Juni Khyat
(UGC Care Group I Listed Journal)
ISSN: 2278-4632
Vol-12 Issue-07 No.03 July 2022

An Automated Bridge Assessment by means of Mobile Ground Robotics

Jayakrushna Rout [1], Satyajit Kumar Mahapatra [2], Jagadish Nayak [3], AMIT GUPTA[4]

[1, 2, 3] Gandhi Institute for Education & Technology, Baniatangi, Khordha, Odisha
[4]NM Institute of Engineering & Technology,Bhubaneswar,Odisha
jkrout@giet.edu.in, skmahapatra@giet.edu.in, jagadishnayak@giet.edu.in

Abstract: By overcoming several typical shortcomings relating to accessibility, repeatability, concealed fault identification, and quantification, the employment of mobile ground and aerial robots provides a potent way to supplement existing visual inspection practise. This research introduces a revolutionary ground robotic bridge inspection platform made up of a tough mobile platform with multiple calibrated and time-synchronized sensors and an onboard computer. The underside of standard concrete bridges, which is sometimes the most difficult to check, is proven to create high-quality 3D point clouds using this platform together with specialised localization and mapping software. Using measurements from terrestrial laser scanners, the accuracy of these maps is compared to the actual ground surface, and it is revealed that there is just a 1.3% scale inaccuracy altogether. When opposed to employing a terrestrial laser scanner, the maps from the suggested system may be created in real time while continually scanning the bridge, significantly cutting down on inspection time (TLS). In addition, an unique method for defect identification and quantification in semi-automated and completely automated point clouds is presented in this paper. With the former, inspectors may check the bridge remotely using accurate visual representations. In contrast to conventional inspections, the latter permits greater fault quantification accuracy while removing inspector subjectivity.

Introduction

Government agencies around the world are looking toward advanced inspection technologies that could help them mitigate the financial and societal risks associated with aging public infrastructure. Bridges pose a particularly high risk in many jurisdictions and are subjected to stringent inspection requirements through the application of standards. Such standards provide guidance to trained inspectors on how to inspect structures using various techniques, ranging from visual inspections to specialized inspections using equipment such as radar or ultrasound. Many jurisdictions (e.g., Ontario, Canada) require visual inspections to be performed every 2 years for all bridges with a minimum specified span length (Ministry of Transportation Ontario 2008) and more detailed inspections are undertaken as warranted by the condition of the bridge.

Visual assessment performed to uncover material defects, performance deficiencies, and maintenance needs for bridge elements often requires an inspector to gain good visual access to all elementlocations, including the underside of the deck. Where easy access is not possible, specialized equipment, such as boom lifts, scissor lifts,and so forth, are used to gain access to the underside components. While these techniques do allow an inspector sufficiently proximateaccess to the bridge underside to perform a visual inspection, theyoften require closing the bridge to traffic temporarily and require specialized training to conduct. Regardless of access issues, ensuring consistent quality, repeatability, and objectivity between inspections still remains a challenge with current visual inspection practices. Missing information, a lack of photographic evidence, and variations of up to 50% between inspectors, with only 68% of ratings lying within an acceptable deviation of 10% from the mean, are someof the key issues that persist with purely visual inspection methods (Moore et al. 2001).

Many of the shortcomings in visual inspections can be overcome using robotics and noncontact sensing technologies. They not only provide a means to gain access to critical and hard-to-reach bridge elements, for example, the underside of decks, but they canalso be used to make inspections repeatable and less prone to variations between inspections. By creating 3D maps in the field (maps refer to point cloud representations, which constitute the spatial coordinates of a structure in 3D), elements of interest can be automatically located on the structure in situ, which makes inspections of the same area repeatable and can also provide geometric information on identified defects. This paper presents an automated robotic ground vehicle equipped with multiple noncontact sensors that can be used to generate maps of bridges and enable automated defect detection and quantification.

Related Works and Contributions

The most prevalent method of robot-aided bridge inspection is the use of unmanned aerial vehicles (UAVs) for image collection. High-quality images of structurally significant areas that are otherwise

ISSN: 2278-4632
Vol-12 Issue-07 No.03 July 2022

not easily accessible to humans can be successfully captured using a UAV and visually assessed by an inspector (Gillins et al. 2016; Zink and Lovelace 2015; Moller 2008). UAVs have also been equipped with thermal cameras to assist in visual inspections (Ellenberg et al. 2016). Such UAV applications simply provide the inspector with a more efficient method to locate defects that would otherwise be difficult to access. Tagging of the images for localization of the defects must still be done manually and damage quantification is not possible from images alone.

Additional valuable information about the structure can be gathered by generating a 3D point cloud representation of structure, enabling the localization and quantification of the defects found in the images. The process of generating point clouds from monocular imagery is known as structure from motion (SfM), which has been applied to bridge modeling and inspection (Khaloo et al. 2018; Khaloo and Lattanzi 2017). The 3D SfM models with tagged defects are useful for tracking defects over time, but the accurate quantification or tracking of small changes in defects is strictly limited by the quality of the point cloud. This is a common issue with SfM-generated point clouds, as they tend to be noisy, often have misaligned objects, and even miss crucial areas when little texture is available in the images. Previous studies have acknowledged the quality issues of point clouds generated using SfM and suggested the use of other sensor types, for example, (Khaloo et al. 2018). Overall, SfM as a means to generate the point clouds of bridges is fraught with challenges when dealing with poor or variable lighting conditions, a lack of visual features to track in the images, and the absence of a global positioning system (GPS) signal for image capture automation and image pose initialization.

Bridge inspections can be further enhanced using robotics and noncontact sensing, for example, autonomous defect detection from image data. Cracks can be automatically detected from pixel intensity variations (Abdel-Qader et al. 2003; Adhikari et al. 2014; Yeum and Dyke 2015; Prasanna et al. 2016), as can delaminations, by detecting heat pattern variations from infrared images (Omar et al. 2018). For more information on other image-based defect detection for concrete and asphalt inspection, refer to (Koch et al. 2015). In the context of this paper, most of the work in the literature has been conducted to detect defects in the images, but images alone do not allow an inspector to determine scale in the defect measurements. Monocular images can provide size information relative to the pixels in the images; however, the pixel scale in world coordinates is unknown unless some reference scale is provided. To determine the size of a defect, each monocular image needs a reference scale. Some studies have attempted to extend image-processing techniques to determine scale automatically by utilizing SfM algorithms (Liu et al. 2016; Torok et al. 2014; Jahanshahi et al. 2013). These have shown some success but the same challenges of creating accurate SfM models, as previously discussed, remain apparent in these methods.

Another possible method for augmenting the structural inspections of bridges is to use high-quality stationary scanning equipment, such as terrestrial laser scanners (TLS), to generate higher-quality point cloud maps. TLS-generated point clouds can be used for condition assessment and defect detection (Law et al. 2018; Laefer et al. 2014; Turkan et al. 2016; Liu et al. 2011). The TLS has also been combined with images to improve defect detection (Valença et al. 2017), where such methods rely on the high-quality and dense maps generated by the TLS sensors. TLS scanning processes are often expensive and time consuming and do not offer the same advantages as mobile platforms, such as UAVs or ground vehicles. There have been a few studies involving sensors mounted on ground robotic platforms for bridge inspections, for example, (Kim et al. 2017, 2018), where a stop-and-go scan procedure is used, but the issue of bridge inspections is not addressed specifically in these studies. Similarly, a ground vehicle equipped with a wide array of sensors has been presented for bridge deck inspection in (La et al. 2013; Gucunski et al. 2017), but the results are limited to inspecting deck elements only, not the underside, and the process of map building and defect quantification has not been addressed.

Robust map building, rapid scanning from mobile platforms, and accurate defect detection and quantification from multiple sensor modalities can augment existing visual inspection practice significantly. This work introduces an unmanned ground vehicle (UGV) designed specifically for the purpose of bridge inspection. The UGV is equipped with various state-of-the-art robotic sensors and a mapping approach is proposed to rapidly and robustly create high-quality 3D maps with the ability to locate and quantify common defects in concrete bridges with minimal human input. The proposed approach allows for more accurate defect quantification, eliminates human subjectivity, and has the potential to reduce inspection time and eliminate accessibility challenges. The main contributions of this work are as follows: (1) a multisensor robotic inspection platform is presented along with a methodology for sensor synchronization and calibration; (2) a localization and mapping algorithm is presented, enabling continuous scanning for repeatable, high-quality 3D mapping of bridges; and (3) a methodology for semi- to fully automated defect detection and quantification is discussed and results are presented for several important structural defects common to the underside of reinforced concrete bridges. While the presented platform is ground based, the sensor package, calibration, mapping, and algorithmic tools for defect detection and quantification are flexible in the sense that they can be modified to work with UAVs as well. Employing a UGV offers some advantages over UAVs, such as the opportunity for higher computational power, greater payload, substantially lower risk to the sensors and fewer operating regulations.

Platform Development

A robotic platform specifically designed for ground-based bridge inspection is presented in this section. The mobility is made possible through a Husky A200 UGV (Clearpath Robotics, Kitchener, Ontario, Canada). This is a skid-steered vehicle capable of reaching 1 m=s velocity with 4 rubber pneumatic tires, a self weight of 50 kg, and a payload of 75 kg. The platform is equipped with an onboard computer for data collection, running the sensor drivers, controlling the UGV and real-time localization, mapping, and data processing. A sensor mounting system is used to accommodate all the sensors as shown mounted on the UGV in Figs. 1(a and b).

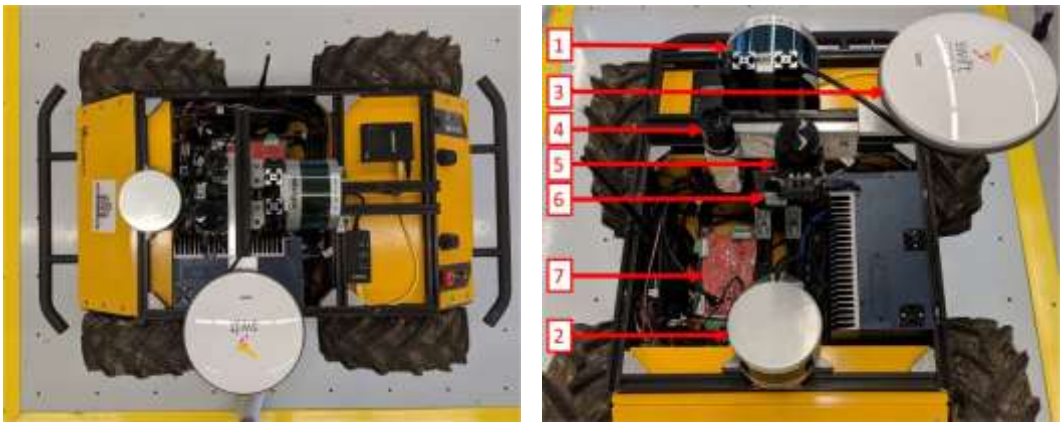
The major challenge of building a robotic 3D mapping system described herein is the sensor integration. The ability to properly integrate data from different sensors becomes a challenging task when using such a variety of sensor types and makes. In this section, a detailed description of the sensor integration process will be presented, including a description of the sensors and details on how they are synchronized and calibrated as one sensor unit.

Sensors

This UGV is equipped with the following sensors:

- Lidar: Two Velodyne VLP-16 light detection and ranging (lidar) sensors are used to map overhead bridge elements and for horizontal localization. The VLP-16 sensors have 16 rotating light beams and the vertical high-resolution lidar weighs 830 g with a field of view of 20°, while the horizontal lidar weighs 590 g with a field of view of 30°.

ISSN: 2278-4632
Vol-12 Issue-07 No.03 July 2022



(a) (b)

Fig. 1. Details of the sensors mounted on the UGV: (a) Plan view of the sensors and the computer mounted on the UGValong with the custom mount; and (b) labeled sensors: (1) vertical lidar, (2) horizontal lidar, (3) GPS antenna, (4) vision camera, (5) IR camera, (6) IMU, and (7) trigger board.

- GPS: Piksi Multi RTK GPS with antenna, produced by Swift Navigation. This GPS package also includes a base station.
- Vision camera: Ximea XiQ MQ013xG-E2 visible spectrum camera by Ximea. This is a USB 3.0 camera with 1,280 × 1,024 resolution, a maximum video rate of 60 frames per second, and a mass of 26 g. The camera is mounted vertically to capture images of the underside of bridges.
- Infrared (IR) camera: Flir Vue Pro infrared spectrum (7.5-13.5 μm wavelength) camera with a resolution of 640 × 512 pixels and a maximum frame rate of 30 Hz. This camera is also mounted vertically to capture the underside of the bridge being scanned.
- Inertial measurement unit (IMU): UM7 IMU made by Redshift Labs. This IMU has a built-in accelerometer, gyroscope, and magnetometer and estimates full 6 degrees of freedom (DOF) orientation. The accelerometer measures linear acceleration in 3 axes, the gyroscope measures angular velocity in about 3 axes, and the magnetometer measures the earth's magnetic field to determine an absolute heading estimate. The device weighs 11 g and outputs data up to a rate of 255 Hz.

