Bridge Inspection with Aerial Robots: Automating the Entire Pipeline of Visual Data Capture, 3D Mapping, Defect Detection, Analysis, and Reporting

Jacob J. Lin, A.M.ASCE[1]; Amir Ibrahim, S.M.ASCE[2];
Shubham Sarwade, S.M.ASCE[3]; and Mani Golparvar-Fard, Ph.D., A.M.ASCE[4]

Abstract: The aging of bridges coupled with increased vehicular traffic requires timely and accurate inspections for elevated highway structures. Recent studies have leveraged the advent of drones and computer vision to automatically conduct quick, safe, and effective inspections for elevated highway structures. However, such studies rarely offer insight or recommendations for an end-to-end integrated system that streamlines data collection, analytics, and reporting. Toward this goal, we present an end-to-end robotic bridge inspection system consisting of five tightly coupled methods to: (1) create automatic data collection missions; (2) assure visual quality of such missions; (3) reconstruct three-dimensional (3D) models of elevated structures; (4) detect and localize surface distresses in 3D; and (5) generate reports complying with highway agencies' requirements. We validate each developed method and the whole system on two representative inspection projects. Results show that our system can objectively satisfy requirements for data collection and provide up to 85.3% average precision over five defect types. We finally share lessons learned while deploying our system to 30 bridge inspection projects in the US and Japan, particularly for documenting, communicating, and following-up with bridge inspectors' recommendations. **DOI:** 10.1061/(ASCE)CP.1943-5487.0000954.

Introduction

The recent collapse of a bridge in Tennessee (WTKR 2019) and the partial collapse of a railroad bridge in Chicago (CBS Chicago 2019) are yet again raising new alarms about the delicate state of infrastructures. Many years after the collapse of the I-35W bridge in Minneapolis, these incidents reminded Americans and the world yet again that aging bridges and elevated highway structures are still at high risk to the safety and security of transportation systems. A 2018 report demonstrates that 8.86% of the 612,408 bridges in the United States are still labeled as either structurally deficient or functionally obsolete (Reason 2020). These bridges are not unsafe; however, a deficient bridge needs immediate attention, and an obsolete one does not meet current design standards.

Inspecting a large number of bridges with limited resources is challenging for transportation agencies and bridge owners given that such inspections are expensive and highly disruptive to traffic (Wells and Lovelace 2018). Nevertheless, extra costs and delays are required for planning lane closures and safe rerouting. Another challenge is onsite documentation, where current practices are time-consuming, result in assessments that are inaccurate and inconsistent, and often require inspectors to re-visit the inspection site for additional data collection. Recent recommendations by the National Cooperative Highway Research Program suggest that local authorities establish expert panels to assess the durability, reliability, and safety of the bridges in local areas to determine which bridges should be inspected more often and the types of inspections they should receive (Washer et al. 2014). These recommendations yet again reinforce the need for quicker, safer, and more effective approaches to inspection.

The advent of affordable aerial platforms with the agility and dexterity to carry digital cameras in close proximity to the underside of elevated highway structures, coupled with a greater overall awareness of the technology, presents an opportunity to conduct quick, safe, and effective bridge inspections. For instance, an aerial platform used for bridge inspection reduces the need for expensive access and traffic control measures compared to using inspection vehicles, man lifts, rope access, and ladders. Nevertheless, utilizing cloud-based information systems and visual analytics significantly reduce the need for site visits, which can help cut inspection costs. The MDOT (2019) has shown 60% cost savings associated with drone-based inspections; moreover, a report from the American Association of State Highway and Transportation Officials (AASHTO 2018) has proclaimed that 35 of 44 reporting state DOTs with previous experience are deploying aerial platforms in some capacity.

Building on pilot projects with unmanned aerial vehicles (UAVs), state DOTs have initiated programs to enhance condition assessment by establishing standard procedures for documenting, communicating, and following up on bridge inspectors' recommendation; offering unified systems for managing bridge maintenance (store and retrieve data remotely); improving inspection documentation,

[1] Assistant Professor, Dept. of Civil Engineering, National Taiwan Univ., Room 808, Civil Engineering Research Bldg. No. 188, Sec. 3, Sinhai Rd., Taipei 10688, Taiwan (corresponding author). Email: jacoblin@ntu.edu.tw

