., 2013, 2014; Lim, La, Shan, & Sheng, 2011; Lim, La, & Sheng, 2014), NDE data processing (Dinh et al., 2015; Gucunski et al., 2015; Gucunski, Kee,

La, Basily, & Maher, 2015; La et al., 2015), automated inspection with drones (Phung, Quach, Dinh, & Ha, 2017; Scherer, Singh, Chamberlain, & Elgersma, 2008; Yan et al., 2016), and mobile sensor network-based inspection (Zhu, Guo, Cho, Wang, & Lee, 2012). A recent survey of automated civil infrastructure inspection can be found in Li et al. (2013).

Our previous work focused on bridge deck inspection, the nature of that work, and robotic systems capable of performing bridge deck inspection (Gibb & La, 2016; Gibb, Le, La, Schmid, & Berendsen, 2017; Le et al., 2017; La, Gucunski, Dana, & Kee, 2017). Bridge deck inspection is a labor-intensive work and requires workers to be trained on multiple types of sensors and equipment. The inspection process also requires traffic regulation and unsafe working conditions in some cases, which motivated our previous work. Our previous work did not look at other types of infrastructure inspection, and instead only focused on bridge deck inspection. In this work, we focus on a fully autonomous robotic system for general civil infrastructure inspection.

In this paper, we develop a multifunctional, autonomous, civil infrastructure inspection robot that is capable of operating both indoors and outdoors. After participating in the National Science Foundation (NSF) Innovation Corps (I-Corps™) in 2015, we found that inspectors and infrastructure owners mainly wish to minimize any interruption to normal operation of the infrastructure during inspection. For this reason, we develop an inspection framework for our autonomous robotic system that fuses data collected from multiple sensors that are equipped on the system. Our inspection framework is capable of two inspection modes: fast inspection and in-depth inspection. In the fast inspection mode, the developed robotic system intelligently decides which areas need more in-depth inspection and saves time by not inspecting all areas in an in-depth manner. On the other hand, the in-depth inspection mode allows all NDE sensors to be deployed across the entirety of the inspection area. Several NDE sensors are equipped on our robotic system, including ground-penetrating radar (GPR), two electrical resistivity (ER) sensors, and a camera system. The GPR system allows us to use our work in Gibb and La (2016) to locate subsurface steel rebar in the reinforced concrete, the ER sensors allow the corrosion of the concrete to be measured, and the camera system is used to build a crack map from the visual data that is collected. Through implementing our various NDE data processing algorithms on the robotic system's on-board computers, we are able to perform civil infrastructure inspection in quasi-real time, where we can process data just after it is collected in small intervals. In addition to collecting and processing NDE data using the variety of sensors equipped on our robotic system, navigation is also performed by the robotic system. Navigation is accomplished through the use of an on-board inertial measurement unit (IMU), robot's wheel encoders, and a global positioning unit (GPS) receiver. The robotic system also employs a PrimeSense camera, which has been shown to work for elevation mapping and autonomous navigation. As stated previously, the robotic system is capable of indoor and outdoor operation, which is accomplished via two separate navigation modes. The odometry of the wheel encoders, IMU and GPS receiver are used for an extended Kalman filter (EKF)-based navigation mode in outdoor areas. For indoor areas, the stereo camera system and inertial information are used in a visual-inertial odometry-based navigation mode as in Bloesch, Omari, Hutter, and Siegwart (2015). The developed

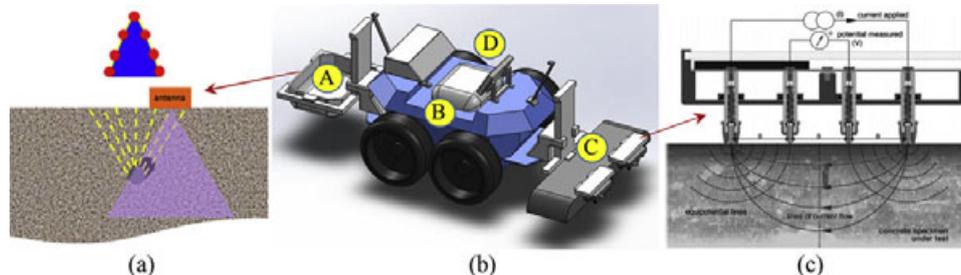


FIGURE 3 A complete system design. (a) How the equipped GPR sensor works; (b) an overall design of the robotic system: A: GPR deployment system, B: Robot main body, C: ER deployment system, D: GPR display/monitor; (c) how the ER sensors work. Electricity is sent through the concrete surface to measure its resistance. The resistance of the surface can be used to gauge the structural health of the concrete

robotic system is also capable of using its navigation modes and NDE data collection capability to perform intelligent inspection through the fast inspection mode, where we propose and develop an NDE sensor fusion technique designed to save time throughout the inspection process.

The remainder of the paper is structured as follows: The design of the robotic system is detailed in Section 2, navigation of the robotic system is discussed in Section 3, the data collection and processing for the GPR sensor is discussed in Section 4, and the data collection and processing for the camera system and the corresponding crack map is discussed in Section 5. Then the sensor fusion process is detailed in Section 6, and concluding discussion and remarks are given in Section 7.

2 | ROBOTIC SYSTEM

In this section, the development and design of the robotic system is described, and short explanations are provided to explain how the equipped NDE sensors work.

2.1 | Seekur Jr mobile robot as a base platform

As the base platform for our robotic system, we used the Seekur Jr mobile robot from Omron Adept Technologies, Inc. This robot base platform comes equipped with skid-steering four-wheel-drive capabilities. This platform is perfect for moving in the zig-zag path that is used for the in-depth inspection mode. An example of the path that the robot would navigate on a bridge deck can be seen in Figure 5. In this example, the robot moves from one end of the bridge to the other and then turns around. The robot then continues to the other end of the bridge and repeats this cycle until the entirety of the inspection area has been covered, and sensors have been deployed. This zig-zag path is also applicable to inspection areas like parking garages and other reinforced concrete infrastructures. The mobile base platform is capable of rotating in place to change direction, which helps to conserve power through minimizing unnecessary movements throughout the inspection process. The Seekur Jr is the smaller alternative to the Seekur, which is too large for easy navigation in narrow areas, such as bridge decks or parking garages where the inspection area can be limited by traffic or parked cars. The Seekur Jr platform is also water-

proof, which makes it possible to keep the robotic system's internal electrical systems and on-board computers safe during light rain or other weather conditions. Although the Seekur has been used in other robotic systems for some similar applications, such as in La et al. (2013), the Seekur Jr is more fitting for the tasks that the developed robotic system is designed to accomplish.

2.2 | GPR and ER

GPR has a long history of being used in civil engineering applications, as can be seen in Maser (1996) and Saarenketo and Scullion (2000). The applications for GPR in civil engineering include localization of utility lines, subsurface tanks, and rebar in reinforced concrete. Specifically, GPR can be used to locate subsurface rebar in civil infrastructures including concrete bridge decks, tunnels, parking garages, concrete ramps, seaports, airport runways, etc. Given the location of top rebar in a concrete bridge deck, it is possible to evaluate the condition of the bridge (Dinh et al., 2015; Gibb & La, 2016).

The GPR sensor works by sending a radar signal through a surface (concrete in this case) and recoding the time it takes for the radar signal to move through the surface, hit an object, and return. This time is referred to as the two-way travel time. Depending on the object that causes the GPR signal to reflect, different data are generated and a unique signature is generated in the output image. Each rebar located by the GPR sensor generates a separate and unique hyperbolic signature in the output image. In Figure 3a, as the antenna moves past the buried rebar (left to right), it constructs the hyperbolic signature of the rebar continuously by recording the reflections (yellow lines) from the rebar. The GPR sensor is equipped on the robotic system using a lift system designed specifically for the inspection purpose. This lift system makes it easy to transport the robotic system without the GPR dragging along the ground, and simultaneously allows the GPR sensor to easily be lowered to the ground when data need to be collected. The deployment system for the GPR can be seen in Figure 3b.

The ER sensor equipped on the robotic system is manufactured by Proceq USA, Inc. Its function is to measure the electrical resistance of the concrete in the civil infrastructure. The data collected from the ER unit allow the condition of the concrete to be assessed, since there is a link between the resistance of the concrete and its condition, where high resistance indicates better concrete condition and lower resistance indicates worse concrete condition (La et al., 2015). The ER

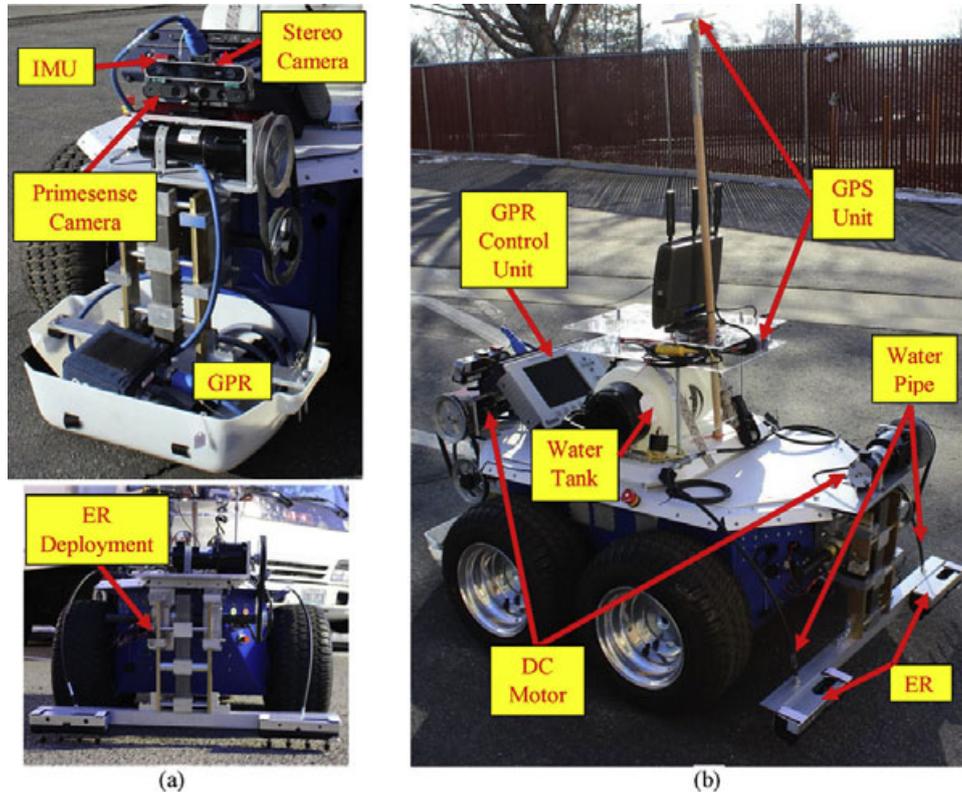


FIGURE 4 A labeled image of the robotic system for civil infrastructure inspection: (a) Front and rear view of the GPR and ER deployment systems; (b) side view with equipped NDE sensors



FIGURE 5 Sample movement of a robot on a bridge deck. The robot moves along the yellow line in the direction of red arrows while collecting data

deployment system can be seen in Figure 3b, and a diagram detailing how the equipped ER sensors work can be seen in Figure 3c.