These sensors are integrated in the Robot Operating System (ROS) on the onboard computer, which allows data collection and playback to be performed seamlessly.

Time synchronization between all the sensors is critical to estimate the location where the data was captured while the UGV is in motion, and is performed using a custom printed circuit board (PCB) (Fig. 2). A 1-Hz pulse per second (PPS) signal is generated by the GPS and sent to the Ximea and lidar through the PCB to trigger data capture. The system clock of the onboard computer is synchronized via the PPS signal and timestamped with the GPS time over a serial cable from the PCB. As a result, all the scanned data is captured at precise intervals and timestamped with the same time source. The IR camera is synchronized through software triggered by generating a pulse-width modulation (PWM) signal output corresponding to the PPS input using an Arduino microcontroller (Fig. 2). With the current configuration, all cameras capture images at 1 Hz, whereas the lidars use the 1-Hz signal for synchronization while capturing scans at 10 Hz. While a 1-Hz image capture rate was deemed sufficient for this application, higher rates can be achieved with some hardware modifications.

Sensor Calibration

The calibration procedure to fuse data from multiple sources consists of both intrinsic and extrinsic calibrations. Intrinsic calibration is required for both the vision and IR cameras, while the manufacture's intrinsics for the remaining sensors are used. The format of the intrinsic camera calibration results is further explained in the Map Postprocessing section. Camera intrinsics are calculated using the Matlab camera calibration toolbox (Bouguet 2015) for the

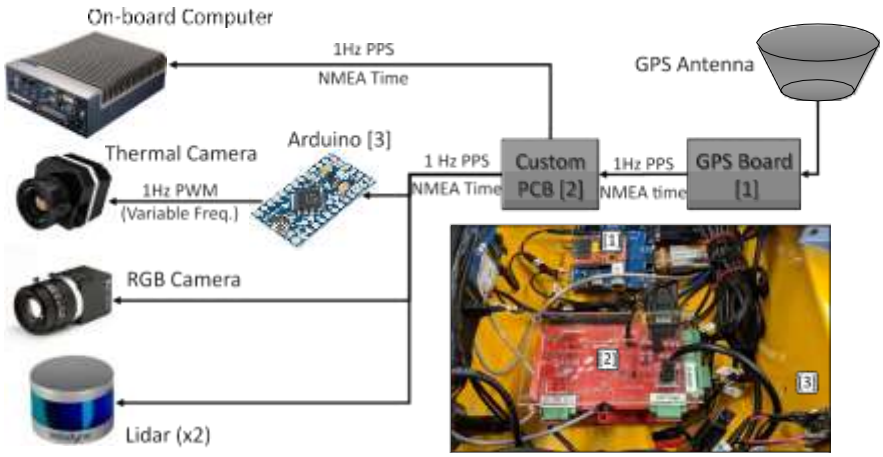


Fig. 2. Sensor synchronization diagram.

ISSN: 2278-4632
Vol-12 Issue-07 No.03 July 2022

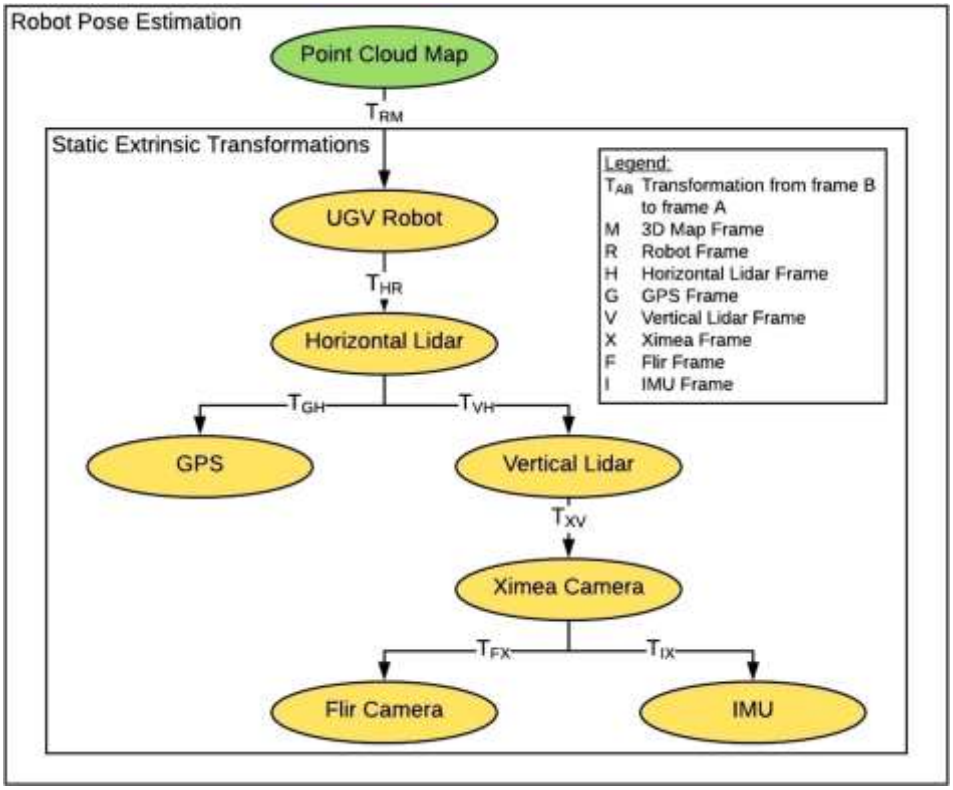


Fig. 3. Robot platform and sensors transformation tree.

calibration of a single camera using a checkerboard target. The intrinsic calibration of the Ximea camera uses 72 images of the checkerboard in Fig. 4(a). The mean reprojection error from calibration is 0.12 pixels. The intrinsic calibration of the IR camera uses the checkerboard in Fig. 4(b) as a target. This target can be used for thermal calibration as the flat black paint (high emissivity) and aluminum (low emissivity) have high contrast in emitted energy in the thermal band. The calibration uses 12 images of the thermal checkerboard taken with the IR camera, resulting in a mean reprojection error of 0.38 pixels.

The extrinsic calibrations consist of: (1) vertical to horizontal lidar; (2) GPS to horizontal lidar; (3) camera to vertical lidar; (4) IR camera to vision camera; and (5) IMU to vision camera. All extrinsic calibrations are used to convert data between the sensor coordinate frames. Each extrinsic calibration calculates the rigid transformation, represented using the SE(3) Euclidean group, that describes the rotation and translation between the two respective coordinate frames (Blanco 2014). For example, points in a lidar scan can be converted between frames A and B using Eq. (1). T_{AB} is the SE(3) transformation matrix to convert a point from frame B to frame A. r_{11} to r_{33} represent the values that make up the 3D rotation matrix. t_x, t_y, and t_z represent the x, y, and z translations, respectively

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_A = T_{AB} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_B ; \quad T_{AB} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Extrinsic calibrations all take the form of T_{AB}. These calibrations are first done manually from a 3D point cloud of the sensors scanned using FARO's Quantum FaroArm and then refined computationally where required. The manual measurements are primarily used for the estimation of t_x, t_y, t_z, as the rotations are estimated to be right angles due to the construction of the sensor mounts. A summary of the extrinsic calibration tree that enables conversion between all coordinate frames is provided in Fig. 3.

The extrinsic calibration between the vision and the IR cameras is refined using the Matlab stereo camera calibration toolbox (Bouguet 2015). The same checkerboard for the calibration of the IR camera intrinsics is also used for stereo calibration [Fig. 4(b)], as it displays the checkerboard pattern in both the IR and vision images taken synchronously. In total, 13 image pairs are used in calibration to achieve a mean reprojection error of 0.71 pixels. In this case, fewer images are used compared to the intrinsic calibrations due to the challenges associated with obtaining good-quality images where the checkerboard is detected well in both the thermal and vision images.

The vision camera to lidar calibration is refined using the square target shown in Fig. 4(c). Scans are taken with the camera and the vertical lidar directed at the stationary square target. The points in the 3D point cloud generated from the individual lidar scans are then projected onto the image plane using the camera to lidar transformation matrix, the camera intrinsics matrix, and the pinhole camera model. These points are overlaid onto the image to visually determine the accuracy of the extrinsic calibration. The transformation matrix is then adjusted for rotation and translation until the location of the target in the lidar scan visually aligns precisely with the location of the target in the image.

Mapping

In order to construct a 3D representation of the structure being inspected, two separate estimation processes need to be performed simultaneously. An aggregated map of the unknown environment

ISSN: 2278-4632
Vol-12 Issue-07 No.03 July 2022

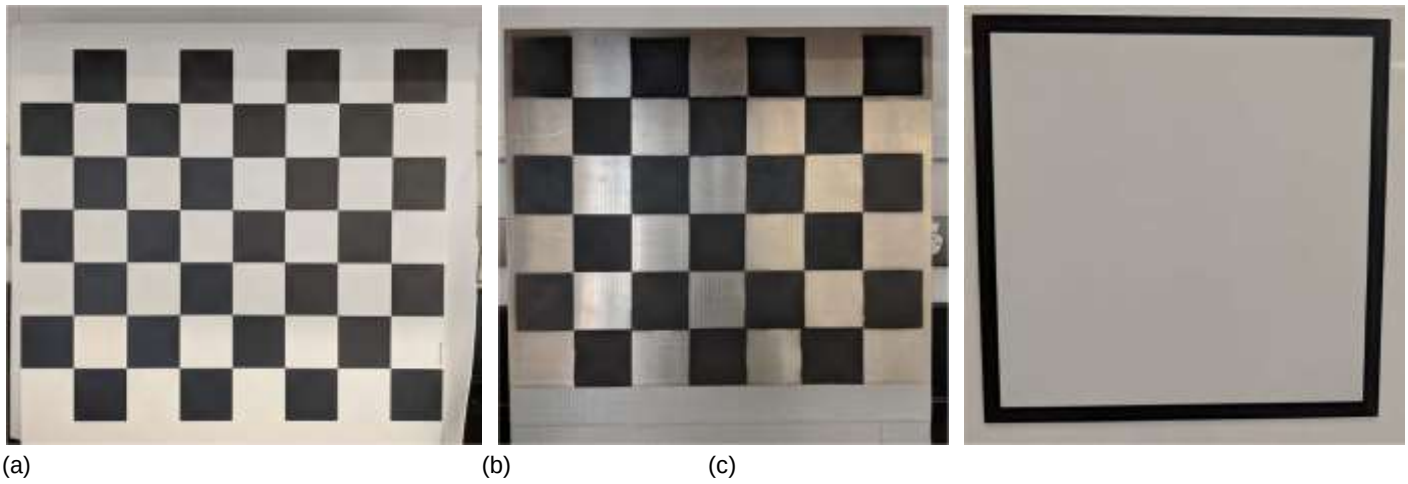


Fig. 4. Calibration targets: (a) vision camera target; (b) IR camera target; and (c) Lidar target.

must be created by combining measurements taken from multiple poses and the pose [usually expressed as $(x, y, z, \phi, \theta, \psi)$, where (x, y, z) represents position, and (ϕ, θ, ψ) represents the orientation in the roll, pitch, and yaw angles] of the robot at each measurement location must be estimated in order to correctly register each set of measurements in a common reference frame. This process of localizing the robot and combining measurements into a map is well studied in the robotics literature, and referred to as simultaneous localization and mapping (SLAM) (Durrant-Whyte and Bailey 2006). SLAM is a challenging problem because localization requires knowledge of the map, while map building requires knowledge of where the robot is within this map (Siegwart and Nourbakhsh 2004). Therefore, errors in the robot pose estimate will propagate to errors in the map and vice versa. Such errors will continue to grow over time unless global measurement sources with bounded errors are incorporated, such as GPS measurements or a return to previously visited locations (loop closure). Despite the existence of a large volume of related literature (Durrant-Whyte and Bailey 2006; Durrant-Whyte and Bailey 2006), SLAM continues to be an active research area with the most recent progress being in task-aware SLAM and semantic representations of environments (Cadena et al. 2016).

Localization and mapping are discussed separately next, while recognizing that these two processes are occurring simultaneously in the SLAM implementation. Localization refers to the determination of the robot's pose relative to a predefined set of landmarks. In this paper, an extended Kalman filter (EKF) approach is adopted. The EKF uses a two-step prediction-correction process to estimate the states of a nonlinear system, including the robot and the environment (map) features (Moore and Stouch 2014). This is a common approach used in robotics, where state predictions are made based on a kinematic motion model (nonlinear), and predicted states are subsequently corrected based on measurements (i.e., measurement updates) from multiple sensors integrated on the robot (Barfoot 2018). The states of the EKF include the robot pose and its derivatives. Additionally, the state vector is augmented using the accelerations of the robot in the three Cartesian coordinates and represented by an augmented state vector, as shown in Eq. (2):

$$X_t = [x_t \quad y_t \quad z_t \quad \phi_t \quad \theta_t \quad \psi_t \quad \dot{x}_t \quad \dot{y}_t \quad \dot{z}_t \quad \dot{\phi}_t \quad \dot{\theta}_t \quad \dot{\psi}_t \quad \ddot{x}_t \quad \ddot{y}_t \quad \ddot{z}_t]^T \tag{2}$$

All elements of the state vector and the motion model are defined in the discrete domain and not in the continuous domain. In the EKF implementation, four sensors provide measurements: the GPS, IMU, wheel encoders, and lidars. GPS measurements are used only during the initial tests as it is expected that GPS signals will not be available while mapping the underside of bridges. Neglecting the GPS measurements in the formulation of the EKF, Eq. (3) shows all the measurements that are incorporated into the EKF localizer

$$Y_t = [y_{\omega_x} \quad y_{\omega_y} \quad y_{\omega_z} \quad y_{v_x} \quad y_{v_y} \quad y_x \quad y_y \quad y_z \quad y_\phi \quad y_\theta \quad y_\psi]_t^T \tag{3}$$

The detailed description of the sources for each of the preceding measurements along with the associated measurement models are presented in the subsequent section.