[2] Ph.D. Candidate, Dept. of Civil and Environmental Engineering, Univ. of Illinois at Urbana-Champaign, 205 N. Mathews Ave., Urbana, IL 61801. Email: aaelsay2@illinois.edu

[3] Graduate Research Assistant, Dept. of Civil and Environmental Engineering, Univ. of Illinois at Urbana-Champaign, 205 N. Mathews Ave., Urbana, IL 61801. Email: sarwade2@illinois.edu

[4] Associate Professor of Civil and Environmental Engineering, Computer Science, and Technology Entrepreneurship, Univ. of Illinois at Urbana-Champaign, 205 N. Mathews Ave., Urbana, IL 61801. Email: mgolpar@illinois.edu

Note. This manuscript was submitted on May 27, 2020; approved on September 29, 2020; published online on December 7, 2020. Discussion period open until May 7, 2021; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Computing in Civil Engineering***, © ASCE, ISSN 0887-3801.

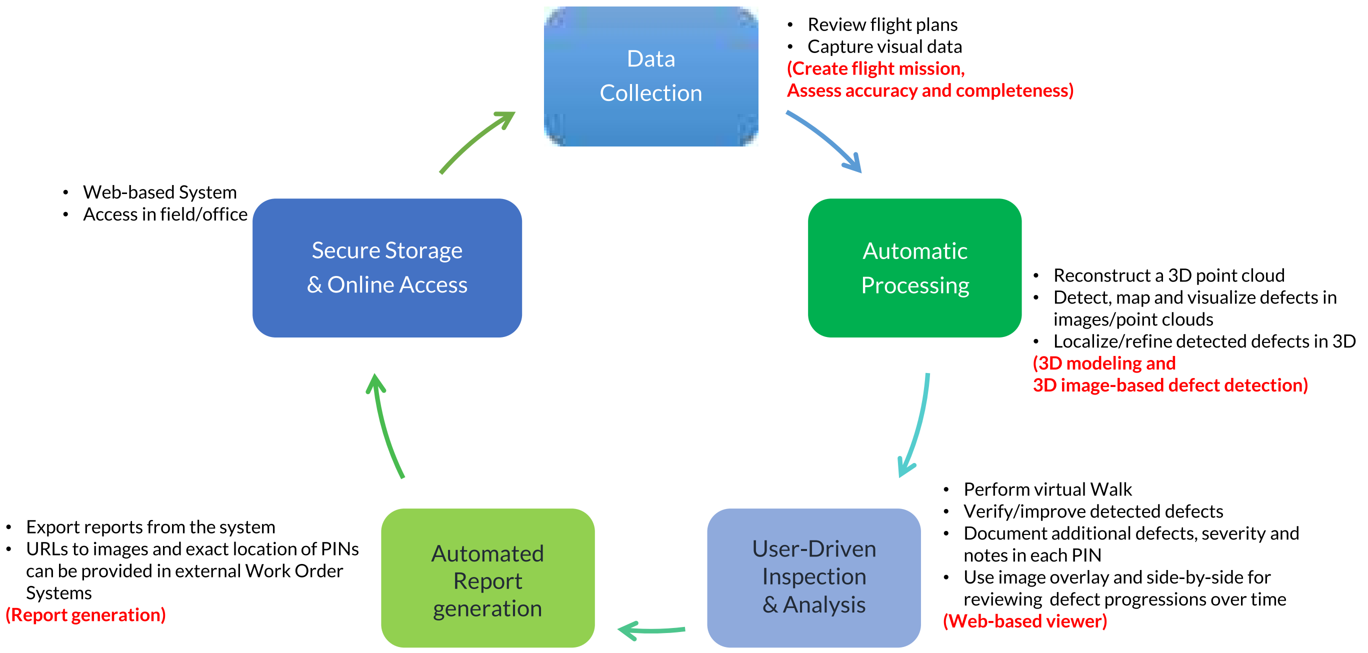

Data Collection
• Review flight plans
• Capture visual data
(Create flight mission, Assess accuracy and completeness)
Automatic Processing
• Reconstruct a 3D point cloud
• Detect, map and visualize defects in images/point clouds
• Localize/refine detected defects in 3D
(3D modeling and 3D image-based defect detection)
User-Driven Inspection & Analysis
• Perform virtual Walk
• Verify/improve detected defects
• Document additional defects, severity and notes in each PIN
• Use image overlay and side-by-side for reviewing defect progressions over time
(Web-based viewer)
Automated Report generation
• Export reports from the system
• URLs to images and exact location of PINs can be provided in external Work Order Systems
(Report generation)
Secure Storage & Online Access
• Web-based System
• Access in field/office