2.3 | System implementation

The ER sensors require spraying a small amount of water on the inspection area, which might prevent the GPR sensor (as a radar-based sensor) from working. Therefore, two ER sensors and a water supply system (including a water tank and water pipes) are mounted on the rear of the robot to make sure that the GPR sensor can collect data before water is sprayed for the ER sensors.

In addition to the NDE sensors equipped on the robotic system, a Novatel Flex6 GPS unit is located at the center of the robot. The GPR sensor and both the stereo and PrimeSense cameras are mounted on the front (facing forward) of the robot. This is necessary for the robot to perform visual-based navigation tasks.

The deployment systems for the GPR and ER sensors are composed of a linear actuator and an aluminum frame. The actuators rely on a 12-V DC motor and operate by turning a screw housed within the actuator plunger and housing. They are capable of deploying with up to 25 lb of force. The frame is composed of two rails, two angled brackets, two threaded rods, four brass bearings, and a clevis. The rails mount to the underside of the robot's deck and provide support for rest of the deployment system. The brackets physically house the actuator and keep the actuator and bearings aligned. The threaded rods allow the sensors to remain aligned when the actuator is deployed. They slide through a set of brass bearings, which provides smooth, lubrication free motion. The clevis is free spinning, which prevents the actuators from burning out by allowing the actuator rod to spin freely before hitting its limit switch.

In order for the ER sensors to correctly read the resistance of the concrete when deployed, the concrete needs to be wet. A spray system,

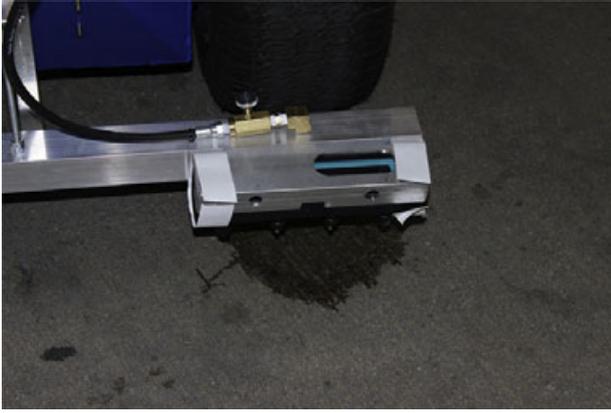


FIGURE 6 The water spray system spraying water on a part of the inspection area prior to deploying the ER sensors



FIGURE 7 An image of the ER deployment system

composed of a pressurized water tank, an assortment of pipe fittings, two needle valves, and four spray nozzles was developed to dampen the concrete. The tank, constructed of 6061 aluminum, was designed to hold up to 30 psi, with the outflow controlled by a solenoid valve. The needle valves allow the flow rate to be adjusted and set manually. Finally, the spray nozzles have a spray angle of 60 and dispense water in a full cone pattern. Four nozzles were required to fully cover the ER sensor's footprint.

A complete three-dimensional (3D) design of the robotic system can be seen in Figure 3b, and a current implementation of the robotic system can be seen in Figure 4 along with labels detailing the previously discussed sensors and navigation components. A picture of the ER water spray system can be seen in Figure 6. A picture of the overall ER deployment system can be seen in Figure 7.

To easily visualize the data being collected by the robot, a graphical user interface (GUI) was implemented. The GUI consists of three tabs: the "Scan Selection" tab, the "GPR/ER Data" tab, and the "Camera Data" tab. The "Scan Selection" tab shows the user a small preview of each finished scan and allows the user to dynamically update the data displayed on the other two tabs by selecting a scan. This tab will also update as more scans are performed, meaning it offers the user real-time access to the data being collected by the robotic sys-

tem. The "GPR/ER Data" tab shows the user a larger representation of the currently selected scan, as well as an image containing red squares where rebar were automatically detected. The "Camera Data" tab lets the user navigate through a gallery of images taken of the ground surface by the camera. This tab automatically updates as the robotic system collects more data.

Screen shots of the GUI implemented for the autonomous robotic system can be seen in Figure 8.

3 | NAVIGATION

In this section, we detail the two complimentary navigation modes implemented on the developed robotic system: GPS and IMU-based outdoor navigation and visual-inertial odometry-based indoor navigation.

3.1 | Outdoor navigation

GPS has a long history of being used in outdoor navigation, which can be seen in Farrell and Barth (1999), Grewal, Weill, and Andrews (2007), and Misra and Enge (2006). For our outdoor navigation, we use a robot operating system (ROS) software package called *robot_localization*, which uses GPS and IMU data for navigation via the EKF (Moore, 2014).

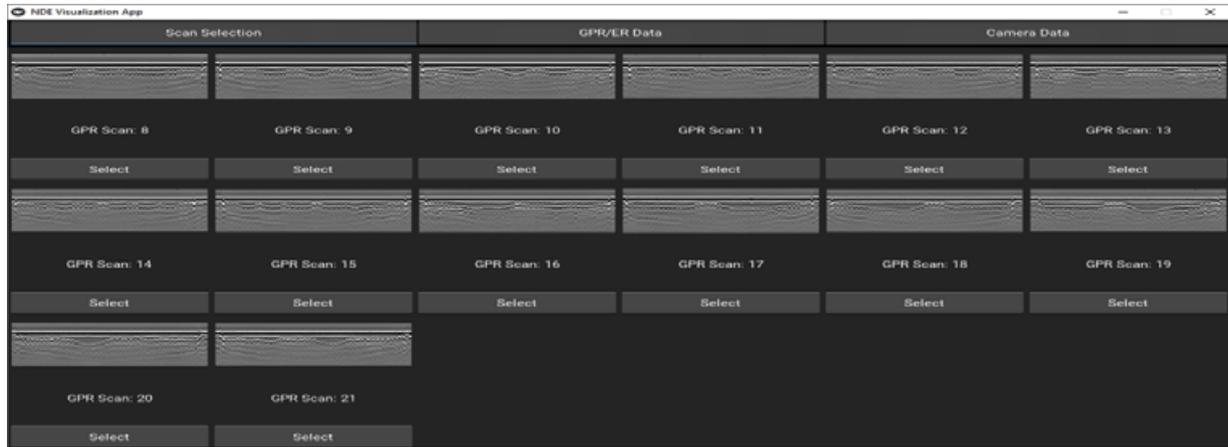
Like GPS, the EKF has been around for many years (Simon, 2006). The 2D pose and velocity of the developed system were estimated using the information from the GPS, IMU, and wheel odometer, as follows:

$$\begin{aligned} x_k &= f_{k-1}(x_{k-1}, u_{k-1}, w_{k-1}) \\ w_{k-1} &\sim (0, Q) \\ z_k &= h(x_k) + v_k \\ v_k &\sim (0, R_k). \end{aligned} \quad (1)$$

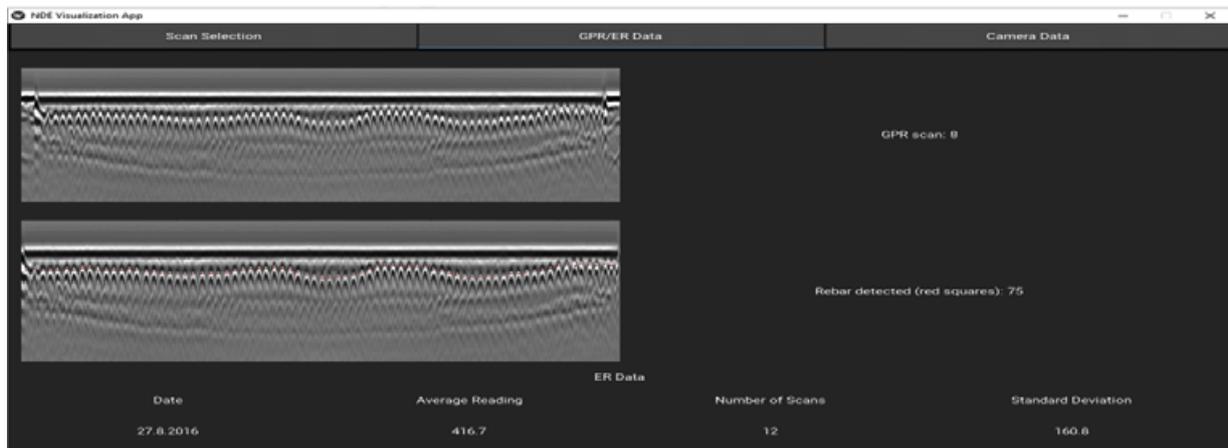
where the robot's state is x_k at time step k . f is a nonlinear state transition, with control input u , zero mean, Gaussian noise w , and covariance Q . Similarly, the measurement, z_k , has a nonlinear sensor model h , zero mean, Gaussian measurement noise v_k , and covariance R_k . Using the information in Equation (2), the prediction and correction steps can be completed:

$$\begin{aligned} \hat{x}_k &= f(x_{k-1}) \\ \hat{P}_k &= F P_{k-1} F^T + Q \\ K &= \hat{P}_k H^T (H \hat{P}_k H^T + R)^{-1} \\ x_k &= \hat{x}_k + K(z - \hat{x}_k) \\ P_k &= (I - KH) \hat{P}_k (I - KH)^T + KRK^T \end{aligned} \quad (2)$$