1. IMU: The IMU natively reports measurements for acceleration, angular velocity, and magnetic field strength; however, only angular velocities (denoted by ω_x, ω_y, ω_z) are used to update the $(\dot{\phi}, \dot{\theta}, \dot{\psi})$ states. Since these measurements are direct measurements of the robot state using another coordinate frame, they simply need to be converted between the IMU frame and the map frame. The resulting measurement model is presented in Eq. (4)

$$\begin{bmatrix} y_{\omega_x} \\ y_{\omega_y} \\ y_{\omega_z} \end{bmatrix}_t = h_I(X_t) = \begin{bmatrix} \dot{\phi}_t - \dot{\psi}_t \sin\theta_t \\ \dot{\theta}_t \cos\phi_t + \dot{\psi}_t \cos\theta_t \sin\phi_t \\ -\dot{\theta}_t \sin\phi_t + \dot{\psi}_t \cos\theta_t \cos\phi_t \end{bmatrix} \tag{4}$$

2. Wheel odometry: The UGV uses wheel encoders to measure instantaneous wheel velocity, which can then be used to calculate change in position (v_x, v_y). The UGV driver automatically outputs the linear velocity in x, y and the rotation about the z axis using the wheel encoder data, therefore these measurements can be fused directly in the EKF. Again, these measurements are in the robot frame and need to be converted between the robot frame and the map frame, yielding the measurement model presented in Eq. (5). Only the velocity measurements are used in the EKF for the current application

$$\begin{bmatrix} y_{v_x} \\ y_{v_y} \end{bmatrix}_t = h_w(X_t) = \begin{bmatrix} \dot{x}_t \cos\psi_t + \dot{y}_t \sin\psi_t \\ -\dot{x}_t \sin\psi_t + \dot{y}_t \cos\psi_t \end{bmatrix} \tag{5}$$

3. Scan registration: Scan registration is performed with the scans from both lidars and provides estimates of the relative position

ISSN: 2278-4632
Vol-12 Issue-07 No.03 July 2022

between the robot frame at one scan to the map frame. This 4×4 transformation matrix is then converted to the pose and orientation measurements of the robot relative to the map frame $\delta y_x \quad y_y \quad y_z \quad y_\phi \quad y_\theta \quad y_\psi \rho$ Since these are direct measurements of the robot state, the measurement model is as shown in Eq. (6). Scan registration will be discussed in more detail in the following section

$$\frac{1}{2} y_x \quad y_y \quad y_z \quad y_\phi \quad y_\theta \quad y_\psi \Big]_t^T \frac{1}{4} h_s \delta X_t \rho \frac{1}{4} \frac{1}{2} x_t \quad y_t \quad z_t \quad \phi_t \quad \theta_t \quad \psi_t \Big]^T \tag{6}$$

Since the scan registration process is computationally expensive, the scan registration measurements are incorporated at a much lower rate and continuous updating without scan registration is performed between such updates. The resulting multirate EKF is shown in Eq. (7) when scan registration measurements are not available and in Eq. (8) otherwise

$$h \underset{1}{\delta} X \underset{t}{\rho} \frac{1}{4} \quad h_t \delta X_t \rho \tag{7}$$

$$h_2 \delta X_t \rho \frac{1}{4} \begin{matrix} 2 \\ 6 \\ 4 \end{matrix} \begin{matrix} h_t \delta X_t \rho \\ h_w \delta X_t \rho \\ h_s \delta X_t \rho \end{matrix} \begin{matrix} 3 \\ 7 \\ 5 \end{matrix} \tag{8}$$

For the EKF used in this paper, the motion model is based on an omnidirectional robot in 3D (Campion et al. 1996). For the sake of brevity, a simplified model for 2D motion is shown in Eq. (9)

$$\begin{matrix} 2 \\ 6 \\ 6 \\ 6 \\ 6 \\ 6 \\ 6 \\ 4 \end{matrix} \begin{matrix} x_t \\ y \\ \psi \\ \dot{x}_t \\ \dot{y}_t \\ \dot{\psi} \\ \ddot{x}_t \\ \ddot{y}_t \end{matrix} \begin{matrix} 3 \\ 7 \\ i \\ 7 \\ 7 \\ i \\ 7 \\ 5 \end{matrix} \frac{1}{4} g \delta X_{t-1} \rho \frac{1}{4} \begin{matrix} 2 \\ 6 \\ 6 \\ 6 \\ 6 \\ 6 \\ 6 \\ 4 \end{matrix} \begin{matrix} x_{t-1} \rho \delta \dot{x}_{t-1} \cos \psi_{t-1} - \dot{y}_{t-1} \sin \psi_{t-1} \rho dt \rho 0.5 \delta \ddot{x}_{t-1} \cos \psi_{t-1} - \ddot{y}_{t-1} \sin \psi_{t-1} \rho dt^2 \\ y_{t-1} \rho \delta \dot{x}_{t-1} \sin \psi_{t-1} \rho \dot{y}_{t-1} \cos \psi_{t-1} \rho dt \rho 0.5 \delta \ddot{x}_{t-1} \sin \psi_{t-1} \rho \ddot{y}_{t-1} \cos \psi_{t-1} \rho dt^2 \\ \psi_{t-1} \rho \dot{\psi}_{t-1} dt \\ \ddot{x}_{t-1} dt \\ \ddot{y}_{t-1} dt \\ \dot{\psi}_{t-1} \\ \ddot{x}_{t-1} \\ \ddot{y}_{t-1} \end{matrix} \begin{matrix} 3 \\ 7 \\ 7 \\ 7 \\ 7 \\ 7 \\ 7 \\ 5 \end{matrix} \tag{9}$$

The accuracy of the prediction step in state estimation could be improved by using a motion model that captures the skid-steer kinematic behavior of the UGV (Goorts et al. 2017). For this application, however, the 3D omnidirectional motion model was deemed sufficient.

The mapping process in this paper relies on two subprocesses, the scan registration, and EKF localization. Scan registration is the process of comparing one scan to previous scans to determine a transformation between those instances. In this work, each incoming scan is compared to the point cloud map being produced, as opposed to comparing between subsequent scans, which increases the robustness of the mapping process as each new scan is matched against a more dense point cloud. Two different scan registration approaches are tested in this paper. Both approaches rely on the iterative closest point (ICP), with slight variations. The ICP is a least-squares optimization problem where the cost function being minimized is the summation of some distance metric between points on the new cloud to points on the reference cloud (Besl and McKay 1992; Barfoot 2018). This allows for the transformation (translation and orientation) to be estimated between each new cloud and the reference cloud. The general ICP approach involves matching each point on the new cloud to a point on the reference cloud, followed by the determination and application of the transformation that minimizes the sum of the distance measures. This procedure is iterated until some threshold measure is reached (Besl and McKay 1992).

The first method of the ICP in this paper is the point to point (PtP) ICP, where the sum of the Euclidean distance between points on the new cloud to the nearest points on the reference cloud is used in minimization. This is expressed in Eq. (10):

$$T_{ML;t} \quad \frac{1}{4} \arg\min_{T_{ML;t}} \Psi \delta M_{t-1} \; ; S_t \rho \frac{1}{4} \arg\min_{T_{ML;t}} \frac{1}{N_s} \sum_{i \frac{1}{4} 1}^{N} \| m_i - T_{ML;t} \; s_i \|^2 \tag{10}$$

where N_s = number of scan points, m_i and s_i = ith corresponding map and scan points in the set of map points (M_{t_1}) and scan points (S_t) and $T_{ML;t}$ = transformation from the lidar frame to the map frame at time t. The second method is commonly referred to as the point to plane (PtPl) ICP, where the cost-minimizing metric is the distance between points on the new cloud to planes on the reference cloud (Rusu and Cousins 2011). The planes in this algorithm are defined by local clusters of points.

The pseudocode for the SLAM algorithm using PtP ICP is provided in Figs. 5 and 6 shows the process in the form of a flowchart. This is a special variation of the traditional SLAM solution, such as the one given in (Thrun et al. 2004). Given the motion model, measurement model, and estimates for the motion and measurement noise covariances, the robot pose and covariance are estimated at each time step, while simultaneously producing a point cloud map. The ICP algorithm is implemented using the Point Cloud Library (PCL) (Rusu and Cousins 2011). Each incoming scan is registered against the map being constructed and that transformation is used as an EKF measurement update, then used to transform the scan into the map frame, which is then added to the map. The registration is also initialized with the most recent estimate of the transform between the lidar frame and the map frame (based on the EKF output).

Preprocessing of the scans is also performed as needed using various filtering techniques to help improve the map building

ISSN: 2278-4632
Vol-12 Issue-07 No.03 July 2022

SLAM Algorithm

INPUT:

X_{t-1} : robot pose at previous timestep, in map frame
Σ_{t-1} : pose covariance at time t - 1
Q_t, R_t : motion & measurement covariance (assume additive Gaussian disturbances)
y_t : measurements at time t (wheel odometry, IMU change in orientation, GPS pose)
S_t : set of points at time t, from one scan of both lidars, in the horizontal lidar frame
M_{t-1} : set of map features (points) at time t - 1, in the map frame
T_{LR} : static transform from robot frame to lidar frame (horizontal lidar)

OUTPUT:

$X_{t,m}$: current robot pose (at time t), in map frame
M_t : set of map features (points) at time t
Σ_t : pose covariance at time t

1: **while** true **do**
2: Perform prediction update:
3: $G_t \leftarrow \frac{\partial}{\partial X} g(X)\Big|_{X=X_{t-1}}$
4: $\overline{X_t} \leftarrow g(X_{t-1})$
5: $\overline{\Sigma_t} \leftarrow G_t \Sigma_{t-1} G_t^T + Q_t$
6:
7: **if** $S_t = \{\}$ **then** ▷ no scan registration
8: Perform measurement update:
9: $H_t \leftarrow \frac{\partial}{\partial X_t} h(X)\Big|_{X=X_t}$
10: $K_t \leftarrow \overline{\Sigma_t} H_t^T (H_t \overline{\Sigma_t} H_t^T + R_t)^{-1}$
11: $X_t \leftarrow \overline{X_t} + K_t(y_t - h_1(\overline{X_t}))$ ▷ using measurement model without registration
12: $\Sigma_t \leftarrow (I - K_t H_t)\overline{\Sigma_t}$
13:
14: **else if** $M_t = \{\}$ **then** ▷ If map is empty: transform scan & add to map
15: $M_t \leftarrow T_{MR,t}\, T_{RL}\, S_t$
16: **else** ▷ perform scan registration
17: $T_{ML,t} \leftarrow \arg\min_{T_{ML,t}} \Psi(M_{t-1}, S_t)$
18: $T_{MR,t} \leftarrow T_{ML,t}\, T_{LR}$
19: $M_t \leftarrow T_{MR,t}\, T_{RL}\, S_t$
20: Perform update (lines 10-13) with $h_2(\overline{X_t})$ and $T_{MR,t}$
21: **end if**
22: **end while**

Fig. 5. SLAM algorithm pseudocode.

and ICP. For example, a voxel grid filter is used to downsample the point clouds prior to the ICP. Given the higher point density with proximity to the sensor, the ICP tends to give higher weights to the fitness of the scan near the lidar. A voxel grid filter helps to smooth the point density throughout the scan, thus alleviating the aforementioned problem. Other techniques are used to further improve the ICP, including cropbox filters to remove ground points, vegetation, and points corresponding to the robot or operator, as well as a dynamic radius outlier filter to remove outlier points in the scan while taking into consideration the density variation in the scan data (Charron et al. 2018).

Mapping Results

Mapping tests were performed at three locations. The first location was the University of Waterloo campus and the next two locations were select concrete bridges in the Region of Waterloo. The three locations were selected to test the accuracy and robustness of the mapping and defect detection algorithms. For mapping, the campus location provided an easy localization case that allowed for confirmation of the SLAM approach. The first bridge location provided a short-range mapping case within a difficult mapping environment, whereas the second bridge allowed for large-scale mapping of a bridge with distant features. The following subsections will discuss the site locations in more detail and present the mapping results.