Fig. 1. General workflow: (a) data collection and evaluation; (b) automatic processing and 3D modeling; (c) automatic inspection and analysis; (d) report generation and sharing; and (e) secure storage and online access.

timelines, and accuracy; and providing inspectors with resources necessary for inspection at their fingertips (AASHTO 2018). Currently, bridge inspection is a manual process conducted by inspectors who first need to prepare onsite examination documents and plan for necessary traffic control measures. The inspectors then visit the site to collect and document inspection feedback through visual and physical inspection means. Finally, after the onsite inspection is concluded, the inspectors need to return back to the office to analyze the data and compile inspection reports. Although the initial manual inspection process is expected to be improved through UAV utilization, there is still no streamlined system that effectively collects, organizes, and analyzes the data in order to create practical informative reports. As a result, a large body of industrial and academic work has proposed the use of computer vision and visualization techniques to process, analyze, and share inspected drone data (Seo et al. 2018; Spencer et al. 2019). However, the main challenge for deploying such technologies for automated inspection using visual data collected by UAVs is that retrieved inspection information is not actionable without proper collection process, data organization, and analysis.

Because later studies usually focus on a single component of the entire inspection process, they rarely offer insight or recommendation on how their developed techniques can be integrated into a system that streamlines data collection, analytics, and end-to-end reporting. Computer vision methods such as three-dimensional (3D) reconstruction, image classification, object detection, and semantic segmentation have been widely implemented to improve the inspection process. While these methods demonstrate promising results, their adoptability and adaptability as part of an end-to-end system have never been examined. For example, several 3D reconstruction methods have been recommended, although there is no study that shows how accuracy and completeness of the capture can be assessed prior to flying the UAVs. Assessing the severity of damage is also highly correlated with its geometry and location. Hence, localization of the damages in images beyond a typical classification, mapping the detected defects in 3D, and characterizing surface geometry are as important as classifying the detected surface defect; however, not much is said in the literature to achieve them (Morgenthal et al. 2019; Sacks et al. 2018). The resulting information also needs to be demonstrated in a way that allows easy measurements, issue documentation, and side-by-side or overlay comparison of the defects over time. Finally, schematics such as elevation or under the bridge deck views are needed to generate reports that are in compliance with the bridge owner requirements.

To address these gaps in knowledge per method and as an end-to-end system, this paper presents a series of methods and their integration in form of an end-end system for robotic bridge inspection. Specifically, the system consists of five tightly coupled methods for: (1) creating flight missions to collect inspection data; (2) evaluating flight plans according to inspection requirements and expected visual quality of the data; (3) generating image-based 3D reconstruction of the elevated highway structures; (4) automatically detecting and localizing of surface distresses in 3D; and (5) rendering inspection reports in compliance with the requirements of highway agencies (Fig. 1). In the following, background research is presented followed by an extensive discussion on the developed methods. Next, we present results from validating each underlying algorithm and the system as a whole on two representative bridges. We also share best practices from deploying this system on 30 bridge inspection projects in the United States and Japan, focusing on procedures for documenting, communicating, and following up on bridge inspectors' recommendations. The lesson learned from this particular procedure has provided the basis for designing the framework and creating the steps for transformation into a fully robotic approach gradually.

Related Work

A substantial amount of work is published on the use of aerial robots for infrastructure inspection (Ham et al. 2016; Seo et al. 2018; Spencer et al. 2019; Agnisarman et al. 2019). The readers are encouraged to consult these articles for comprehensive reviews of the literature over the past decade. To offer a new perspective, we only discuss the gaps in knowledge associated with the most recent methods and their relevance to bridge robotic inspection.

Model-Driven Data Capture Using UAVs

Visual data collection using aerial platforms have been investigated for various construction-related applications including earthwork surveying (Siebert and Teizer 2014), construction inspection and monitoring (Ham et al. 2016; Bang et al. 2017; Freimuth and König 2018), construction safety (Irizarry et al. 2012; Chen et al. 2019), and construction progress and performance monitoring (Yang et al. 2015; Lin et al. 2015; Ham et al. 2016). Planning for visual data collection can be categorized into nonmodel based and model-driven approaches (Almadhoun et al. 2019). Generally, nonmodel based approaches are based on lawn mowing data collection patterns to assure a minimum overlap between the collected images (Nedjati et al. 2016; Baik and Valenzuela 2019; Majeed and Lee 2019), while model-driven approaches are based on advanced path planning algorithms for improved 3D visual coverage of structures (Bircher et al. 2015; Ibrahim et al. 2017a; Lindner et al. 2019). While previous research studies focus mainly on improving the data collection plan for complete visual coverage, they do not investigate planning for data collection that satisfies the requirements of bridges' visual inspection and increases its accuracy.