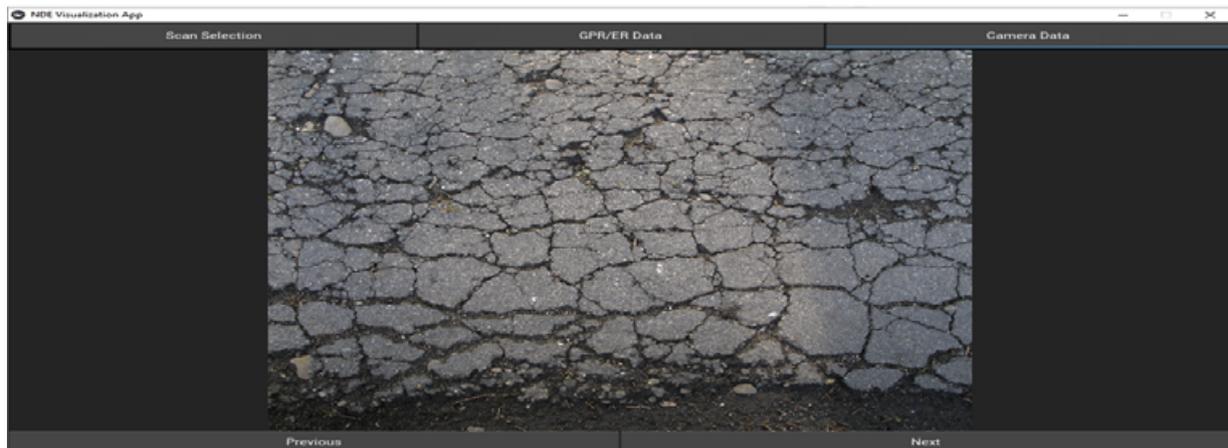
where, P is an estimated error covariance, F is a Jacobian of f , K is the Kalman gain, and H is an observation matrix. Further details on the implementation of this algorithm and how it is used for our robotic



(a) The “Scan Selection” tab of the data viewing GUI.



(b) The “GPR/ER Data” tab of the data viewing GUI.



(c) The “Camera Data” tab of the data viewing GUI.

FIGURE 8 The various tabs of the data viewing GUI for the developed robotic system

system can be seen in Moore and Stouch (2014). The trajectory of the robot during a sample outdoor navigation can be seen in Figure 9.

3.2 | Indoor navigation

Although it would be preferable to use the same navigation algorithm for both indoor and outdoor areas, this is not easily accomplished. The

GPS and IMU-based EKF localization package used for our outdoor navigation mode requires the GPS to obtain a stable satellite signal, which could be difficult or impossible when the robotic system is inside a parking garage or building constructed mostly of concrete. For this reason, we chose to implement a visual-inertial odometry-based navigation mode for indoor environments. This navigation mode makes use of the stereo camera system and IMU equipped on the developed

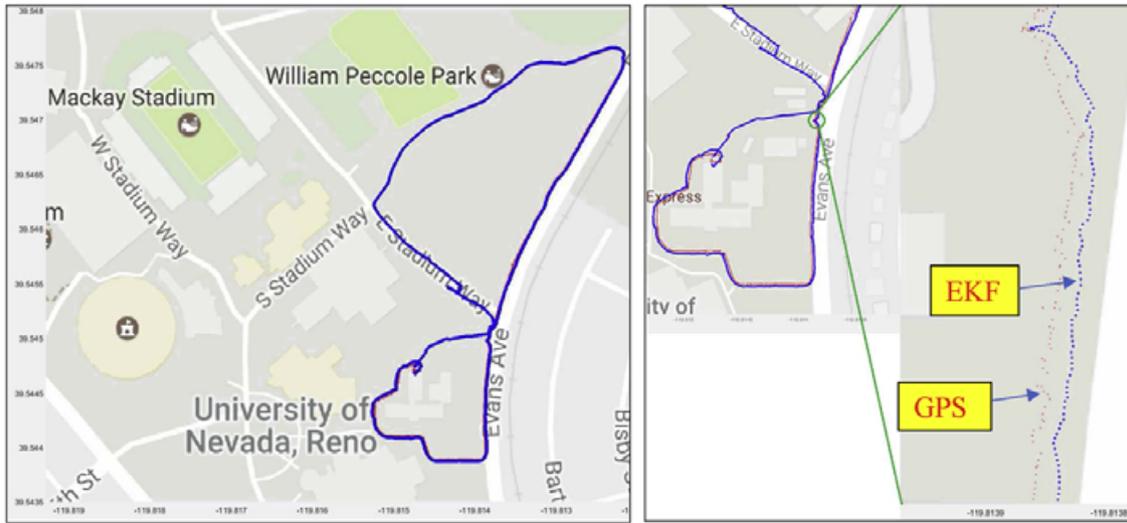


FIGURE 9 The trajectory of our outdoor navigation using the *robot_localization* ROS package (left). Red dots indicate GPS, and the dots show the output of the EKF. A close-up view of the EKF and GPS signals (right)

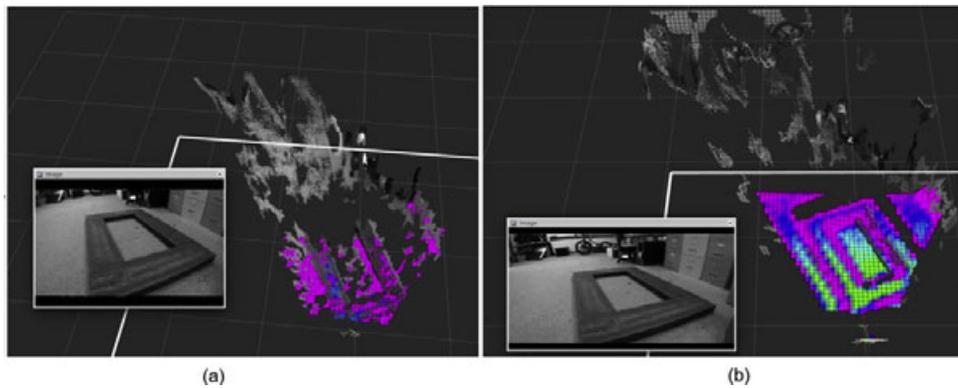


FIGURE 10 A comparison of elevation maps from two point cloud data sources: (a) point cloud data from a standard stereo camera system; (b) point cloud data from the PrimeSense camera

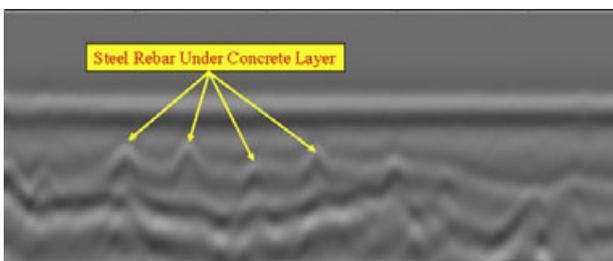


FIGURE 11 A GPR scan image where subsurface rebar are labeled

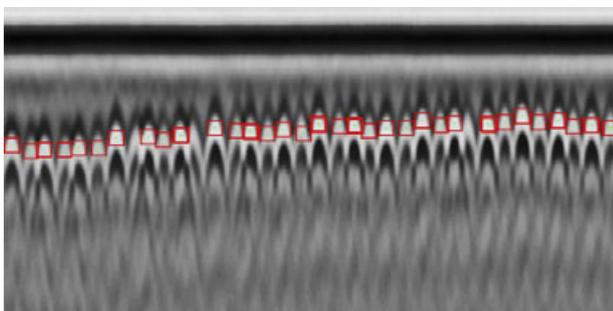


FIGURE 12 A result image from the GPR rebar localization process

robotic system. Visual-inertial odometry-based navigation has gained traction in research in recent years, as can be seen in Shen, Mulgaonkar, Michael, and Kumar (2016), Sola, Vidal-Calleja, Civera, and Montiel (2012), Wooden et al. (2010). For the developed robotic system, we chose to use the method originally proposed in Bloesch et al. (2015), including adapting this method to be used with our stereo camera system. The implemented method uses the pixel intensity error of images of the surrounding area, in combination with the EKF, to locate landmarks within the captured images. Using the location of landmarks in the captured images, it is possible to estimate the camera pose and by extension, the pose of the robotic system, which can be found through using a transformation that relates the location of the robot and the location of the camera. This implementation is available in a ROS package authored by the authors of Bloesch et al. (2015), called the *rovio* package (Bloesch, 2016).

Using the stereo camera equipped on the robotic system, it is possible to generate point cloud data of the inspection environment, which allows the robotic system to avoid obstacles during the navigation process (Hiep, Keriven, Labatut, & Pons, 2009; Fankhauser, Bloesch, Gehring, Hutter, & Siegwart, 2014). The avoidance of obstacles during the inspection process is even more crucial when the area being

inspected contains cars or other valuable obstacles that could either be damaged by the robotic system, or damage the robotic system.

To justify the use of the PrimeSense camera instead of a standard stereo camera system, a comparison between the two was conducted. This comparison tested the difference between the elevation maps generated when each respective camera was used to gather point cloud data. The difference between the elevation map generated using point cloud data from a standard stereo camera system and the elevation map generated using point cloud data from the PrimeSense camera can be seen in Figure 10.

It is visually apparent that the elevation map generated using point cloud data from the PrimeSense camera is superior to the standard elevation map. This difference can be explained through comparing the difference between the way the two cameras work. The standard stereo camera system works through matching features between successive frames captured by the camera and using them to generate a disparity map, which shows the motion between frames. Using the disparity map method can lead to very little matches between frames, and in turn, less point cloud data. Without a dense point cloud, it is not possible to accurately generate an elevation map.

The PrimeSense camera instead uses an infrared speckle pattern, which is projected by the camera and recaptured by an internal sensor. This process generates a depth map, which yields point cloud data when correlated with an RGB camera, as in Garcia and Zalevsky (2007). Consequently, the point cloud data from the PrimeSense camera are more dense than that generated by the standard stereo camera system. Given the results of this comparison, the PrimeSense camera is superior to the standard stereo camera system, but the PrimeSense only works in environments where the present objects will reflect the infrared speckle. Without this reflection, the PrimeSense camera cannot be relied upon, so the robotic system is equipped with two cameras to complement each other and allow for robust indoor navigation in the presence of varying objects and object densities. We can further enhance the reliability of the robot indoor/outdoor navigation by integrating laser range finders or LiDAR with 3D mapping technique (Nguyen, La, Sanchez, & Vu, 2016; Nguyen & La, 2015).