University of Waterloo Campus

The first test location was the University of Waterloo campus between the Engineering 3 and Centre for Environmental and Information Technology (EIT) buildings, as shown in Fig. 7. This test area is relatively open, enabling the UGV to receive GPS data. The location provides flat ground and a heavily structured environment for the EKF and ICP to perform well. These tests are used to confirm the functionality of the localization and mapping modules. With this data set, the results from both the PtP and PtPl ICP yield similar findings. The mapping results without utilizing the GPS measurements for the updates are shown in Fig. 8, depicting a top-down view of the map with a callout showing an isometric view of a portion of the map. Large detailed maps were produced where

ISSN: 2278-4632
Vol-12 Issue-07 No.03 July 2022

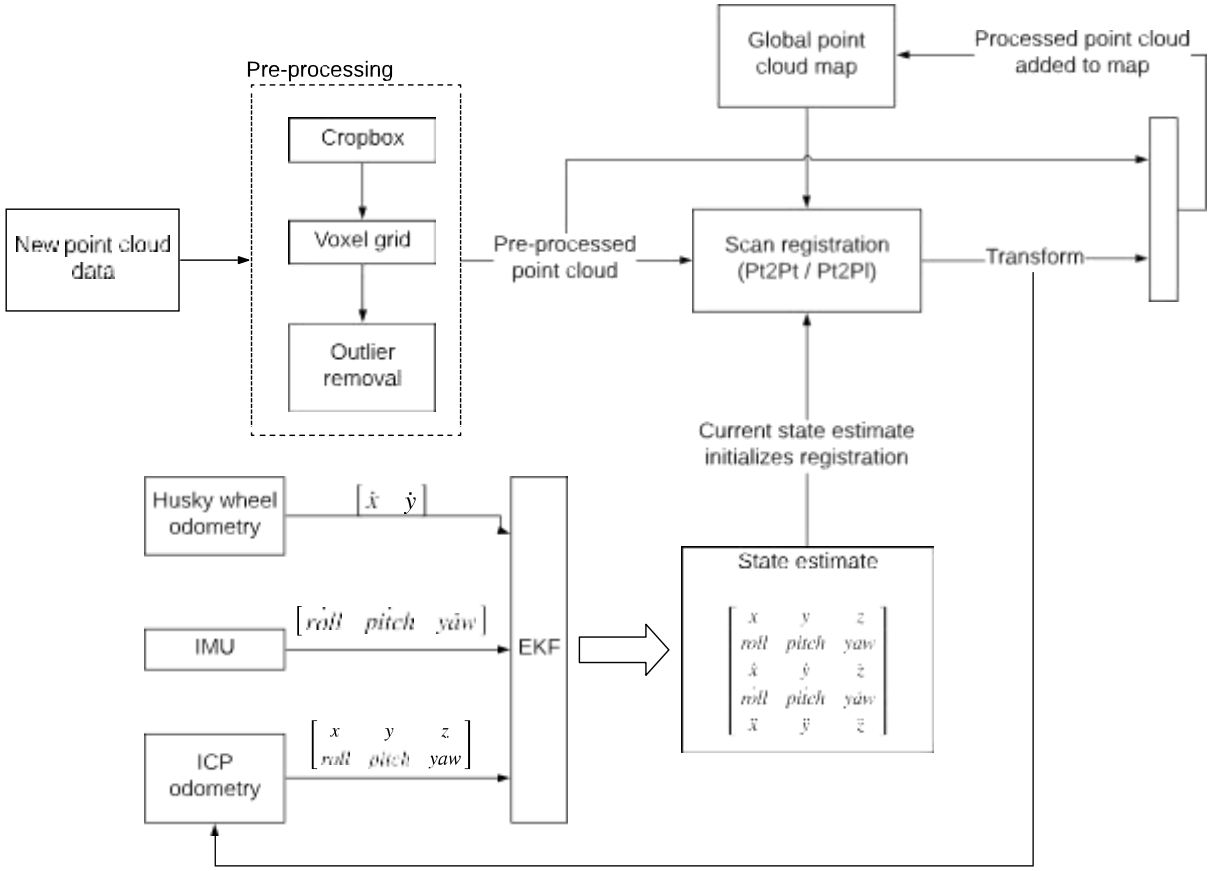


Fig. 6. Conceptual overview of SLAM algorithm.

the robot was able to accurately track its pose and map the environment, despite traveling large distances with many changes in orientation. The results without GPS show some error buildup over time due to the drift in the ICP and other measurement sources, which is corrected using GPS if available. Fig. 7 shows this map overlaid onto satellite imagery of the site to quantitatively assess the quality of the map. These results show that accurate short-range mapping without GPS is achievable; however, the use of GPS when available can result in more accurate large-scale maps by correcting the pose intermittently.

Conestogo River Bridge

The second test location is the Conestogo River bridge, situated at the intersection of Northfield Drive and the Conestogo River in Waterloo, Ontario. This bridge is a four-span cast-in-place concrete girder bridge, with three interior concrete piers situated within the Conestogo river. This bridge was constructed in 1960 and measures approximately 108 m in length and 4 m in height at its lowest location. The southernmost bay of the bridge was scanned as this area is easily accessible for the UGV. The mapping was done by driving the UGV under the bridge in a straight line directly under one of the girders to simulate a girder inspection. For this data set, the ICP PtPl performs significantly better compared to the PtP method. The PtP method creates significant drift in the mapping and this drift was eliminated with the implementation of the PtPl. The drift, which did not occur in the first location, can be explained by the lack of good distant features to localize against. The resulting point cloud using the ICP PtPl is shown in Fig. 9. Even without the GPS, an accurate map of the bridge is created with little observable error or drift in the final map. Further analysis on the accuracy of this map will be presented in the following section.

Fairway Bridge

The third test site is a newer bridge located at the intersection of Fairway Road North and the Grand River in Kitchener, Ontario. This is a four-span concrete box girder bridge constructed in 2012 that is approximately 250 m in length and 9 m in height (at its lower location). This bridge is significantly longer and higher than the second test location. The bridge and the results obtained from the scan are shown in Fig. 10. The results from this test site also reinforce the robustness and scalability of the robot platform and the mapping algorithm. This test case shows that environmental conditions similar to those of the Conestogo River bridge can still result in large-scale, low-drift maps. In this case, mapping the distant features over a much greater distance was just as successful, demonstrating that this procedure can be applied to a variety of bridge environment conditions.

Evaluation Metrics

A detailed evaluation of the scans and relevant bridge features for the Conestogo test site are presented in this section. The accuracy of the UGV-generated point cloud for this test site is evaluated in a variety of ways, both qualitatively and quantitatively. The 3D visualization of the point cloud enables the detection of objects of interest, such as concrete spall areas or cracks. The detected objects in the point cloud are then compared in terms of shape, size, and location to the visual inspection results. Quantitative evaluation is performed by comparing the UGV-generated point cloud against a reference point cloud constructed using a Faro Focus^M laser scanner. This system uses a rotating mirror to generate point clouds with full 360° horizontal coverage and vertical range of 305°. The scanner is capable of measuring 488,000 points per second at a

ISSN: 2278-4632
Vol-12 Issue-07 No.03 July 2022

Fig. 7. Satellite view of first test location with overlay of map created using the described SLAM algorithm. (Imagery @2018 Google, Map data @2018 Google.)

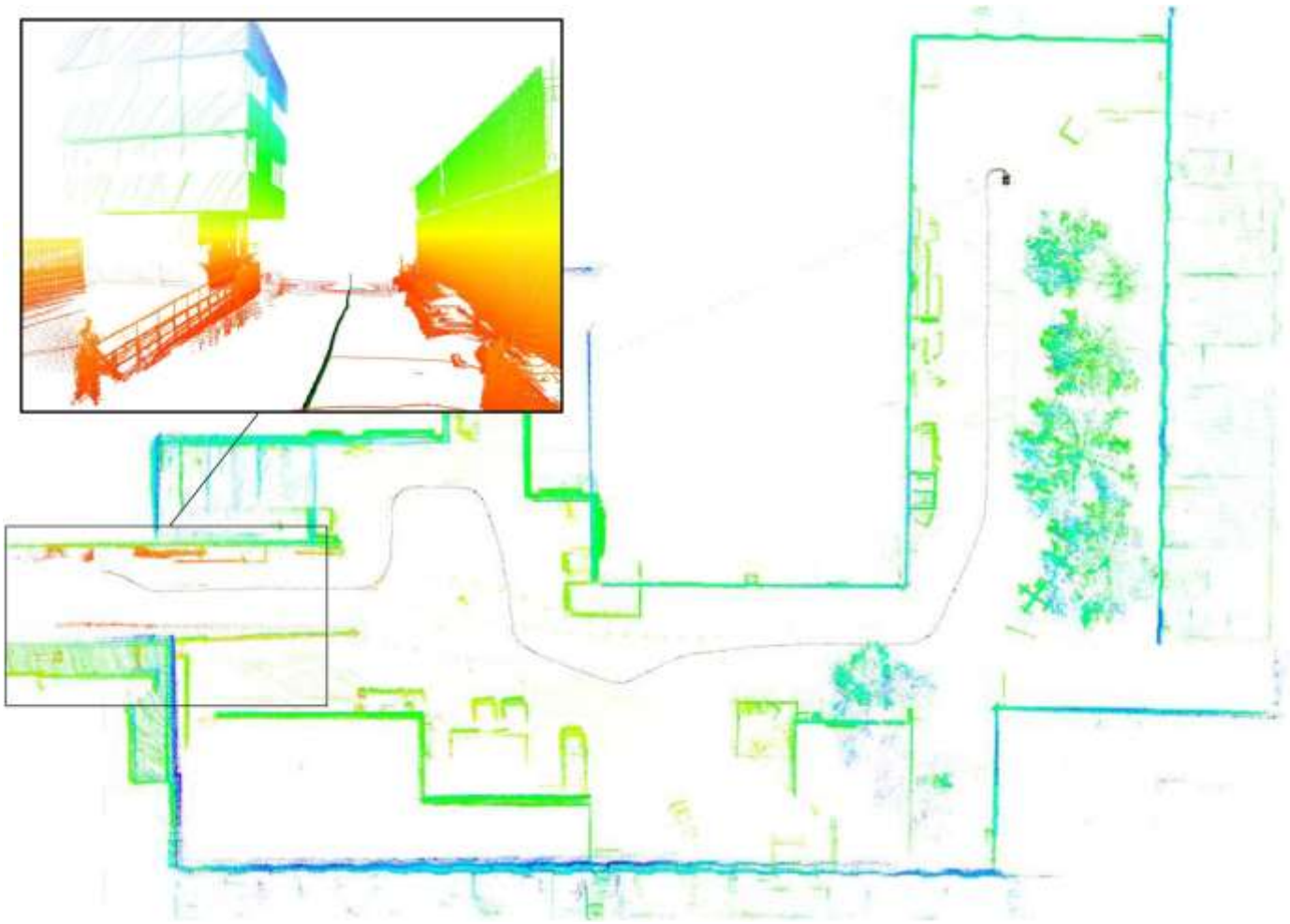


Fig. 8. Map created outdoors using the described SLAM algorithm at first site location.

ISSN: 2278-4632
Vol-12 Issue-07 No.03 July 2022

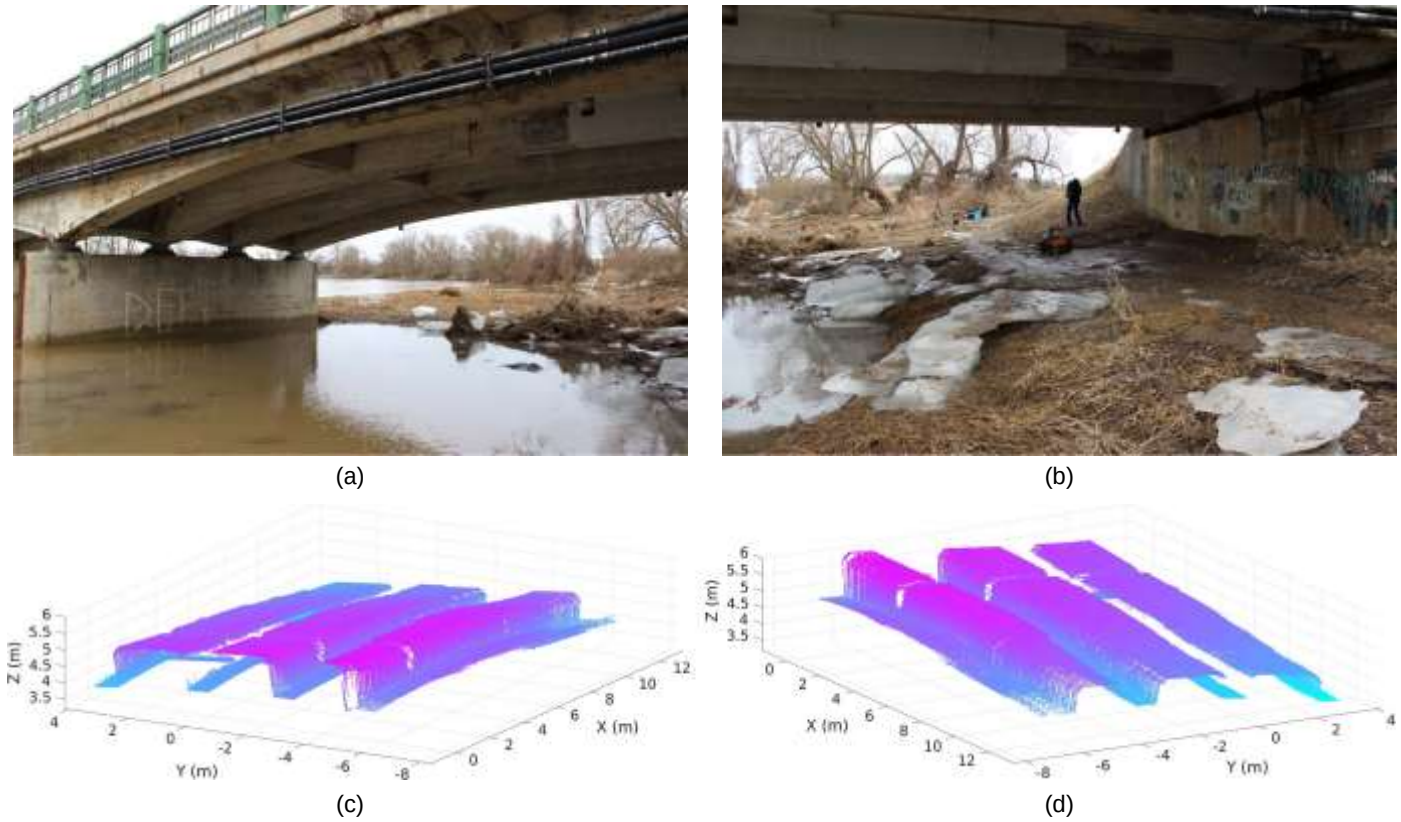


Fig. 9. Conestogo River bridge images and mapping results: (a) isometric view of bridge soffit, looking northeast (image by Nicholas Charron); (b) isometric view of bridge deck soffit, looking southeast (image by Nicholas Charron); (c) bridge soffit 3D map, looking north; and (d) bridge soffit 3D map, looking south.