To this end, the application of UAVs for autonomous data collection is limited due to safety concerns (Tuttas et al. 2016) where aerial missions, especially those in close proximity to vertical structures, are highly susceptible to risks of collision. Adding to that, aviation regulations such as those enforced by the Federal Aviation Agency (FAA) in the United States have restricted flying drones in proximity to people and require drone operators to maintain a visual line-of-sight during the entire operation. Aiming to improve aerial data collection safety, model-driven flight planning has been investigated (Bircher et al. 2015; Freimuth and König 2018; Majeed and Lee 2019), where waypoints are sampled at a safe offset from the modeled structure to account for GPS errors during navigation. Moreover, to account for expected changes in the structure and existing geometry of the construction site, Ibrahim et al. (2017b) proposed the use of four-dimensional (4D) building information models (BIMs) for planning data-collection missions. Because all previous work focuses only on safe proximity of the flight to the structure, we extend such research to focus on the line-of-sight requirement during UAV mission planning.

Image-Based 3D Reconstruction

Image-based 3D reconstruction with structure from motion (SfM) and simultaneous localization and mapping (SLAM) algorithms have been widely used for the 3D modeling of infrastructure and documenting their existing conditions. Coupled with dense reconstruction algorithms, these techniques generate high-fidelity 3D point cloud and mesh models from images or videos collected manually or automatically using aerial platforms. Nevertheless, such technologies allow for better organization of the collected data by identifying the locations and viewpoints of images relative to the 3D models. Despite the maturity of such algorithms over the past decade, under many practical implementations, these methods still result in incomplete and inaccurate models (Wu 2014; Lin et al. 2015). To conduct successful 3D reconstructions, recent efforts have formulated data collection path using simulators (Degol et al. 2018) or have used empirical metrics such as ground sampling distance (GSD) based on image overlap, spatial resolution of images, perpendicularity of camera viewpoints (Paine and Kiser 2012; Byrne et al. 2017), or working distance (Chen et al. 2019). While these metrics are algorithmically validated, the context has been limited to remote sensing and mapping applications where aerial robots capture images at top-down viewpoints and at a fixed height from the ground. As such, these metrics do not account for variations in a structure's height or camera viewpoint with respect to the structure, and their GSDs are at best a guesstimate. In addition to 3D reconstruction's challenges, bridge inspector surface mapping techniques and tools are required to generate inspection orthophotos and depict elevation or under-the-deck views as a replacement for their conventional schematic designs used in bridge inspection reports. However, current 3D reconstruction pipelines and tools do not produce such inspection orthoviews.

Metrics for Visual Quality Assurance

The accuracy of automatic bridges' defect detection from images is mainly dependent on the quality of the 3D reconstruction, the coverage of the collected data to the inspection region, and the resolution of the bridge elements in the collected frames. Previous work has investigated different solutions to this problem; for instance, model-based 3D flight planning was developed by Wu et al. (2014), which formulates 3D viewpoints to generate a comprehensive flight plan, but their method requires a large number of images without optimization. Similarly, Freimuth and König (2018) presented a 3D flight planning method; however, their later work does not consider the quality requirements for 3D reconstruction. Others such as Chen et al. (2019) focused on evaluating the 3D reconstruction through projecting 3D interest areas into an orthographic view for analyzing point distribution, and Morgenthal et al. (2019) presented an object-aligned automatic 3D flight path planning for bridge inspection that aims for full coverage, image overlap, and spatial resolution.