4 | GPR DATA AND PROCESSING

GPR data are collected by the developed robotic system throughout the inspection process. Any subsurface object with a dielectric constant different than its surroundings is detected by the GPR scanning process. This typically means that metallic objects are visually noticeable in GPR scan images, but large air pockets or plastic pipes can cause noticeable holes in the image as well. Each subsurface metallic object will have a different signature, such as hyperbolic signatures in the case of the rebar in reinforced concrete. An example image containing subsurface rebar in reinforced concrete can be seen in Figure 11, where the hyperbolic rebar signatures are labeled.

To extract information on the condition of the concrete being inspected, it is necessary to first locate the rebar within the GPR scan images. Automated rebar localization is performed using the

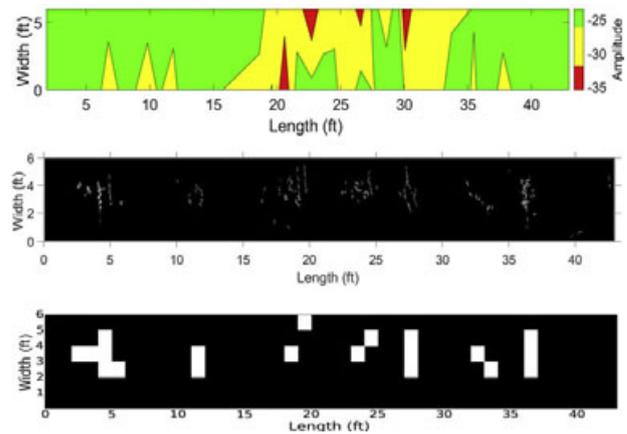


FIGURE 13 The GPR contour plot (top) and crack map (middle) of one part of the collected data. The ER deployment map generated by the NDE sensor fusion process with $\omega = 5$ (bottom)

method first proposed in Gibb and La (2016). This method utilizes a Naive Bayes classifier trained on features from images containing rebar hyperbolas. The feature descriptor used to train the Naive Bayes classifier is histogram of oriented gradients (HOG), which were first described in Dalal and Triggs (2005). The Naive Bayes classifier is applied across a sliding window that moves throughout the GPR image. After final rebar localization using the classifier results, which can be seen in Gibb and La (2016), this process yields the location of all of the rebar in the GPR scan image. An example image including the location of rebar found by this process can be seen in Figure 12. This process is entirely automated and highly accurate.

Using the rebar locations from the rebar-picking algorithm described previously, it is possible to generate a contour, or condition, map of the inspected area. These condition maps show the condition of the inspected area based on the depth of the rebar in the inspected area. GPR condition maps are represented as contour plots with colors representing the depth of rebar underneath the inspected surface. Examples of the condition maps generated using the subsurface rebar locations can be seen in the top of Figures 13 and 14, which show maps with three different color levels. The green, yellow, and red in the condition maps correspond to good, unknown, and poor condition respectively. The unknown condition simply means that the GPR data are not enough to make a final decision on the condition of the scanned area, so more inspection needs to be performed. Because the condition maps show colors based on depth, the colors can be situation specific, as there are different specifications and regulations regarding construction in different areas, which can lead to the rebar in reinforced concrete being laid at different depths.

5 | CAMERA DATA AND PROCESSING

As the robotic system performs inspection and collects data using the GPR sensor, visual data are also being recorded via the camera system on-board the robotic system. Since most modern cameras capture many frames per second, it is not feasible to process each frame

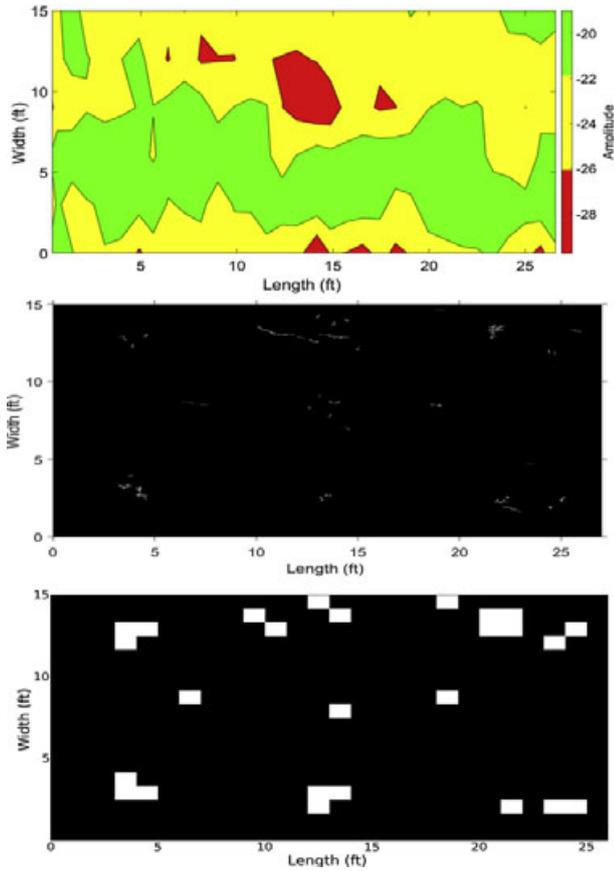


FIGURE 14 The GPR contour plot (top) and crack map (middle) of another part of the collected data. A correlation between crack density and low points on the contour map is evident here. The ER map generated by the NDE sensor fusion process with $\omega = 5$ (bottom)

captured by the camera. For this reason, Laplacian scoring is applied to each input frame. Using this scoring method allows images that are blurry to be discarded prior to further processing. This greatly reduces the number of frames present in later processing steps and helps to ensure that the images processed later contain sharp pictures of the inspection area that can reliably be used for crack detection. Despite the equipped camera system being fastened tightly to the robotic system throughout the inspection process, it is necessary to eliminate many blurry frames as the movement of the robot during inspection and the wind or bridge deck vibration can cause the camera to move and take blurry images. The steps of the Laplacian scoring method as follows.

The first step of Laplacian scoring is to convolve the current frame with a mask that serves as a discrete approximation of the Laplacian. The Laplacian mask can be seen in Equation 3:

$$L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (3)$$

The variance of the input frame is then calculated in Equation 4,

$$\sigma^2 = \frac{\sum (X - \mu)^2}{N} \quad (4)$$

where X is the value of the current pixel, μ is the mean pixel value after the Laplacian mask is applied, and N is the number of pixels in the image.

The Laplacian score yields a number that can help determine how blurred an image is. Through the process of testing the Laplacian scoring method, it was determined that any image with a Laplacian score greater than 200 could be processed further. Any image with a Laplacian score less than 200 was discarded at this step.

In addition to using the Laplacian scoring method as a preprocessing step to eliminate unnecessary frames, the speed of the robot was also used to eliminate additional frames. In the experiments run for this research, the robot generally moved at a speed of 0.5 m/s, which permits the assumption that not all of the frames captured are necessary because they have a large overlap. The number of frames that can be discarded based on this assumption varies as the speed of the robotic system does, and can even vary based on the angle at which the camera is mounted on the robotic system. This process yields a set of frames that can later be stitched together easily and eliminates the possibility of processing duplicate frames that have overlap with each other.

After preprocessing is performed to remove additional unnecessary and blurred frames, a convolutional neural network (CNN) structure is optimized and then applied to classify subimages in each frame as either containing cracks or not containing cracks. CNNs have shown to perform better than most existing methods at image classification tasks as a result of the separable and complex features descriptors that can be generated through their use (Dinh, Ha, & La, 2016; Girshick, Donahue, Darrell, & Malik, 2014; Huang, Zhou, Ding, & Zhang, 2012; Taigman, Yang, Ranzato, & Wolf, 2014). In the following subsections, an overview of CNN structures is presented, along with a description of the method used to optimize the CNN structure used in the proposed method. The genetic algorithm based optimizations for CNN structures used in the proposed method first appeared in Gibb, La, and Louis (2018).

5.1 | An overview of convolutional neural networks

There are several types of layers that make up the structure of CNNs. These layers include convolution layers, pooling layers, and fully connected layers. Convolution layers perform multiplication between the input image, or 2D array, and a filter that is also a 2D array. This multiplication happens at every location in the input array, and the result is then summed. This process yields another 2D array and can be described mathematically as follows:

$$f[x, y] * g[x, y] = \sum_{i=-\frac{s}{2}}^{\frac{s}{2}} \sum_{j=-\frac{s}{2}}^{\frac{s}{2}} f[i, j] \cdot g[x - i, y - j], \quad (5)$$

where f is the input image, or array, g is the filter, s is the size of the filter, which is a square filter, and x and y are locations in the input arrays. Equation 5 is applied many times throughout each convolution layer, with different filters used for g . The set of filters used in each convolution layer can be called g_1, g_2, \dots, g_M , where M is the number of filters in each convolution layer. With a filter size of S , and a number of convolution layers in a CNN, L , there are at least $s^2 \cdot M \cdot L$ weights that need

to be optimized during the training process. As M , s , or L increase, so does the complexity of the problem of structure optimization. Such a complex problem can be time consuming on its own, without further consideration of the additional CNN structure parameters.

The second class of layers in CNNs are pooling layers. Several types of subtypes of pooling layers exist, including mixed, max, average, tree, and gated. Of these subtypes, max-pooling is often used because it is efficient and capable of offering an abstract representation of the input data, while maintaining important features. Max-pooling subsamples its input by searching subregions of input data for local maxima and passes only those values along to the following layer. This process reduces the complexity of the input data by definition and subsequently reduces the complexity of operations in the following layers.

The final type of layer that is commonly used in CNN structure is the fully connected layer. Fully connected layers work like classic neural networks (multilayer perceptron networks). Fully connected layers, as indicated by their name, connect each input to each node in the hidden layer, and again each hidden layer node is connected directly to each output. Each hidden layer node has a corresponding weight. These weights are optimized using the back-propagation process, which will not be discussed here; however, an output node in the fully connected layer sums its value as follows:

$$\text{Output} = \omega_1 V_1 + \dots + \omega_N V_N, \quad (6)$$

where there are N nodes connected to the output node in Equation 6, and each node has an associated weight, ω_i , and value, V_i , where $i = 1, \dots, N$. Fully connected layers are typically connected to the end of a CNN as a final way to introduce nonlinearity into the network. However, large fully connected layers need to be used with caution since they can quickly introduce many weights to be optimized and greatly increase the amount of training time required for the network. This is a result of the N^2 weights that need to be optimized for each fully connected layer.