maximum distance of 70 m with ranging error of 3 mm measured at 25 m. A rendering of the reference point cloud generated using the Faro Focus^M is shown in Fig. 11.

Although the laser scanner achieves a high level of accuracy, this method of point cloud generation is time consuming and is associated with large computational demands. The Faro Focus^M scan of the Conestogo test site required approximately 2.5 hours to complete scans from 6 different locations and 1 h of semimanual postprocessing. In comparison, the same scan using the proposed UGV system was completed in approximately 10 min with the results automatically processed within minutes.

A wide variety of techniques exist for the quantitative evaluation of point clouds, most of which can be generally categorized as control point approaches, subset approaches, or full cloud approaches (Lehtola et al. 2017). The control point approaches typically involve comparing the Euclidean distance between two or more features in each point cloud. The subset and full cloud approaches apply various metrics to compare aspects such as local noise or density to a portion of the cloud and entire cloud respectively (Lehtola et al. 2017). Since only a portion of the clouds overlap in this case, full cloud metrics are rendered unsuitable. A control point method and a subset approach are used to evaluate the scale accuracy and local noise level, respectively.

After registering the two point clouds, a total of 100 evenly spaced cross sections are cut along the length of the scanned portion of the bridge. Each cross section has a thickness of 20 mm, meaning all points falling within ± 10 mm of each cross-sectional plane are grouped into subsets. Fig. 12 compares the cross-sectional geometry of the reference and UGV-generated point clouds for three different cross sections, namely S1, S2, and S3, corresponding to the start, middle, and end of the first bay.

Although the UGV-generated point cloud shows higher noise levels compared to the reference cloud (represented graphically by surface thickness), the cross-sectional geometries of the two clouds are consistent throughout the length of the bridge bay. To quantitatively compare the scale of the two clouds, the girder depth is calculated at each cross section. For the UGV-generated cloud, the girder depth was measured to the centroid of the surface. Table 1 summarizes the maximum and average error for the inspected girder (directly above the UGV) and an adjacent girder (associated with a sparser set of measurements associated with the same pose). An average scale error of 1.3% for the inspected girder is indicative of the relative accuracy in mapping and defect detection using the UGV-generated point cloud. For example, when using the UGV-generated cloud, one can expect error on the order of only 1.3% relative to the FARO Focus^M-generated cloud.

The local noise level in a 3D point cloud can be approximated based on the residual between a given point and the best-fitting plane to the neighboring points (Lehtola et al. 2017; Khaloo et al. 2018). To compare the noise level between the two point clouds, subsets are extracted from two different locations: the underside of a girder and the side of a girder. In each case, the subset spans approximately 2 m. A best-fit plane is computed for each subset and the residual distance from each point to the plane is computed. The maximum and average residuals for each subset and each cloud are summarized in Table 2.

The local noise for the UGV-generated cloud is higher than the reference cloud, which can be a direct result of two sources of error:

ISSN: 2278-4632
Vol-12 Issue-07 No.03 July 2022

1. errors in the localization, ICP, and calibrations, or 2. errors in the sensor due to accuracy limitations (compared to the more accurate TLS scanner). The exact source of this error is still unknown. It is expected that more advanced SLAM algorithms and more sophisticated calibration procedures will lead to improved mapping and in turn will reduce noise. Nevertheless, the calculated local noise levels herein are considered acceptable for inspection purposes, as the primary objective is to demonstrate a platform to augment visual inspections. While the scale accuracy directly affects the quantification of defects, such as crack lengths and spall areas, the noise

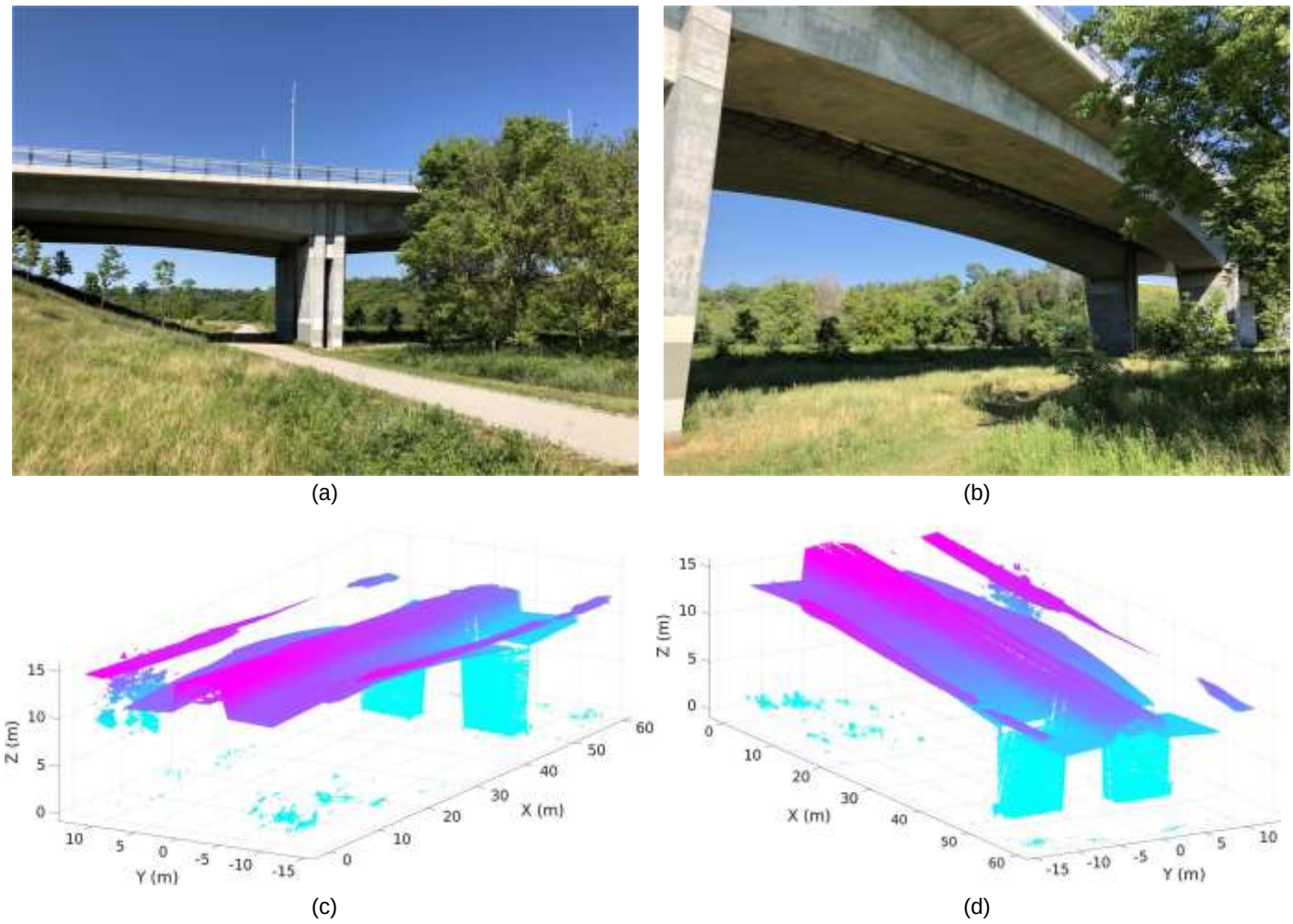


Fig. 10. Fairway Road bridge images and mapping results: (a) isometric view of bridge soffit, looking east (image by Nicholas Charron); (b) isometric view of bridge soffit, looking southeast (image by Nicholas Charron); (c) bridge soffit 3D map, looking southeast; and (d) bridge soffit 3D map, looking northeast.

Fig. 11. Scan results of Conestogo River bridge using the Faro Focus.

ISSN: 2278-4632
Vol-12 Issue-07 No.03 July 2022

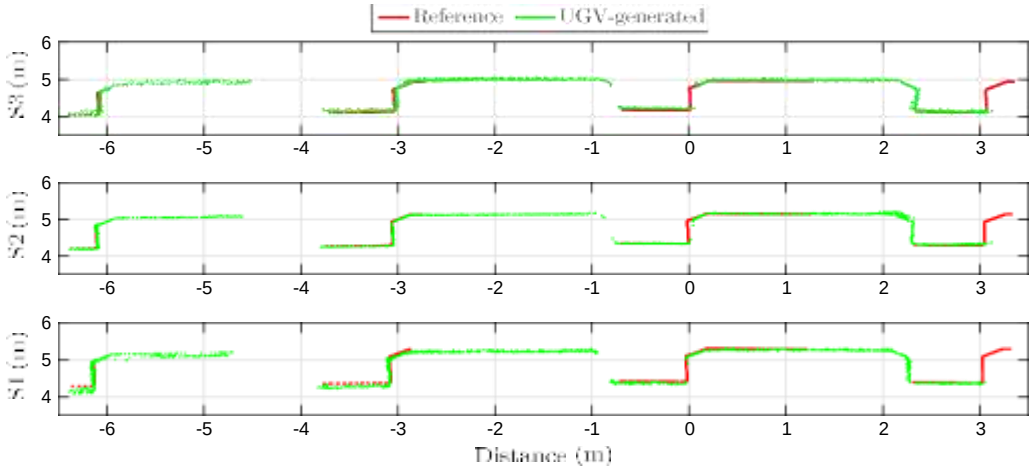


Fig. 12. Cross-section comparison showing the UGV-generated Conestogo River bridge map versus the map generated by the Faro Focus.

Table 1. Point cloud scale comparison

Metric	Inspected girder		Adjacent girder	
Maximum error	44.5 mm	(5.7%)	83.2 mm	(10.7%)
Average error	10.2 mm	(1.3%)	33.6 mm	(4.27%)
Standard deviation	8.6 mm	—	18.6 mm	—

Table 2. Point cloud noise level comparison

	Underside of girder		Side of girder	
Platform	Average residual (mm)	Maximum residual (mm)	Average residual (mm)	Maximum residual (mm)
Faro Focus 3D	−1.89	5.64	0.99	5.29
UGV	−5.47	30.52	11.17	40.34

level presents challenges when using the point cloud to calculate volumes, which is not a primary objective in visual inspections.

Map Postprocessing

Using the final point cloud map of a bridge combined with the image data, calibration results, and a full robot trajectory, the data can be combined into a format that can assist bridge inspectors. First, using the visual images, the color information can be used to color the point cloud. This colorization enables two major improvements: inspectors can now perform remote inspections using the colored map and the results from the automated defect detection can also be validated by observing a realistic rendering of the structure. The second step to the proposed bridge inspection procedure is to perform the defect detection on the images and transfer that data onto the point cloud for quantification and tracking. This step can be done with both sets of images to detect surface and subsurface defects. The following sections will discuss these two aspects in detail.

Point Cloud Colorization

The image data obtained from the vision and IR cameras during the scan can be used to colorize the 3D point cloud map of the bridge. The color information can be transferred to the cloud using the continuous 6 DOF pose estimate from the SLAM process and the camera positions relative to the robot from the calibration process. The transformation between the robot frame and the map frame is exported automatically during the mapping process each time an image is captured. The diagram in Fig. 3 shows the transformation tree for this process.