While previous efforts investigated several visual quality attributes, there is still a need for additional parameters. For example, the proximity of the camera to the target bridge elements from canonical view points is considered a main requirement for automatic image-based defect detection (Adhikari et al. 2014). Furthermore, Lin et al. (2015) and Ham et al. (2016) showed that canonical orientations of the collected images and close proximity to the target structure are important for accurate and precise appearance-based classification of a structure's components. Other metrics are also required for complete and accurate 3D reconstruction for defect detection and localization. Specifically, Ibrahim and Golparvar-Fard (2019) show that the expected quality of the 3D reconstruction can be assessed through the visibility and redundant visibility of the target structure's elements across the collected data frames. They recommend the data collection plan to be evaluated before execution to improve the quality of the 3D reconstruction in terms of accuracy and density of the reconstructed point clouds. In addition to these metrics, Liu et al. (2019) proposed using GSD as a metric to improve feature detection in the collected images. In this paper we build upon all the previously studied visual quality assurance metrics and investigate new metrics focusing on the requirements of bridge inspection and defect detection. Mainly, we consider preflight visual quality assurance while incorporating new metrics such as perpendicularity of image viewpoints against structure and achieving a specific GSD per inspection target.

Defect Recognition, Severity Assessment, and Mapping

A large body of research has focused on computer vision methods to automatically detect and classify surface defects such as crack, spalling, exposed rebar, and efflorescence. These techniques focus on distinctive feature extraction such as edge detection (Abdel-Qader et al. 2003), morphological operators (Jahanshahi et al. 2009), and image filter banks (Nishikawa et al. 2012) for detection and classification. However, they rely on manual workflows, handcrafted features, and fine-tuned parameters to achieve similar results outside their controlled settings. Deep learning has shown promising results that could address these issues by learning model parameters from annotated training datasets. The literature on this topic can be categorized into methods for defect classification, detection, and localization where the difficulty of computer vision methods grow correspondingly. On surface defects, classification methods identify the defect type in the input image; object localization techniques classify single defect type and localize them in an image; and object detection techniques classify multiple defect types and localize them in an input image.

Deep convolutional neural network (CNN) architecture such as VGG, AlexNet, ResNet, and MetaQNN have all been heavily used for classifying single and multiclass defects (Dorafshan et al. 2018; Liu et al. 2019; Mundt et al. 2019; Xu et al. 2019). These methods have yielded exceptional results (e.g., VGG architecture achieved 70.61% for multiclass defect classification), yet the input images are often already cropped to demonstrate the defect region. Not being able to locate the defect in a full-sized two-dimensional (2D) image and map it in 3D environment weakens the direct usage of these methods in an end-to-end bridge inspection solution. Cha et al. (2018) has utilized Regional-CNN (R-CNN)–based architecture to detect defects; however, the input images for the model are also cropped around the defect. Although related works like Morgenthal et al. (2019) have investigated an end-to-end framework for bridge inspection, to the best of our knowledge, no previous work has considered detecting and predicting multiple classes of bridge defects per image and localizing them in 3D. While classifying defect types is important for preparing maintenance plans, knowing where these defects are is vital for assessing the seriousness of defects and prioritizing maintenance operations. For instance, exposed rebar at the concrete girders and drainage at bearings exhibit greater potential structural problems than spalling at the concrete deck. Understanding the geometry of a defect and calculating its width and size from the detected area is also key towards detecting the severity of such defects (Torok et al. 2014); however, research focusing on these problems is very limited.

With the recent development of modern deep CNN, object detectors such as R-CNN (Girshick et al. 2014) have reported significant accuracy improvement compared to traditional classification and detection. R-CNN adopted the regional proposal strategy to normalize the scale of each proposal before being fed to the CNN for classification. Fast-RCNN and faster-RCNN (Ren et al. 2017) have also advanced the single-scale feature computation strategy for better training efficiency, and their integration with feature pyramid networks (FPN) (Lin et al. 2016) has demonstrated multiscale feature extraction strategy in the network architecture with minimal extra cost. In the method section, we build on these techniques for damage detection.

Virtual Inspection and Reporting Generation

While researchers (Morgenthal et al. 2019; Khaloo et al. 2018) have generated high-quality 3D point clouds of bridges and others have detected multiple types of defects, some practical but critical links are missing to complete the bridge inspection process. First, an interactive interface that allows measurement and annotation on images that are overlaid on 3D point clouds is needed. As demonstrated and validated by Hoiem (2018), image-based measurements provide better accuracy for measuring geometry in point cloud. The measurement is a crucial factor for defect severity assessment, such as cracks. Second, the comparison between point clouds and images overtime is necessary to assess and track the bridge condition over time. Third, interactive orthoview generation is needed to reduce the full-time employee (FTE) time exerted in creating schematic diagrams of bridge elevation or under deck views. Fourth, preservation of the link between images and 3D point clouds. Lin and Golparvar-Fard (2018) showed example virtual walkthrough approaches for construction project controls, although these methods have not been extended to inspection tasks. These steps can lower the time spent on creating reports (Fig. 2) and enable the process to remain in compliance with the state DOT and federal agency requirements (Ryan et al. 2012).