5.2 | Convolutional neural network structure optimization

As a result of the large number of parameters that need to be optimized for convolution layers, pooling layers, and fully connected layers, in addition to the number of layers in the network and their relative order, optimizing the structure of a CNN for image classification is a nontrivial task that requires a deep understanding of the various parameters and methodology. For this reason, a GA was used to search for high-performance (accuracy) network structures (Gibb et al., 2018). GAs perform well on search problems that have nonlinear relationships, like the relationship between an input image and the features generated by a certain CNN structure. In order for the GA to have as much control as possible over the network structure, it is necessary to allow for many types of networks through the evolution process. The network structure parameters can be seen in Table 1. Each individual in the population consists of a chromosome represented, in binary, by 19 bits total. Although $2^{19} \approx 525,000$, it is important to note that an exhaustive search is not feasible. The variety of possible network archi-

TABLE 1 An overview of the main parameters that can be evolved by the GA

Parameter	Value range	Bit representation
Convolution layers	1–16	4 bits
Max-pooling layers	1–4	0 bits
Size of Conv. filters	1–32 (pixels)	5 bits
Number of Conv. filters	1–32	5 bits

tectures reinforce the fact that this problem is nontrivial and time consuming.

Each time an evaluation of an individual takes place, a network must first be constructed based on the individual's chromosome. The structure of the network depends largely on the number of convolution layers specified in the chromosome. A max-pooling layer is added to the network after every set of four convolution layers to reduce the complexity of the network. The network always ends with a fully connected layer with two nodes, which uses a softmax activation function. The final fully connected layer will output probabilities that the input image belongs to the "crack" class and the "no-crack" class, respectively. Everywhere else, the rectified linear unit (ReLU) activation function is used. Each pooling layer uses a 4

4 filter with a stride of 4. Stochastic gradient descent (SGD) is used as the optimization method, with a learning rate of 0.001. SGD is an optimization technique that is used in the back-propagation process to optimize the weights within the network. SGD is described mathematically as

$$\theta = \theta - \alpha \cdot \nabla_{\theta} \cdot J(\theta; x^{(i)}, y^{(i)}), \quad (7)$$

where θ is the parameter being optimized, α is the learning rate, J is the objective function, $x^{(i)}$ is sample i , and $y^{(i)}$ is label i .

Once a model is constructed using a chromosome, the model is trained on a set of training data. The training data set has been labeled manually as either belonging to the "crack class" or to the "no-crack" class. Example of images from both classes can be seen in Figure 15. The training data consist of 3,000 images: 1,500 crack images, and 1,500 no-crack images. Each network is then given 30 epochs of training time to back-propagate and the epoch displaying the highest training accuracy is kept, so that any decrease in network performance or errors will not be accidentally incorporated into the evolution process.

Once a network is constructed, its fitness is evaluated for use in selection in the GA. The fitness function used by the GA is simply the classification accuracy of the constructed network. Each individual in the network is evaluated on 600 test images, 300 containing cracks, and 300 not containing cracks. The accuracy of a constructed network is represented as a decimal in $[0, 1]$, and can be described as

$$f = \frac{c}{t}, \quad (8)$$

where c is correct classifications, t is total classifications, and f is fitness.

The method detailed here is then allowed to evolve using the (μ, λ) algorithm, which is a special version of the canonical GA. In this evolutionary strategy, λ parents are generated each generation and then μ individuals are chosen from the parents and offspring to continue on to

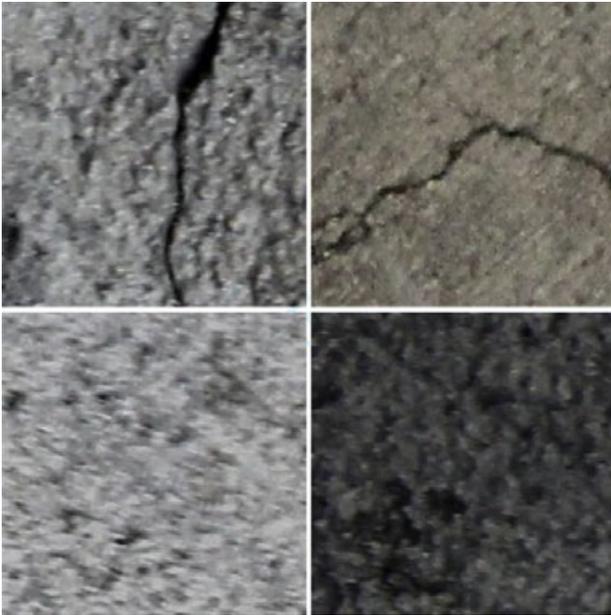


FIGURE 15 Images from the “crack” training data (top), and images from the “no-crack” training data (bottom)

the next generation. This algorithm is run for 10 times, for 10 generations each run, with a population size of 30. The results of the best CNN structure found through the evolution process can be seen in Table 2. The CNN is then used in a sliding window approach along the input

TABLE 2 The results from the best CNN structure obtained from the evolution process after classifying 10 test images split into 256 pixel by 256 pixel subimages. The total number of subimages classified was 4,880

Correct classifications	4,586
Incorrect classifications	294
Accuracy	0.94
False positives	280
False negatives	14

image. The output of the CNN is an image where only areas containing cracks remain. An example of the CNN output can be seen in Figure 16.

5.3 | Local edge detection

After the CNN is applied to concrete surface images, the Canny edge detection algorithm is used locally in areas that the CNN classified as containing cracks. The Canny edge detection method was first proposed in 1986 and remains an easy to implement algorithm that has been used in many applications (Canny, 1986). This edge detection method is used to locate cracks in the local areas. Because of the wide use of the Canny edge detection algorithm, only an overview is provided here, and interested readers can consult the original for more information on the method.

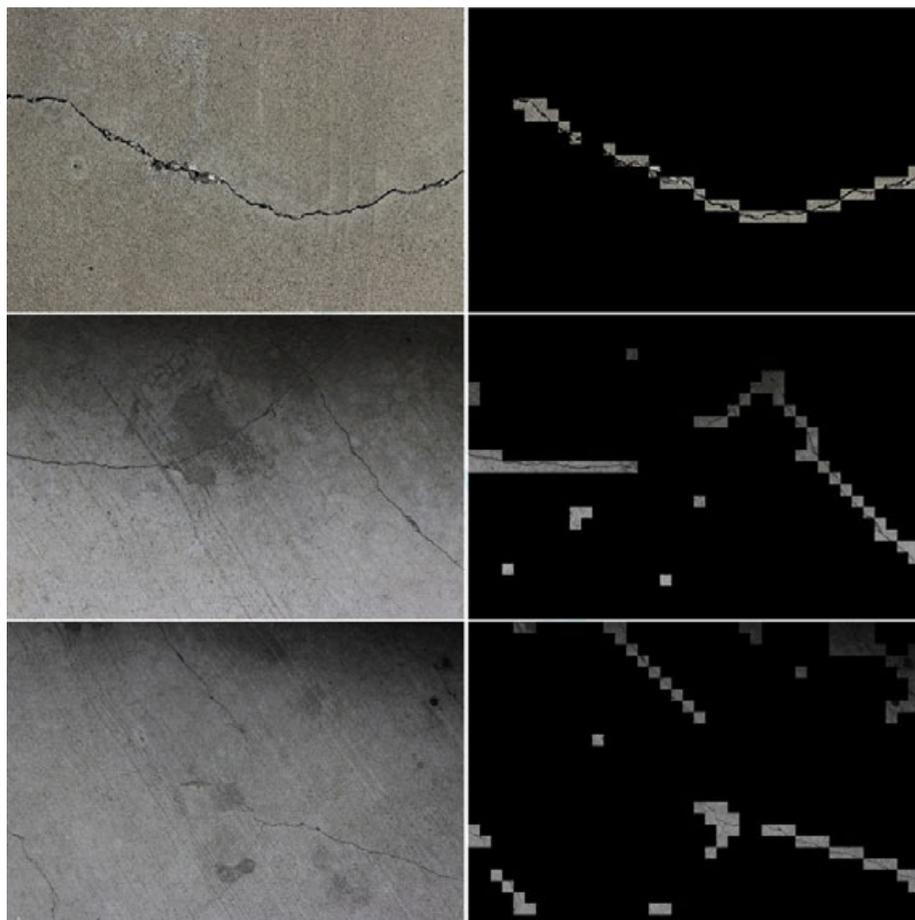


FIGURE 16 Input frames (left), and output images from the CNN after classification (right)

The first step of the Canny method is to smooth each of the input images via convolution with a Gaussian mask (filter), which serves to remove noise from the image that can propagate through later steps and affect the success of the edge detection method. A horizontal filter, vertical filter, and two diagonal filters are applied to an input image to get horizontal, vertical, and diagonal edges. Then the gradient and direction are calculated, as can be seen in Equations 9 and 10.

$$G = \sqrt{G_x^2 + G_y^2}, \quad (9)$$

$$\theta = \text{atan2}(G_y, G_x), \quad (10)$$

where G_x and G_y are image derivatives in the x and y directions, respectively.

This yields a grayscale image with values closer to white indicating the presence of edges. Nonmaxima suppression is performed on the grayscale image to remove edges that are not as pronounced as those in the same local area. Finally, double thresholding is performed along with hysteresis to keep only strong edges.

Once Canny edge detection has been performed on all of the input frames, they are stitched together to form a crack map of the inspection area, which shows all of the cracks that were detected throughout the edge detection process. A label matrix is created that corresponds to the crack map. The label matrix contains the segmented edges in the map along with what pixels belong to those edges, to make further processing easier. Using the labeled cracks in the label matrix, the major axis of each crack is calculated, as well as the length of each crack along the major axis. The length of each crack along its major axis is used to apply a length filter to the cracks in the map. Filtering the cracks by length allows for us to remove extremely small edges that are insignificant in the inspection process, as well as those cracks that are abnormally large and typically actually caused by concrete slabs meeting within the captured frame. Empirically, we found that cracks between 50 and 500 pixels should be kept, and the remainder of the cracks should be filtered out. An example binary image showing the detected cracks within an image, after the edge length filter is applied, can be seen in Figure 17. Example crack maps can be seen in the middle of Figures 13 and 14.

6 | NDE SENSOR FUSION

In this section, we propose an inspection framework based on NDE sensor data fusion. To reduce inspection time, the robot needs to decide in which area it can perform a fast inspection (using less sensors) and which area requires an in-depth inspection (using all available sensors).