To colorize the cloud, the transformation chain is used to map 3D cloud coordinate points to an image pixel that captures that 3D point. The color data from the image pixel can then be assigned to the corresponding point in the point cloud. One common method for mapping cloud points to image pixels is to project the 3D points to the image plane. This would be done using the pinhole camera model (Sturm 2014) combined with the transformation from the map frame to the camera frame. Once converted, the projected points can be assigned a color based on their location relative to the pixels on the image plane. The projection from 3D coordinates to a pixel coordinate can be done using Eq. (11):

$$s[u \quad v \quad 1]^T = K[R_{CM} \quad t_{CM}]P_M^k \qquad (11)$$

where s = scaling constant, u, v = integer pixel coordinates on the image plane, and R_{CM} and t_{CM} = respective 3×3 rotation and 3×1 translation matrices that correspond to the transformation from the map frame (F_M) to the camera frame (F_C or F_F corresponding to the vision and IR cameras, respectively). P_M^k and K (Zhang 2002) are shown expanded in Eq. (12). P_M^k is the kth homogeneous coordinate point in the point cloud, presented in the map frame F_M. K is the intrinsic camera matrix that is calculated from the calibration. K consists of the focal lengths (f_x and f_y) and the image center location (c_x and c_y) in both x and y directions. All values in K are provided in units of pixels

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f & c \\ 0 & 0 & 1 \end{bmatrix}; \quad P_M^k = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad (12)$$

A few problems arise when using this projection process for colorization. First, this method would not account for occlusions, which becomes an increasingly bigger issue as the cloud size and complexity increase. For example, both surfaces on a single bridge beam may project onto the image plane, whereas the image can only capture one of those planes since the other is occluded. Furthermore, surfaces that are normal to the image plane would also

ISSN: 2278-4632
Vol-12 Issue-07 No.03 July 2022

create problems when colorizing. Thus, an alternate approach is followed, where the points on the image plane are mapped to corresponding points on the 3D point cloud using the inverse of the pinhole camera model.

This colorization process iterates through all (i, j) pixels of all n images taken. The first step of this process is to remove distortion in the current image. This is done using Eq. (13) and the distortion parameters obtained during camera intrinsic calibration (Heikkila and Silven 1997)

$$\tilde{u} = u(1 + k_1 r^2 + k_2 r^4) + 2p_1 uv + p_2(r^2 + 2u^2)$$
$$\tilde{v} = v(1 + k_1 r^2 + k_2 r^4) + 2p_2 uv + p_1(r^2 + 2v^2)$$
$$r^2 = u^2 + v^2 \quad (13)$$

where $\tilde{u}$, $\tilde{v}$ = distorted image coordinates, k_1, k_2 = radial distortion coefficients, and p_1, p_2 = tangential distortion coefficients.

The second step is to convert the undistorted pixel coordinates on the current image plane to points in the camera coordinate frame. To do so, the inverse of the camera matrix, K, is taken and premultiplied on both sides of Eq. (14) to isolate $P_C^{i,j}$, the coordinates of the (i, j)th pixel point in the camera frame F_C for the current image. Eq. (14) differs from Eq. (11) since the points are being calculated in the camera frame, therefore R_{CM} and t_{CM} are not applied. Isolating for $P_C^{i,j}$ yields Eq. (15). Eq. (14) is iterated through each (i, j)th pixel as opposed to Eq. (11), which is iterated through each individual point in the cloud

$$s[u \quad v \quad 1]^T = KP_C^{i,j} \quad (14)$$

$$P_C^{i,j} = [x \quad y \quad z]^T = K^{-1} s[u \quad v \quad 1]^T = s\left[\frac{u - c_x}{f_x} \quad \frac{v - c_y}{f_y} \quad 1\right]^T \quad (15)$$

The scaling constant, s, is equal to the distance from the camera optical center to the image plane in this application and is obtained by multiplying the focal length in the pixels by the physical pixel size, which is obtained from the manufacturer.

Once the u, v points are represented in the camera coordinate frame, it is possible to convert the coordinates of the camera optical center—located at (0, 0, 0) in the camera frame—and the pixel 3D coordinates ($P_C^{i,j}$) to the map coordinate frame. This is done using the transformation between the camera and the robot frame, T_{CR}, and the transformation from the robot frame to the map frame, T_{MR}. A linear ray direction vector can then be calculated by subtracting the camera optical center from the pixel point on the image plane. The ray is extended in the calculated direction until this ray makes contact with (or within a specified threshold distance of) a point on the point cloud, as illustrated in Fig. 13.

For each pixel, the calculated ray is extended incrementally and a KD-tree search algorithm (Bentley 1975) is performed at each iteration to check if the endpoint of the ray is within a specified distance threshold from a point in the point cloud. This process is accelerated in two ways, first by eliminating all points in the cloud that are not located within the image plane (i.e., have u, v coordinates outside of the active area) when projected using the camera model in Eq. (11). This ensures that the search does not check points that are irrelevant to the current image. Second, the rays are incrementally extended by a distance equal to the distance between the ray endpoint and the closest point on the reduced cloud. After each ray extension, if a point is found to be within the threshold distance from the ray endpoint, then that point is assigned the color of the pixel. Otherwise, the ray is extended. This process continues until the current pixel is successfully mapped to a cloud point

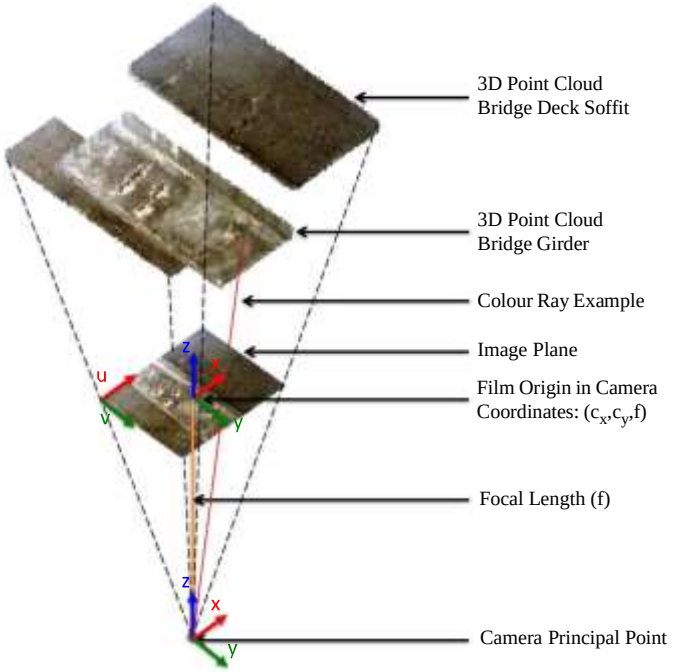


Fig. 13. Point cloud colorization illustration.

or until a set number of maximum iterations is reached. This incremental ray extension and contact check process is necessary, instead of immediately colorizing the closest point to a pixel, because the closest point may not necessarily be along the path of the ray. This algorithm thus ensures that the cloud point colored is the first point that the ray contacts, not simply the closest point to the pixel. This process is summarized in Fig. 14.

This colorization process can be done automatically with both the vision and the IR images. Fig. 15 shows a bottom view of the colored point cloud using both cameras and Fig. 16 shows an isometric view. Note that the vision images are in color, but the lack of exposure under the bridge makes the colorized point cloud primarily gray. In both sets of colorized clouds, the colorized portion depicts a concrete girder running in the approximate center of the colorized portion, with concrete deck soffit on either side.

Automated Defect Detection

Typical defects in concrete structures, such as spalls, delaminations, cracks, and exposed rebar, are important elements of bridge visual inspections. The terms cracks and exposed rebar are self-explanatory. A delamination is concrete that is internally separated, but not fully detached from the main structure. It is caused by internal forces, such as expansion due to the corrosion of the rebar. As a delamination worsens, it will eventually become a spall, which is an area where the concrete has fully separated and fallen off the structure.

Images can provide important information regarding such defects and about the general condition of reinforced concrete bridges. In addition to defects detected using images in the visual spectrum, IR images can be used to detect delaminations. The air gap created by delaminations acts as insulation between the delaminated concrete and the remaining bridge deck, thus causing variations in the heating and cooling patterns of the concrete at the location of the delamination. For example, a delaminated mass exposed to solar heating will appear warmer in an IR image, as the heat flow is

ISSN: 2278-4632
Vol-12 Issue-07 No.03 July 2022

Point Cloud Colourization Algorithm

INPUT:

n : image number
N : set of all images
$T^n_{MR,t}$: transformation at time t, from robot frame to map frame for image n
T_{RC} : transformation from camera frame to robot frame (time invariant)
i, j : pixel coordinates in the horizontal (u) and vertical (v) directions, respectively
I, J : set of horizontal and vertical pixels, respectively
p^n_C : camera optical center in camera frame coordinates for image n
$P^{i,j}_M$: pixel i,j coordinates in map frame
K : camera intrinsic matrix
s : scale coefficient equal to focal length in meters
M_f : matrix containing every point in the final 3D point cloud map
α : distance threshold for determining contact
β : maximum number of iterations per pixel

for $n \in N$ **do**
  $T^n_{MC,t} \leftarrow T^n_{MR,t} \times T_{RC}$ ▷ compute transform from $\mathcal{F}_\mathcal{C}$ to $\mathcal{F}_\mathcal{M}$ for image n
  $p^n_M \leftarrow T_{MC,t} \times p^n_C$ ▷ convert camera optical center in $\mathcal{F}_\mathcal{C}$ to $\mathcal{F}_\mathcal{M}$ for image n
  for $i \in I$ **do** ▷ iterate through all pixels
    for $j \in J$ **do**
      $P^{i,j}_M \leftarrow T_{MC,t} * K^{-1}\ s \begin{bmatrix} i & j & 1 \end{bmatrix}^T$ ▷ convert pixel point from $\mathcal{F}_\mathcal{C}$ to $\mathcal{F}_\mathcal{M}$
      $\phi_r \leftarrow P^{i,j}_M - p^n_M$ ▷ calculate ray direction vector, ϕ_r
      $P^E_M \leftarrow P^{i,j}_M$ ▷ initialize the ray endpoint, P^E_M, in the map frame
      while true **do**
        $P^C_M \leftarrow \arg\min(dist(P^E_M, M_f))$ ▷ calculate the closest point, P^C_M, to P^E_M
        $D^C \leftarrow \min(dist(P^E_M, M_f))$ ▷ calculate the distance, D^C, from P^C_M to P^E_M
        if $D^C < \alpha$ **then**
          return P^C_M, n, i, j ▷ return point and pixel
        else if $iter > \beta$ **then**
          return -1 ▷ return error: no point painted/colourized
        end if
        $P^E_M \leftarrow P^E_M + \phi_r\ D^C$ ▷ extend ray by D^C
      end while
    end for
  end for
end for

Fig. 14. Point cloud colorization algorithm pseudocode.

trapped in the delaminated concrete by the air gap (Clemena and McKeel 1978).

In addition to manual detection from images, various algorithms exist to automatically detect defects from both visible spectrum and IR images. Simply knowing there is a defect on a bridge, while useful, does not complete the picture; both the quantification and localization of such defects are important. The ray tracing algorithm for colorizing the point cloud (Fig. 14) can facilitate this quantification and localization. As a proof of concept, two semi-automated (semiautomated being a process that requires very little human intervention with some minor tuning by the inspector for best results) algorithms for defect detection in images are considered to localize the damages on a 3D point cloud generated using the UGV inspection platform. These algorithms are edge detection (Parker 2011) and thresholding (Otsu 1979).

Edge detection presents poor initial results for this data and hence thresholding with connectivity filtering is used to complete the preliminary defect detection in this study. Additionally, the thresholding operation is more applicable than edge detection for delamination detection in thermal images as well as for finding exposed rebar in the concrete. There are no significant cracks on the bridge that are not part of delaminations; as such, only delamination and exposed rebar results are detected for localization and quantification on the point cloud. Fig. 17 shows examples of thresholding used for detecting exposed rebar in the vision images and delaminations in the IR images.

Fig. 18(a) shows the areas of exposed rebar on the bridge deck soffit and the concrete girder. These points on the cloud are highlighted for better visibility. Two small areas of exposed rebar are detected and localized on the point cloud. The locations of the exposed rebar were confirmed through visual site inspection.

Fig. 18(b) shows the areas of delamination detected in the IR images. The results of transferring the delaminated areas from IR images to the point cloud show a large number of delaminations located in the middle section of the bridge soffit. Few delaminations exist on the concrete girder or on the outside section of the soffit. Not all areas of delamination are able to be confirmed precisely by human inspection, as hammer sounding is not feasible at all locations; however, these results were consistent with the delaminations that were visually located during the site inspection.