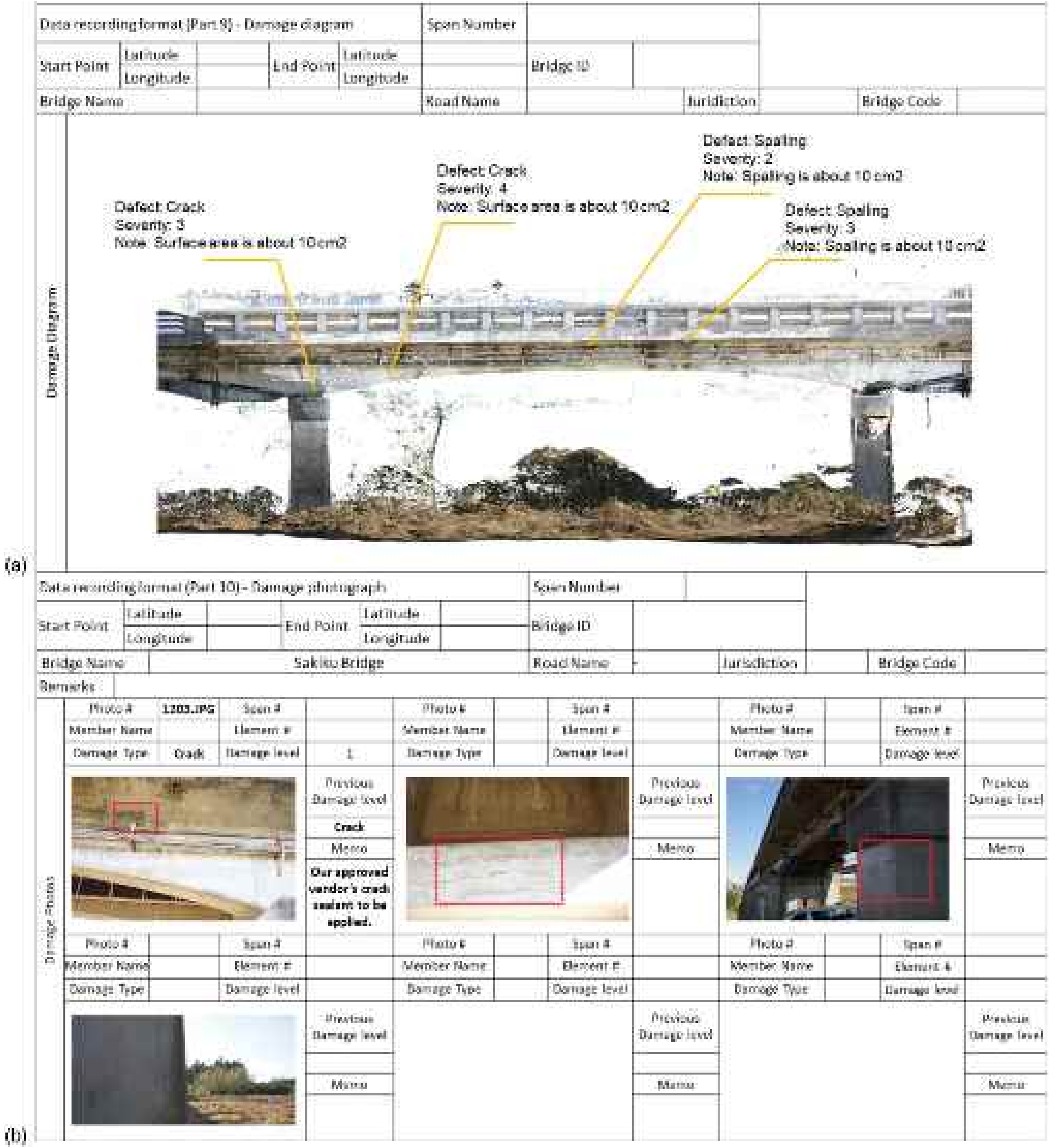
Sacks et al. (2018), Morgenthal et al. (2019), and Isailović et al. (2020) have recently demonstrated methods to localize defect information in 3D and generate reports. These methods streamline the inspection process and visualize defects in the context of a 3D bridge building information model (BIM). Creating bridge BIM models is an excellent addition; however, creating a BIM from point cloud is not fully automated, and hence the application of bridge BIM is still limited in the practice. In conclusion, the current state of research and practice shows the need to have a new system that provides: (1) automatic generation of visual data collection plans assuring the visual quality of the collected data and the safety of the collection process; (2) complete and accurate 3D reconstruction for bridge structures that satisfies the GSD requirements for bridge inspection; (3) automatic generation of inspection orthophotos while providing user-friendly environment and tools for bridge inspector surface mapping; (4) quantitative and visual feedback to assess the quality of the data collection plan while providing tools for final refinements before execution; (5) automatic defects detection and classification model that maps the defects to the 3D reconstructed bridge for improved defects localization and severity assessment; (5) a 3D environment to visualize the reconstructed 3D point cloud while showing the locations, dimensions and categories of the defects; and finally (6) automatic generation of bridge inspection reports that utilized the defects information in 3D. In the following section, we present our developed system discussing each underlying algorithm and the system as a whole.