One of the primary contributions of the proposed method is that it fuses sensor data to minimize inspection time. Of the sensors on the robotic platform, three are used for inspection: the camera, GPR, and the ER sensor. Of these three sensors, ER takes the most time to deploy. The robot first needs to stop and spray water on the concrete (inspected spot) to create a conductive environment, and then deploy the ER sensors to collect data. This process means it may not be viable



FIGURE 17 An example image of the surveyed area (top) and an example output image from the crack detection algorithm with length filter applied (bottom)

to deploy the ER sensor everywhere in an inspection area. Through deploying the camera and GPR unit, it is possible to eliminate a majority of the inspection time that would result in using the ER sensor everywhere in the inspection area.

The crack map generated from the camera data, and the condition map generated from the GPR data can be fused into an ER deployment map so that the ER sensor is only deployed in areas where it is needed to ascertain the condition of the inspection area, and not everywhere. Sensor fusion is a technique for improving the results of data from an individual sensor through combining it with data from another sensor. This technique is used for many types of sensors and data (Duraismy, Bertolucci, Loehlein, & Schwarz, 2016; Romero, Borges, Elfes, & Pfrunder, 2016; Tanzmeister & Steyer, 2016; Zhang, Zhong, Wu, & Qian, 2016). Although GPR data for condition mapping has not been fused with camera data for crack detection, research on fusing GPR data with other sensors does exist (Boubaki, Saintenoy, & Tucholka, 2011; Larionova, Marques, & Almeida, 2006; Nath & Bhuiyan, 2007), and research on fusion of other NDE sensors also exists (Heideklang & Shokouhi, 2013; Khan, Ramuhalli, & Dass, 2011; Tian, Sophian, Taylor, & Rudlin, 2005; Lu & Michaels, 2009).

In our proposed method, sensor fusion is accomplished through use of a confidence matrix. This matrix, M_d , can be seen in Equation 11.

$$M_d = \begin{bmatrix} d & 0 \\ d & 0/1 \\ 0 & 0 \end{bmatrix} \quad (11)$$

Equation 11 shows the decision matrix where the first row is the case where the condition map is green in the given area, the second row is the case where the condition map is yellow in the given area, and the third row is the case where the decision map is red in the given area. The first column of the decision matrix represents the case where there are cracks present in the corresponding area of the crack map, and the second column represents the case where there are no cracks present in the corresponding area of the crack map.

The explanation for the matrix in Equation 11 is that GPR serves as the primary source of information about the area that is being inspected. When the GPR data indicate the inspection area is in good condition (green on the condition map), then crack information is not necessary, as it does not provide any additional certainty that the inspection area is in good or poor condition. Element (1, 2) in the decision matrix represents this case. In addition, if the GPR shows that the inspection area is in poor condition, crack information is not required since the GPR provides high certainty that the inspection area needs repair and in what precise location. Elements (3, 1) and (3, 2) in the decision matrix represent these cases. The final cases are where the GPR data do not show conclusively what the condition of the inspection area is, which requires fusion with crack data to determine whether the ER sensor should be deployed to make a final decision on the condition of the area in question. Elements (1, 1) and (2, 1) in the decision matrix represent these cases. In the case of element (2, 2), since there are no cracks present in that area, it is not possible to use the crack information in conjunction with the GPR data, so the deploying the ER can be left up to the user. In the case of element (2, 1) in the decision matrix, it is necessary to analyze the presence of cracks in the given area with respect to the surrounding area, as in Equations 12–15. This decision matrix is applied across the inspection area by splitting the area into equally sized cells and making a decision on a cell by cell basis as to whether the ER sensor should be deployed.

$$d = \text{round}(\omega * \frac{\sigma_C - \mu_{IA}}{m_{IA} - \mu_{IA}}) \quad (12)$$

In Equation 12, ω represents an adjustable weight that is a value greater than 1, which can be increased to place more importance on the cracks in the crack map.

$$\sigma_C = \sum_{i,j=1}^N I_C(i,j) \quad (13)$$

In Equation 13, $I_C(i,j)$ is the intensity value at pixel (i,j) in the crack map, where i and j both go from 1 to N . These values are summed across the entire cell, which gives the total amount of cracks in that cell, C .

$$\mu_{IA} = \frac{1}{M} \sum_{i=1}^M \sum_{j=1}^N I_C(i,j) \quad (14)$$

In Equation 14, μ_{IA} represents the average intensity value of each cell, C , across the entirety of the inspection area, IA , where M is the number of cells in the inspection area.

$$m_{IA} = \max(\sigma_1, \sigma_2, \dots, \sigma_C) \quad (15)$$

In Equation 15, m_{IA} is the maximum amount of cracks present in a single cell across the entire inspection area. Using Equation 12 allows each cell to be represented in terms of its crack density (the amount of cracks present in the respective cell) with respect to the rest of the cells in the inspection area. The value of d is rounded to yield a binary decision on whether the ER sensor should be deployed in that cell.

The robot was deployed on several indoor and outdoor parking garages as well as concrete decks on University of Nevada, Reno campus as shown in Figure 18. Two main areas were used for the exper-

TABLE 3 ER deployment time savings

Importance of cracks in sensor fusion (ω)	Percentage of ER deployment time saved (inspection area 1 as in Figure 13 (bottom))	Percentage of ER Deployment Time Saved (Inspection Area 2 as in Figure 14 (bottom))
1	98.837	98.931
1.25	97.674	98.504
1.5	96.511	98.290
1.75	96.511	98.076
2	94.961	98.076
2.5	94.186	97.008
3	93.798	95.512
3.5	93.41	95.299
4	92.635	95.085
5	92.248	94.444
10	89.922	92.094
50	85.271	88.247

imental results in this paper, as can be seen in Figures 13 and 14. If the proposed sensor fusion method were not employed, with a cell size of one square foot, the ER sensor would need to be deployed at each cell in the inspection area. There are a total of 258 cells in the first inspection area, and 390 in the second inspection area. Considering the time required to move the robot to the cell, spray water, move the robot again to align the ER sensor with the damp area, and finally deploy the ER sensor, several seconds are required for each cell to be inspected. An ER deployment time of several seconds can lead to an overall inspection time of more than an hour if the inspection area is large, which is not feasible depending on the cost of traffic control to keep the inspection area closed down for inspection.

Examples of the ER maps generated by the proposed NDE sensor fusion method can be seen in Figures 13 and 14. A table showing the percentage of ER deployment time can be seen in Table 3. The table shows that the proposed NDE sensor fusion method saves a large percent of the inspection time that would otherwise be spent on deploying the ER sensor every foot within the inspection area. As ω increases, the number of areas where the ER is deployed also increases, meaning time saved decreases. Despite the increase in ω , a large portion of time is saved throughout the inspection process when the proposed sensor fusion method is employed. The possibility of cutting inspection time from over an hour to only a few minutes means the proposed sensor fusion method could have a meaningful impact on the use of NDE sensors for inspecting civil infrastructure.

Figure 19 shows a snapshot of the robot autonomously navigating in indoor environment. Figure 20 shows that the robot is inspecting the concrete floor of an indoor parking garage at UNR campus while being able to recognize obstacles/cars based on the real-time evaluation map.

For more detail of the robot deployment for civil infrastructure inspection, please see the video link here: <https://www.youtube.com/watch?v=DSpRxc18hnM&feature=youtu.be>

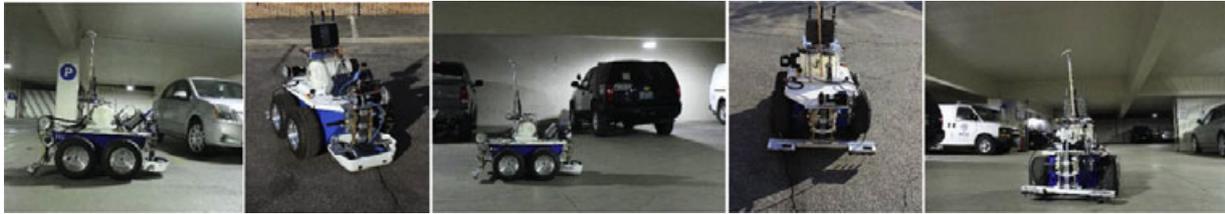


FIGURE 18 Deployment of the developed robotic system for inspection of different indoor/outdoor parking garages and concrete decks on the University of Nevada, Reno campus

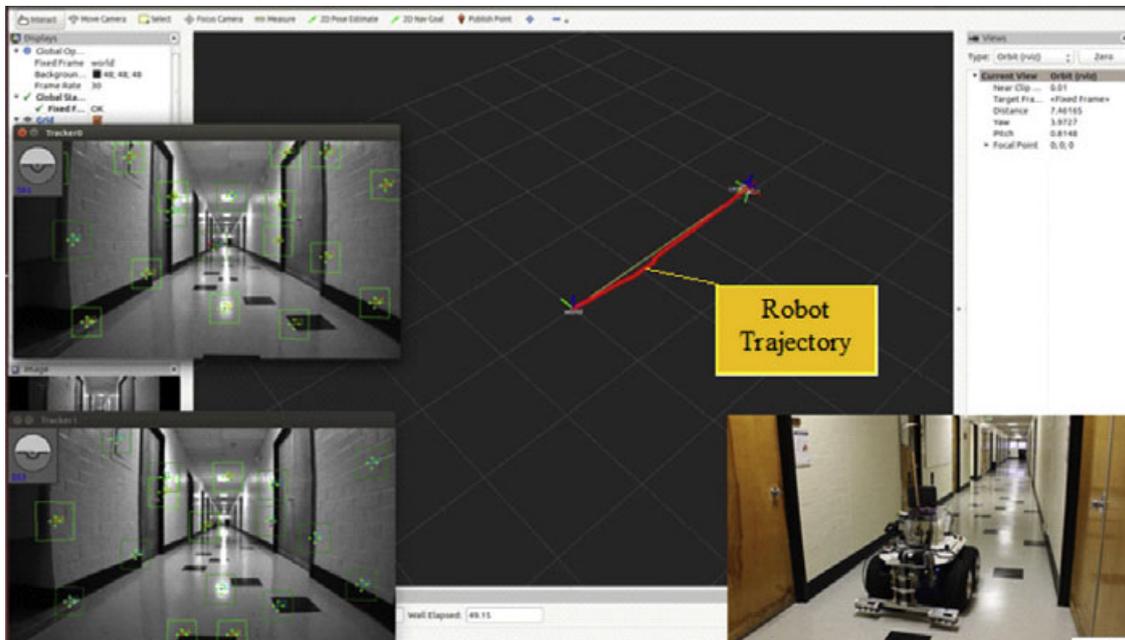


FIGURE 19 A snapshot of the autonomous localization and navigation with visual odometry in ROS visualization (RVIZ) interface