ISSN: 2278-4632
Vol-12 Issue-07 No.03 July 2022

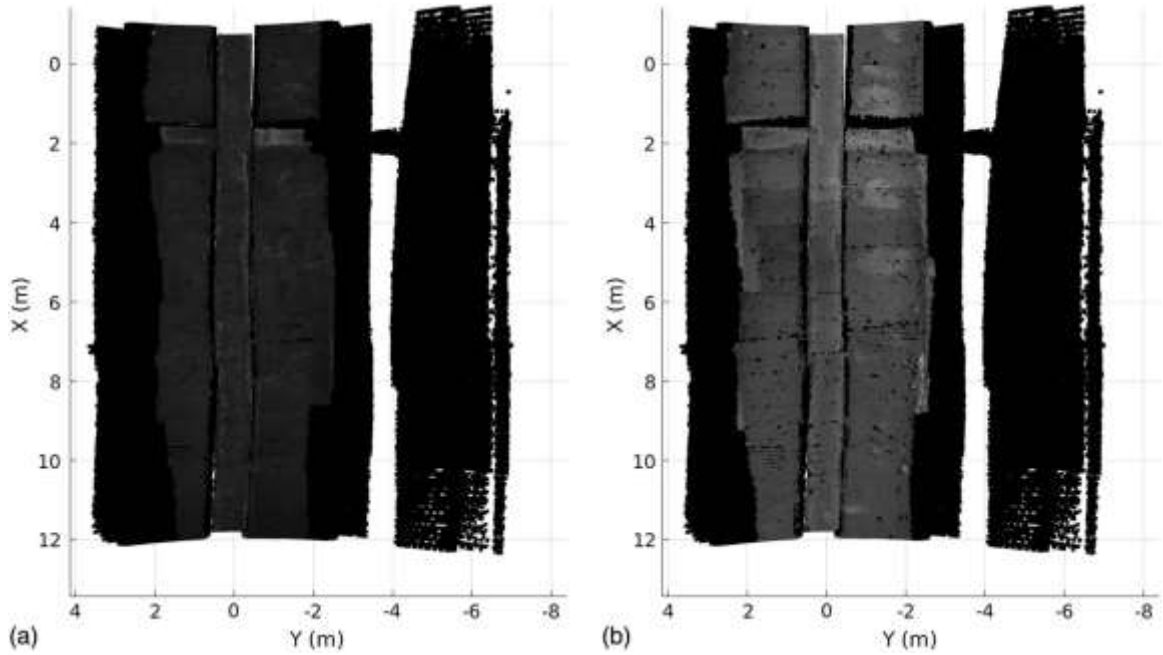


Fig. 15. 3D bridge point cloud with bottom view of deck soffit: (a) cloud colorized with vision camera; and (b) cloud colorized with IR camera.

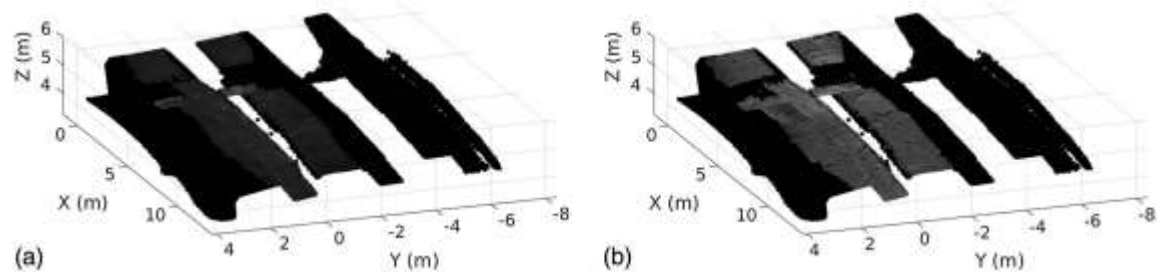


Fig. 16. 3D bridge point cloud with isometric views: (a) cloud colorized with vision camera; and (b) cloud colorized with IR camera.

The quantification of the exposed rebar or delaminations can be performed by measuring the lengths using standard tools for manipulating a 3D point cloud. Fig. 19 shows an example of estimating the size of delaminations on an enlarged section of the point cloud. The axes of the figure and bounding box dimensions are shown for size reference. The bounding boxes are used to estimate size, as this step is similar to what would be done in the field. The delaminated areas from the top of the figure to the bottom of the figure are estimated to be 0.48, 0.22, and 0.33 m^2. The accuracy of these quantifications was not tested and compared to actual onsite measurements by inspectors due to accessibility limitations. However, it can be assumed that these errors would be of similar magnitude to the errors determined in the Evaluation Metrics section because the scale given to calculate distances from the image data is determined by the lidar-generated point cloud. As a result of the fusion of the lidar and vision sensor data, the level of accuracy for defect quantification is expected to be significantly more accurate than current visual estimates from trained inspectors, especially for areas of difficult accessibility.

The results of applying semiautomated defect detection algorithms to images show that the defects detected in the visible spectrum and IR images of reinforced concrete images can be localized and quantified onto a 3D point cloud using a SLAM algorithm for the pose estimate of the robot, transformations based on sensor calibration, and image pixel data. This process exhibits success in detecting and quantifying exposed rebar and delaminations. Even though the bridges that were inspected did not have any visible cracking, it is expected that such features can also be detected and quantified using this methodology.

Conclusion and Future Work

Increasing the quality of critical infrastructure inspections remains of high importance for many municipalities burdened with aging infrastructure, specifically bridges. Infrastructure research is shifting to find new ways to improve upon current bridge inspection practice; the use of robotics and remote sensing is a natural progression in this regard. This work proposes a robotic bridge inspection framework, introducing a solution that will help increase inspection accuracy, eliminate human subjectivity, decrease inspection time, and has the potential to be implemented on aerial or ground robots to enable easy inspection of hard-to-access bridges. The platform shown herein consists of a UGV equipped with an onboard computer and several state-of-the-art robotic sensors. A procedure for continuous scanning of the underside of concrete bridges is also introduced for accurate 3D mapping and defect quantification.

ISSN: 2278-4632
Vol-12 Issue-07 No.03 July 2022

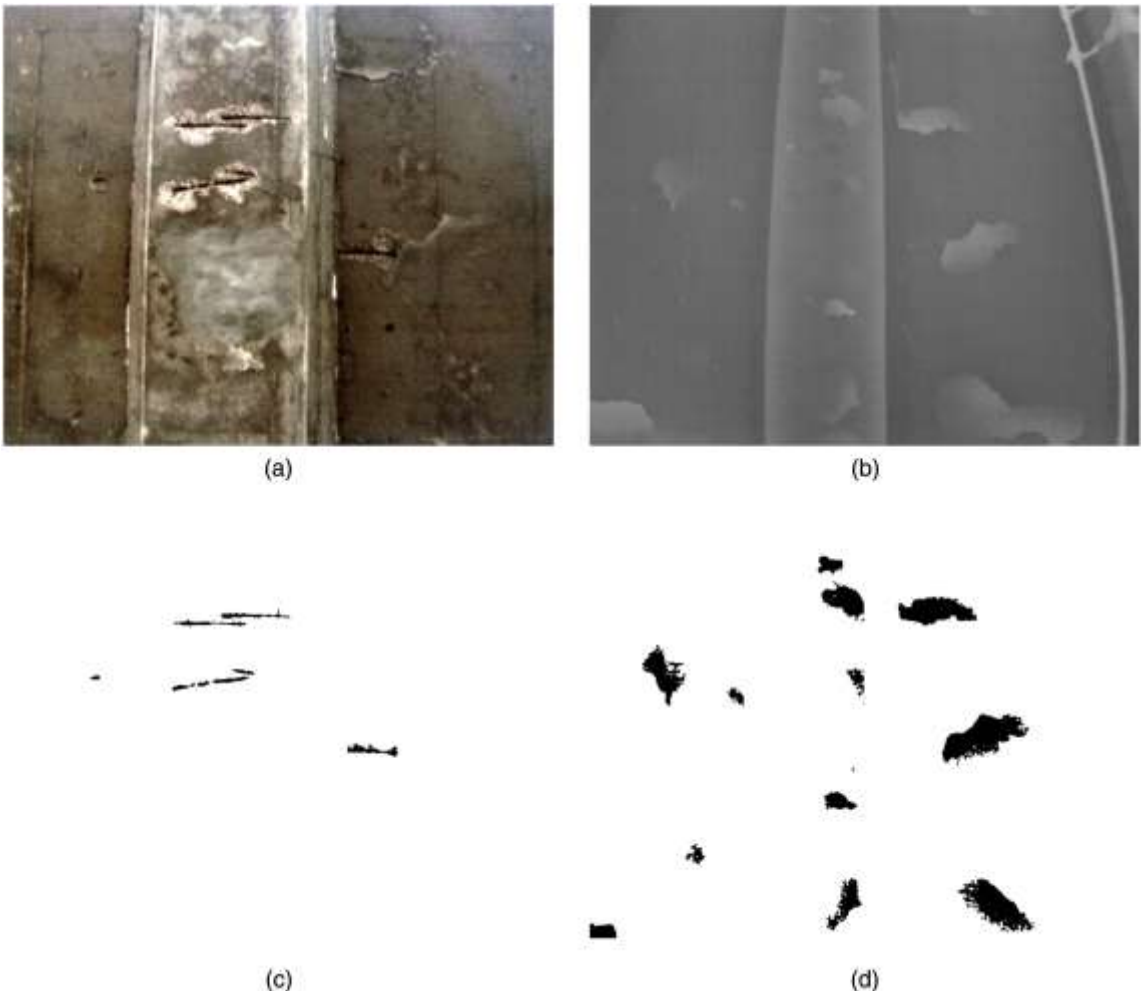


Fig. 17. Semiautomated defect detection using thresholding: (a) vision camera image; (b) IR camera image; (c) detected rebar in the vision camera image; and (d) detected delaminations in the IR image.

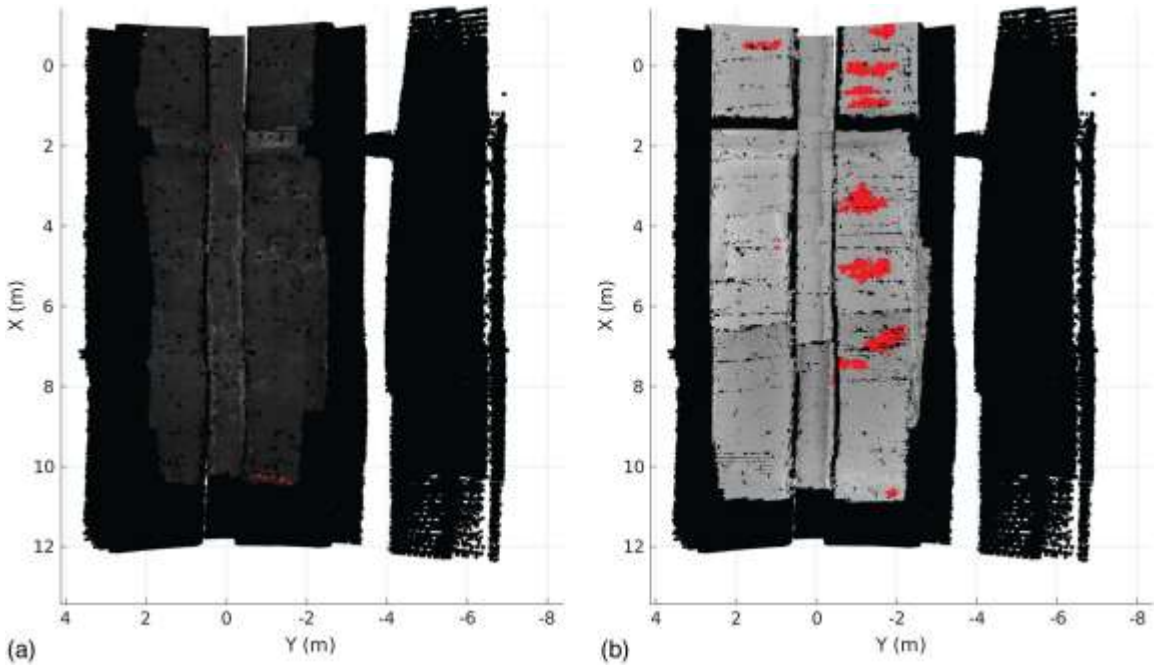


Fig. 18. 3D bridge point cloud labeled with defects: (a) points colored red to show exposed rebar; and (b) points colored red to show delaminations.