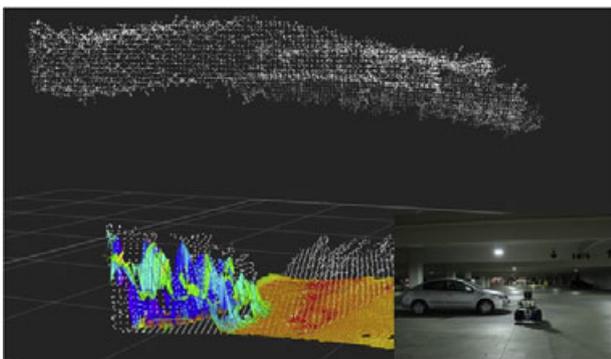


FIGURE 20 The robot is building an elevation map for autonomous path planning while inspecting a garage parking lot

7 | CONCLUSIONS

In this paper, a robotic system was developed for civil infrastructure inspection using the proposed NDE sensor fusion method. The robotic system is equipped with multiple NDE sensors, including a GPR sensor, ER sensors, and a camera. The data collected from these sensors is fused to reduce inspection time through only deploying the time intensive ER sensors in the necessary areas. The robotic system is capable of both indoor and outdoor navigation, which allows it to navi-

gate regardless of the ability to get a GPS signal. The proposed fusion method is validated through two sets of test data, both of which show the greatly reduced inspection time as a result of the proposed method. The results indicate a correlation between cracks in the concrete surface and the attenuation level in the reinforced concrete. This shows in the ER deployment map, where only the areas that show some correlation between the crack data and GPR data are inspected using the ER sensor. The proposed method was a valuable experiment in ways to reduce the time required to perform complete inspections of civil infrastructure.

In the future, we intend to integrate a thermal sensor on the robotic system to decrease the inspection time even more by offering an additional input to the NDE data driven processes. This will require an extension of the proposed sensor fusion method. We also intend to implement path planning based on the output of the sensor fusion process, which will allow the robot to complete the entire inspection process autonomously, including the planning of its path across the inspection area to deploy the ER sensors where necessary. We have considered one way of accomplishing this, which is through the use of a belief-based system, modeled by partially observable Markov decision processes. Such an improvement on our system could greatly improve on the overall inspection quality and the time required to perform the inspection.

ACKNOWLEDGMENTS

This work is supported by National Science Foundation under grants: NSF-IIP-I-Corps #1639092, and the National Aeronautics and Space Administration (NASA) under Grant No. NNX10AN23H issued through the Nevada Space Grant, and the Nevada Advanced Autonomous Systems Innovation Center (NAASIC) at the University of Nevada, Reno. This work is also supported by 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 (<http://inspire-utc.mst.edu>) at Missouri University of Science and Technology. 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 USDOT/OST-R, or any State or other entity.

ORCID

Spencer Gibb  <http://orcid.org/0000-0002-3560-6067>

Hung Manh La  <http://orcid.org/0000-0003-2183-2634>

REFERENCES

- ASCE. (2016). Failure to act report - Closing the infrastructure investment gap for America's economic future. <https://www.asce.org/failuretoact/>
- Bloesch, M. (2016). Robust visual inertial odometry. Retrieved from <https://github.com/ethz-asl/rovio>
- Bloesch, M., Omari, S., Hutter, M., & Siegwart, R. (2015). Robust visual inertial odometry using a direct EKF-based approach. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 298–304). Piscataway, NJ: IEEE.
- Boubaki, N., Saintenoy, A., & Tucholka, P. (2011). GPR profiling and electrical resistivity tomography for buried cavity detection: A test site at the abbaye de l'Ouye (France). In *2011 6th International Workshop on Advanced Ground Penetrating Radar (IWAGPR)* (pp. 1–5). <https://doi.org/10.1109/IWAGPR.2011.5963852>
- Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6), 679–698. Piscataway, NJ: IEEE. <https://doi.org/10.1109/TPAMI.1986.4767851>
- Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (Vol. 1, 886–893). Piscataway, NJ: IEEE. <https://doi.org/10.1109/CVPR.2005.177>
- Dinh, K., Gucunski, N., Kim, J., Duong, T. H., & La, H. M. (2015). Attenuation-based methodology for condition assessment of concrete bridge decks using GPR. In *ISARC. Proceedings of the 32nd International Symposium on Automation and Robotics in Construction*, June 15–18, 2015, Oulu, Finland (Vol. 32, p. 1). International Association for Automation and Robotics in Construction (I.A.A.R.C.). <http://www.iaarc.org/index.html>
- Dinh, T. H., Ha, Q. P., & La, H. M. (2016). Computer vision-based method for concrete crack detection. In *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)* (pp. 1–6). Piscataway, NJ: IEEE. <https://doi.org/10.1109/ICARCV.2016.7838682>
- Duraisamy, B., Bertolucci, M., Loehlein, O., & Schwarz, T. (2016). Object management strategy for an unified high level automotive sensor fusion framework. In *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)* (pp. 134–141). Piscataway, NJ: IEEE. <https://doi.org/10.1109/MFI.2016.7849479>
- Fankhauser, P., Bloesch, M., Gehring, C., Hutter, M., & Siegwart, R. (2014). Robot-centric elevation mapping with uncertainty estimates. In *International Conference on Climbing and Walking Robots (CLAWAR)* (pp. 433–440). Singapore: World Scientific.
- Farrell, J., & Barth, M. (1999). *The global positioning system and inertial navigation*. New York, NY: McGraw-Hill Professional.
- FHWA. (2015). Highway bridges by wearing surface 2015. <https://www.fhwa.dot.gov/bridge/nbi/no10/wearing15.cfm>
- Frangopol, D. M., Lin, K.-Y., & Estes, A. C. (1997). Life-cycle cost design of deteriorating structures. *Journal of Structural Engineering*, 123(10), 1390–1401.
- Furuta, H., Kameda, T., Fukuda, Y., & Frangopol, D. M. (2004). Life-cycle cost analysis for infrastructure systems: Life-cycle cost vs. safety level vs. service life. In *Life-cycle performance of deteriorating structures: Assessment, design and management* (pp. 19–25). Reston, VA: American Society of Civil Engineers.
- Garcia, J., & Zalevsky, Z. (2007). Range mapping using speckle decorrelation. U.S. Patent App. 11/712,932. Retrieved from <https://www.google.com/patents/US20070216894>.
- Gibb, S., & La, H. M. (2016). Automated rebar detection for ground-penetrating radar. In G. Bebis et al. (Eds.), *Advances in visual computing*, 12th International Symposium, ISVC 2016, Las Vegas, NV, December 12–14, 2016 (Part I, pp. 815–824). Cham, Switzerland: Springer International Publishing. https://doi.org/10.1007/978-3-319-50835-1_73.
- Gibb, S., La, H. M., & Louis, S. (2018). A genetic algorithm for convolutional network structure optimization for concrete crack detection. In *2018 IEEE Congress on Evolutionary Computation*. To appear.
- Gibb, S., Le, T., La, H. M., Schmid, R., & Berendsen, T. (2017). A multi-functional inspection robot for civil infrastructure evaluation and maintenance. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2672–2677). Piscataway, NJ: IEEE. <https://doi.org/10.1109/IROS.2017.8206091>
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 580–587). Piscataway, NJ: IEEE. <https://doi.org/10.1109/CVPR.2014.81>
- Grewal, M. S., Weill, L. R., & Andrews, A. P. (2007). *Global positioning systems, inertial navigation, and integration*. Hoboken, NJ: John Wiley & Sons.
- Gucunski, N. (2015). Condition assessment of bridge deck using various nondestructive evaluation (NDE) technologies. *LTBP*, 5, 1–7.
- Gucunski, N., Kee, S., La, H. M., Basily, B., Maher, A., & Ghasemi, H. (2015a). Implementation of a fully autonomous platform for assessment of concrete bridge decks RABIT. In *Structures Congress 2015* (pp. 23–25). Reston, VA: American Society of Civil Engineers.
- Gucunski, N., Kee, S.-H., La, H. M., Basily, B., & Maher, A. (2015b). Delamination and concrete quality assessment of concrete bridge decks using a fully autonomous RABIT platform. *Structural Monitoring and Maintenance*, 2(1), 19–34.
- Gucunski, N., Maher, A., Basily, B., La, H. M., Lim, R., Parvardeh, H., & Kee, S. (2013). Robotic platform rabbit for condition assessment of concrete bridge decks using multiple NDE technologies. *CrSNDT Journal*, 3(4), 5–12.
- Heideklang, R., & Shokouhi, P. (2013). Application of data fusion in nondestructive testing (NDT). In *Proceedings of the 16th International Conference on Information Fusion* (pp. 835–841). Piscataway, NJ: IEEE.
- Hiep, V. H., Keriven, R., Labatut, P., & Pons, J.-P. (2009). Towards high-resolution large-scale multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition, 2009* (pp. 1430–1437). Piscataway, NJ: IEEE.

- Huang, G. B., Zhou, H., Ding, X., & Zhang, R. (2012). Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2), 513–529. <https://doi.org/10.1109/TSMCB.2011.2168604>
- Kang, Y.-J., & Wen, Y. (2000). Minimum life-cycle cost structural design against natural hazards (Technical Report). University of Illinois Engineering Experiment Station and College of Engineering. University of Illinois at Urbana-Champaign.
- Khan, T., Ramuhalli, P., & Dass, S. C. (2011). Particle-filter-based multi-sensor fusion for solving low-frequency electromagnetic NDE inverse problems. *IEEE Transactions on Instrumentation and Measurement*, 60(6), 2142–2153. <https://doi.org/10.1109/TIM.2011.2117170>
- La, H. M., Dinh, T. H., Pham, N. H., Ha, Q. P., & Pham, A. Q. (2018). Automated robotic monitoring and inspection of steel structures and bridges. *Robotica*, 1–21. <https://doi.org/10.1017/S0263574717000601>
- La, H. M., Gucunski, N., Dana, K., & Kee, S. (2017). Development of an autonomous bridge deck inspection robotic system. *Journal of Field Robotics*, 34(8), 1489–1504. Retrieved from <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21725>, <https://doi.org/10.1002/rob.21725>
- La, H. M., Gucunski, N., Kee, S.-H., & Nguyen, L. V. (2015). Data analysis and visualization for the bridge deck inspection and evaluation robotic system. *Visualization in Engineering*, 3(1), 6. <https://doi.org/10.1186/s40327-015-0017-3>
- La, H. M., Gucunski, N., Kee, S.-H., Yi, J., Senlet, T., & Nguyen, L. (2014). Autonomous robotic system for bridge deck data collection and analysis. In *2014 IEEE/RISJ International Conference on Intelligent Robots and Systems* (pp. 1950–1955). Piscataway, NJ: IEEE.
- La, H. M., Lim, R. S., Basily, B., Gucunski, N., Yi, J., Maher, A., ... Parvardeh, H. (2013a). Autonomous robotic system for high-efficiency non-destructive bridge deck inspection and evaluation. In *2013 IEEE International Conference on Automation Science and Engineering (CASE)* (pp. 1053–1058). Piscataway, NJ: IEEE. <https://doi.org/10.1109/CoASE.2013.6653886>
- La, H. M., Lim, R. S., Basily, B. B., Gucunski, N., Yi, J., Maher, A., ... Parvardeh, H. (2013b). Mechatronic systems design for an autonomous robotic system for high-efficiency bridge deck inspection and evaluation. *IEEE/ASME Transactions on Mechatronics*, 18(6), 1655–1664.
- Larionova, S., Marques, L., & Almeida, A. T. D. (2006). Multi-stage sensor fusion for landmine detection. In *2006 IEEE/RISJ International Conference on Intelligent Robots and Systems* (pp. 2943–2948). Piscataway, NJ: IEEE. <https://doi.org/10.1109/IROS.2006.282148>
- Le, T. D., Gibb, S., Pham, N. H., La, H. M., Falk, L., & Berendsen, T. (2017). Autonomous robotic system using non-destructive evaluation methods for bridge deck inspection. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 3672–3677). Piscataway, NJ: IEEE.
- Li, Y., Wang, Y., Chase, J. G., Mattila, J., Myung, H., & Sawodny, O. (2013). Survey and introduction to the focused section on mechatronics for sustainable and resilient civil infrastructure. *IEEE/ASME Transactions on Mechatronics*, 18(6), 1637–1646. <https://doi.org/10.1109/TMECH.2013.2283537>
- Lim, R. S., La, H. M., Shan, Z., & Sheng, W. (2011). Developing a crack inspection robot for bridge maintenance. In *2011 IEEE International Conference on Robotics and Automation* (pp. 6288–6293). Piscataway, NJ: IEEE.
- Lim, R. S., La, H. M., & Sheng, W. (2014). A robotic crack inspection and mapping system for bridge deck maintenance. *IEEE Transactions on Automation Science and Engineering*, 11(2), 367–378. <https://doi.org/10.1109/TASE.2013.2294687>
- Lu, Y., & Michaels, J. E. (2009). Feature extraction and sensor fusion for ultrasonic structural health monitoring under changing environmental conditions. *IEEE Sensors Journal*, 9(11), 1462–1471. <https://doi.org/10.1109/JSEN.2009.2019339>
- Maser, K. R. (1996). Condition assessment of transportation infrastructure using ground-penetrating radar. *Journal of Infrastructure Systems*, 2(2), 94–101.
- Misra, P., & Enge, P. (2006). *Global positioning system: Signals, measurements and performance* (2nd ed.). Lincoln, MA: Ganga-Jamuna Press.
- Moore, T. (2014). robot_localization. Retrieved from http://wiki.ros.org/robot_localization
- Moore, T., & Stouch, D. (2014). A generalized extended Kalman filter implementation for the robot operating system. In E. Menegatti, N. Michael, K. Berns, & H. Yamaguchi (Eds.), *Intelligent Autonomous Systems 13* (pp. 335–348). *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*. Cham, Switzerland: Springer.
- Nath, B. K., & Bhuiyan, A. (2007). A geometrical feature based sensor fusion model of GPR and IR for the detection and classification of anti-personnel mines. In *Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007)*. (pp. 849–856). Piscataway, NJ: IEEE. <https://doi.org/10.1109/ISDA.2007.21>
- Nguyen, L. V., & La, H. M. (2015). Development of a smart shoe for building a real-time 3D map. In *The 32nd International Symposium on Automation and Robotics in Construction and Mining (ISARC)*, Oulu, Finland (pp. 1–8). International Association for Automation and Robotics in Construction (I.A.A.R.C.). <http://www.iaarc.org/index.html>
- Nguyen, L. V., La, H. M., Sanchez, J., & Vu, T. (2016). A smart shoe for building a real-time 3D map. *Automation in Construction*, 71, 2–12. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0926580516300401> (the special issue of 32nd International Symposium on Automation and Robotics in Construction). <https://doi.org/10.1016/j.autcon.2016.03.001>
- Pham, N. H., & La, H. M. (2016). Design and implementation of an autonomous robot for steel bridge inspection. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 556–562. <https://doi.org/10.1109/ALLERTON.2016.7852280>
- Pham, N. H., La, H. M., Ha, Q. P., Dang, S. N., Vo, A. H., & Dinh, Q. H. (2016). Visual and 3D mapping for steel bridge inspection using a climbing robot. In *The 33rd International Symposium on Automation and Robotics in Construction and Mining (ISARC)* (pp. 1–8). International Association for Automation and Robotics in Construction (I.A.A.R.C.). <http://www.iaarc.org/index.html>
- Phung, M. D., Quach, C. H., Dinh, T. H., & Ha, Q. (2017). Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection. *Automation in Construction*, 81, 25–33. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0926580517303825> <http://doi.org/10.1016/j.autcon.2017.04.013>
- Romero, A. R., Borges, P. V. K., Elfes, A., & Pfrunder, A. (2016). Environment-aware sensor fusion for obstacle detection. In *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)* (pp. 114–121). Piscataway, NJ: IEEE. <https://doi.org/10.1109/MFI.2016.7849476>
- Saarenketo, T., & Scullion, T. (2000). Road evaluation with ground penetrating radar. *Journal of Applied Geophysics*, 43(2), 119–138.
- Scherer, S., Singh, S., Chamberlain, L., & Elgersma, M. (2008). Flying fast and low among obstacles: Methodology and experiments. *The International Journal of Robotics Research*, 27(5), 549–574. <https://doi.org/10.1177/0278364908090949>
- Shen, S., Mulgaonkar, Y., Michael, N., & Kumar, V. (2016). Initialization-free monocular visual-inertial state estimation with application to autonomous MAVs. In M. Hsieh, O. Khatib, & V. Kumar, (Eds.), *Experimental robotics* (pp. 211–227). Cham, Switzerland: Springer.

- Simon, D. (2006). *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. Hoboken, NJ: Wiley.
- Sola, J., Vidal-Calleja, T., Civera, J., & Montiel, J. M. (2012). Impact of landmark parametrization on monocular EKF-SLAM with points and lines. *International Journal of Computer Vision*, 97(3), 339–368.
- Solla, M., Asorey-Cacheda, R., Nez-Nieto, X., & Riveiro, B. (2015). Combined approach of GPR and thermographic data through FDTD simulation to analyze masonry bridges: The evaluation of construction materials in a restored masonry arch bridge. In *2015 8th International Workshop on Advanced Ground Penetrating Radar* (pp. 1–4). Piscataway, NJ: IEEE. <https://doi.org/10.1109/IWAGPR.2015.7292641>
- Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). DeepFace: Closing the gap to human-level performance in face verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1701–1708). Piscataway, NJ: IEEE. <https://doi.org/10.1109/CVPR.2014.220>
- Tajdini, M. M., & Rappaport, C. M. (2013). Analytic analysis of ground penetrating radar wave scattering of reinforced concrete bridge decks. In *2013 IEEE International Geoscience and Remote Sensing Symposium*. (pp. 4066–4069). <https://doi.org/10.1109/IGARSS.2013.6723726>
- Tanzmeister, G., & Steyer, S. (2016). Spatiotemporal alignment for low-level asynchronous data fusion with radar sensors in grid-based tracking and mapping. In *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems* (pp. 231–237). Piscataway, NJ: IEEE. <https://doi.org/10.1109/MFI.2016.7849494>
- Tian, G. Y., Sophian, A., Taylor, D., & Rudlin, J. (2005). Multiple sensors on pulsed eddy-current detection for 3-D subsurface crack assessment. *IEEE Sensors Journal*, 5(1), 90–96. <https://doi.org/10.1109/JSEN.2004.839129>
- Wang, J., Li, G., Jiang, R., & Chen, Y. (2013). Real-time monitoring of the bridge cable health by acoustic emission and b-value analysis. In *2013 6th International Congress on Image and Signal Processing* (Vol. 3, pp. 1340–1345). Piscataway, NJ: IEEE. <https://doi.org/10.1109/CISP.2013.6743881>
- Wooden, D., Malchano, M., Blankespoor, K., Howardy, A., Rizzi, A. A., & Raibert, M. (2010). Autonomous navigation for BigDog. In *2010 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 4736–4741). Piscataway, NJ: IEEE.
- Yan, Y., Guldur, B., Yoder, L., Kasireddy, V., Huber, D., Scherer, S., ... Hajjar, J. F. (2016). Automated damage detection and structural modelling with laser scanning. In Paper presented at Proceedings of the Annual Stability Conference Structural Stability Research Council.
- Zhang, R., Zhong, W. D., Wu, D., & Qian, K. (2016). A novel sensor fusion based indoor visible light positioning system. In *2016 IEEE Globecom Workshops* (pp. 1–6) Piscataway, NJ: IEEE. <https://doi.org/10.1109/GLOCOMW.2016.7848823>
- Zhu, D., Guo, J., Cho, C., Wang, Y., & Lee, K. M. (2012). Wireless mobile sensor network for the system identification of a space frame bridge. *IEEE/ASME Transactions on Mechatronics*, 17(3), 499–507. <https://doi.org/10.1109/TMECH.2012.2187915>

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of the article.

How to cite this article: Gibb S, La HM, Le T, Nguyen L, Schmid R, Pham H. Nondestructive evaluation sensor fusion with autonomous robotic system for civil infrastructure inspection. *J Field Robotics*. 2018;35:988–1004. <https://doi.org/10.1002/rob.21791>