ISSN: 2278-4632
Vol-12 Issue-07 No.03 July 2022

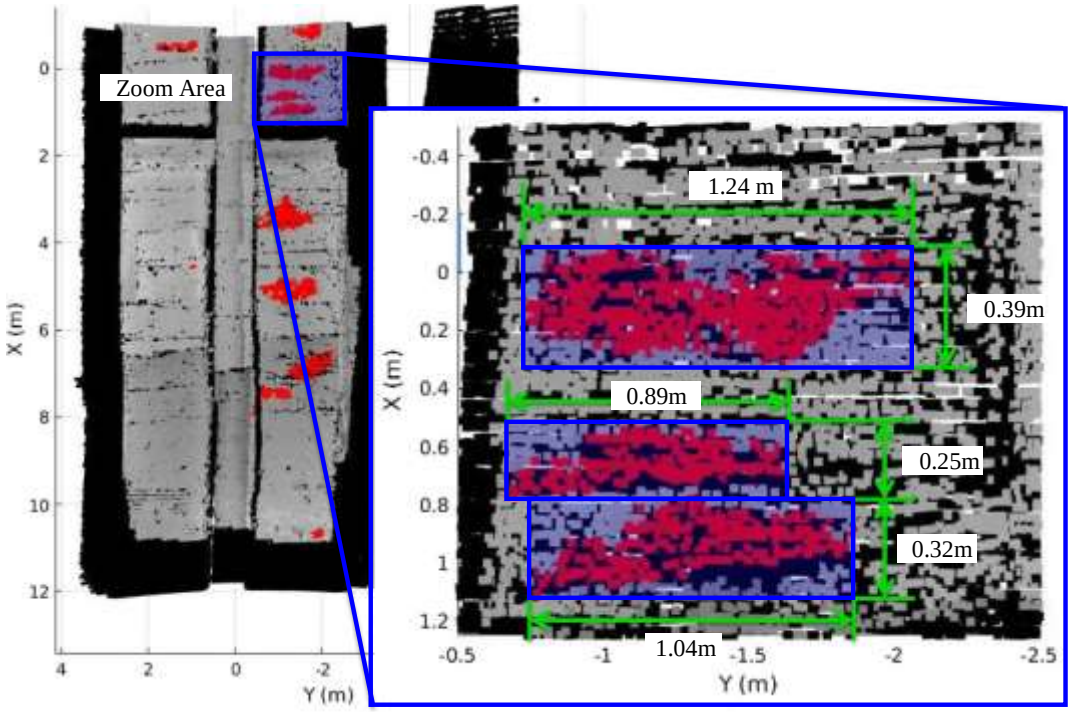


Fig. 19. Point cloud labeled to show delamination sizes.

Tests conducted on three different sites show that high-quality maps can be generated using the proposed UGV-based platform rapidly. Defect detection and quantification results performed on the scanned information show that it is possible to achieve sufficient accuracy for typical defects of interest in concrete bridge inspections. The procedure presented to colorize the point clouds demonstrates the potential to both visualize and quantify a range of defects observed from multiple sensor modalities.

The presented platform and the methodology are merely first steps in the process of the adoption of robotics in bridge inspection applications. Many improvements can be pursued within the proposed framework; the quality of mapping can be increased significantly with advanced SLAM techniques, such as a graph SLAM problem formulation, while also incorporating full loop closure and visual odometry sources. With such techniques, it may be possible to achieve measurement accuracy very close to that of the TLS. Additionally, there is ongoing work by the authors on deep learning approaches to image pixel classification and point cloud analysis for determining geometric defects (once more high-quality maps can be achieved). Future work will include identifying the level of accuracy of defect detection and extracting volumetric information from defects, while also providing a measure of the minimum defect sizes that can be detected with this procedure.

References

Abdel-Qader, I., O. Abudayyeh, and M. E. Kelly. 2003. "Analysis of edge- detection techniques for crack identification in bridges." *J. Comput. Civ. Eng.* 17 (4): 255-263. https://doi.org/10.1061/(ASCE)0887-3801(2003)17:4(255).

Adhikari, R. S., O. Moselhi, and A. Bagchi. 2014. "Image-based retrieval of concrete crack properties for bridge inspection." *Autom. Constr.* 39: 180-194. https://doi.org/10.1016/j.autcon.2013.06.011.

Barfoot, T. D. 2018. *State estimation for robotics*. Cambridge, UK: Cambridge University Press.

Bentley, J. L. 1975. "Multidimensional binary search trees used for associative searching." *Commun. ACM* 18 (9): 509-517. https://doi.org/10.1145/361002.361007.

Besl, P., and N. McKay. 1992. "A method for registration of 3-D shapes." *IEEE Trans. Pattern Anal. Mach. Intell.* 14 (2): 239-256. https://doi.org/10.1109/34.121791.

Blanco, J. 2014. *A tutorial on SE (3) transformation parameterizations and on-manifold optimization*. Technical Rep. Málaga, Spain: Univ. Malaga.

Bouguet, J.-Y. 2015. "Camera calibration toolbox for MATLAB." Accessed July 10, 2018. http://www.vision.caltech.edu/bouguetj/calib_doc/.

Cadena, C., L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. Leonard. 2016. "Past, present, and future of simultaneous localization and mapping: Towards the robust-perception age." *IEEE Trans. Rob.* 32 (6): 1309-1332. https://doi.org/10.1109/TRO.2016.2624754.

Campion, G., G. Bastin, and B. Dandrea-Novel. 1996. "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots." *IEE Trans. Rob. Autom.* 12 (1): 47-62. https://doi.org/10.1109/70.481750.

Charron, N., S. Phillips, and S. L. Waslander. 2018. "De-noising of Lidar point clouds corrupted by snowfall." In *Proc., 2018 15th Conf. on Computer and Robot Vision (CRV)*, 254-261. New York: IEEE.

Clemena, G., and W. T. McKeel. 1978. *Detection of delamination in bridge decks with infrared thermography*. Transportation Research Record No. 664. Charlottesville, VA: Virginia Transportation Research.

Durrant-Whyte, H., and T. Bailey. 2006. "Simultaneous localization and mapping: Part I." *Rob. Autom. Mag.* 13 (2): 99-110. https://doi.org/10.1109/MRA.2006.1638022.

Ellenberg, A., A. Kontsos, F. Moon, and I. Bartoli. 2016. "Bridge deck delamination identification from unmanned aerial vehicle infrared imagery." *Autom. Constr.* 72: 155-165. https://doi.org/10.1016/j.autcon.2016.08.024.

Gillins, M. N., D. T. Gillins, and C. Parrish. 2016. "Cost-effective bridge safety inspections using unmanned aircraft systems (UAS)." In *Geotechnical and Structural Engineering Congress August 2016*, 1931-1940. Reston, VA: ASCE.

Goorts, K., S. Phillips, A. Ashasi-Sorkhabi, and S. Narasimhan. 2017. "Structural control using a deployable autonomous control system." *Int. J. Intell. Rob. Appl.* 1 (3): 306-326. https://doi.org/10.1007/s41315-017-0025-7.

Gucunski, N., B. Basily, J. Kim, J. Yi, T. Duong, K. Dinh, S.-H. Kee, and A. Maher. 2017. "RABIT: Implementation, performance validation and integration with other robotic platforms for improved management of bridge decks." *Int. J. Intell. Rob. Appl.* 1 (3): 1-16.

Heikkila, J., and O. Silven. 1997. "A four-step camera calibration procedure with implicit image correction." In *Proc., IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, 1106-1112. New York: IEEE.

Jahanshahi, M. R., S. F. Masri, C. W. Padgett, and G. S. Sukhatme. 2013. "An innovative methodology for detection and quantification of cracks through incorporation of depth perception." *Mach. Vision Appl.* 24 (2): 227-241. https://doi.org/10.1007/s00138-011-0394-0.

Khaloo, A., and D. Lattanzi. 2017. "Hierarchical dense structure-from-motion reconstructions for infrastructure condition assessment." *J. Comput. Civ. Eng.* 31 (1): 04016047. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000616.

Khaloo, A., D. Lattanzi, K. Cunningham, R. Dell'Andrea, and M. Riley. 2018. "Unmanned aerial vehicle inspection of the Placer River Trail Bridge through image-based 3D modelling." *Struct. Infrastruct. Eng.* 14 (1): 124-136. https://doi.org/10.1080/15732479.2017.1330891.

Kim, P., J. Chen, and Y. K. Cho. 2018. "Autonomous mobile robot localization and mapping for unknown construction environments." In *Proc., ASCE Construction Research Congress (CRC) 2018*, 147-156. Reston, VA: ASCE.

Kim, P., B. Jingdao Chen, and B. K. Yong Cho. 2017. "Robotic sensing and object recognition from thermal-mapped point clouds." *Int. J. Intell. Rob. Appl.* 1 (3): 243-254. https://doi.org/10.1007/s41315-017-0023-9.

Koch, C., K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth. 2015. "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure." *Adv. Eng. Inf.* 29 (2): 196-210. https://doi.org/10.1016/j.aei.2015.01.008.

La, H. M., R. S. Lim, B. B. Basily, N. Gucunski, J. G. Yi, A. Maher, F. A. Romero, and H. Parvardeh. 2013. "Mechatronic systems design for an autonomous robotic system for high-efficiency bridge deck inspection and evaluation." *IEEE-ASME Trans. Mechatron.* 18 (6): 1655-1664. https://doi.org/10.1109/TMECH.2013.2279751.

Laefer, D. F., L. Truong-Hong, H. Carr, and M. Singh. 2014. "Crack detection limits in unit based masonry with terrestrial laser scanning." *NDT & E Int.* 62: 66-76. https://doi.org/10.1016/j.ndteint.2013.11.001.

Law, D. W., D. Silcock, and L. Holden. 2018. "Terrestrial laser scanner assessment of deteriorating concrete structures." *Struct. Control Health Monit.* 25 (5): e2156. https://doi.org/10.1002/stc.2156.

Lehtola, V. V., et al. 2017. "Comparison of the selected state-of-the-art 3D indoor scanning and point cloud generation methods." *Remote Sens.* 9 (8): 796. https://doi.org/10.3390/rs9080796.

Liu, W., S. Chen, and E. Hauser. 2011. "LiDAR-based bridge structure defect detection." *Exp. Tech.* 35 (6): 27-34. https://doi.org/10.1111/j.1747-1567.2010.00644.x.

Liu, Y.-F., S. Cho, and J.-S. Fan. 2016. "Concrete crack assessment using digital image processing and 3D scene reconstruction." *J. Comput. Civ. Eng.* 30 (1): 04014124. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000446.

Ministry of Transportation Ontario. 2008. *Ontario structural inspection manual (OSIM)*. St. Catharines, ON, Canada: Publications Ontario.

Moller, P. 2008. *CALTRANS bridge inspection aerial robot*. Rep. No. CA08-0182. Davis, CA: Univ. of California at Davis.

Moore, M., B. Phares, B. Graybeal, D. Rolander, and G. Washer. 2001. *Reliability of visual inspection for highway bridges*. Rep. No. FHWA-RD-01-020. McLean, VA: Federal Highway Administration.

Moore, T., and D. Stouch. 2014. "A generalized extended kalman filter implementation for the robot operating system." In *Proc., 13th Int. Conf. on Intelligent Autonomous Systems (IAS-13)*. New York: Springer.

Omar, T., M. L. Nehdi, and T. Zayed. 2018. "Infrared thermography model for automated detection of delamination in RC bridge decks." *Constr. Build. Mater.* 168: 313-327. https://doi.org/10.1016/j.conbuildmat.2018.02.126.

Otsu, N. 1979. "A threshold selection method from gray-level histograms." *IEE Trans. Syst. Man Cybern.* 9 (1): 62-66. https://doi.org/10.1109/TSMC.1979.4310076.

Parker, J. R. 2011. *Algorithms for image processing and computer vision*. Indianapolis: Wiley.

Prasanna, P., K. J. Dana, N. Gucunski, B. B. Basily, H. M. La, R. S. Lim, and H. Parvardeh. 2016. "Automated crack detection on concrete bridges." *IEEE Trans. Autom. Sci. Eng.* 13 (2): 591-599. https://doi.org/10.1109/TASE.2014.2354314.

Rusu, R. B., and S. Cousins. 2011. "3D is here: Point cloud library (PCL)." In *Proc., IEEE Int. Conf. on Robotics and Automation*, 1-4. New York: IEEE.

Siegwart, R., and I. R. Nourbakhsh. 2004. *Introduction to autonomous mobile robots*. Cambridge, MA: MIT Press, Massachusetts Institute of Technology.

Sturm, P. 2014. "Pinhole camera model." In *Computer vision*. Boston: Springer.

Thrun, S., Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte. 2004. "Simultaneous localization and mapping with sparse extended information filters." *Int. J. Rob. Res.* 23 (7-8): 693-716. https://doi.org/10.1177/0278364904045479.

Torok, M. M., M. Golparvar-Fard, and K. B. Kochersberger. 2014. "Image-based automated 3D crack detection for post-disaster building assessment." *J. Comput. Civ. Eng.* 28 (5): A4014004. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000334.

Turkan, Y., S. Laflamme, and L. Tan. 2016. *Terrestrial laser scanning-based bridge structural condition assessment*. In Trans Project Rep. 199. Ames, IA: Iowa State Univ.

Valença, J., I. Puente, E. Júlio, H. González-Jorge, and P. Arias-Sánchez. 2017. "Assessment of cracks on concrete bridges using image processing supported by laser scanning survey." *Constr. Build. Mater.* 146: 668-678. https://doi.org/10.1016/j.conbuildmat.2017.04.096.

Yeum, C. M., and S. J. Dyke. 2015. "Vision-based automated crack detection for bridge inspection." *Comput.-Aided Civ. Infrastruct. Eng.* 30 (10): 759-770. https://doi.org/10.1111/mice.12141.

Zhang, Z. 2002. "A flexible new technique for camera calibration." *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (11): 1330-1334. https://doi.org/10.1109/34.888718.

Zink, J., and B. Lovelace. 2015. *Unmanned aerial vehicle bridge inspection demonstration project*. No. MN/RC 2015-40. Washington, DC: Transportation Research Board.