Method: A Virtual and Visual Bridge Inspection System

System Overview

Our system takes multiple inputs to generate data collection flight missions and evaluate the data validity for virtual bridge inspection and automated damage detection. These inputs include the 3D map, geofenced inspection areas, target capture resolution, cameras relative orientation, required visibility, completeness, and safety parameters. The system outputs the flight plan and the images after the capture. These images are inputted into the 3D reconstruction pipeline to generate 3D point clouds, localized images in 3D, and camera viewpoints against the structure. This 3D information with images are fed to the defect detection model to locate the damages and map them to their respective 3D locations attributed by the defect type and defect dimensions.

Within the system, the inspector interactively documents the damages through measurements, adding pins with manually defined defects and additional notes or can reuse or correct aurtomatically detected defects. The system also generates inspection 2D orthoviews with the user-inputted desired areas (elevation and underdeck



(a)

Data recording format (Part 9) - Damage diagram					Span Number			
Start Point	Latitude		End Point	Latitude		Bridge ID		
	Longitude			Longitude				
Bridge Name				Road Name		Juridiction		Bridge Code

Damage Diagram
Defect: Crack
Severity: 3
Note: Surface area is about 10cm2
Defect: Crack
Severity: 4
Note: Surface area is about 10cm2
Defect: Spaling
Severity: 2
Note: Spaling is about 10 cm2
Defect: Spaling
Severity: 3
Note: Spaling is about 10 cm2

(b)

Data recording format (Part 10) - Damage photograph					Span Number			
Start Point	Latitude		End Point	Latitude		Bridge ID		
	Longitude			Longitude				
Bridge Name	Sakito Bridge			Road Name	-	Jurisdiction		Bridge Code
Remarks								

Photo #	1203.JPG	Span #		Photo #		Span #		Photo #		Span #	
Member Name		Element #		Member Name		Element #		Member Name		Element #	
Damage Type	Crack	Damage level	1	Damage Type		Damage level		Damage Type		Damage level	
			Previous Damage level				Previous Damage level				Previous Damage level
			Crack								
			Memo				Memo				Memo
			Our approved vendor's crack sealant to be applied.								
Photo #		Span #		Photo #		Span #		Photo #		Span #	
Member Name		Element #		Member Name		Element #		Member Name		Element #	
Damage Type		Damage level		Damage Type		Damage level		Damage Type		Damage level	
			Previous Damage level				Previous Damage level				Previous Damage level
			Memo				Memo				Memo

Fig. 2. (a) Inspection form that requires bridge elevation view to illustrate the defect locations; and (b) inspection form that requires detail description of each defect with additional notes. (Images by authors.)

views). The final step compiles all the information and generates inspection forms in compliance with the requirements. The overall workflow is illustrated in Fig. 3.

Model-Driven Visual Data Collection

The fist step in the end-to-end bridge inspection system is automatic visual data collection. Accordingly, we developed a web-based virtual environment with client-server architecture for fast and easy access to aerial data collection missions.

In this research, we present a novel approach that accounts for the practical and safety aspects of data collection at the same time. The system includes user-friendly flight planning tools that help users quickly create and alter data collection plans with visual feedback of the quality and safety assessment. Previous methods for automatic UAV mission planning (as discussed in the related-works

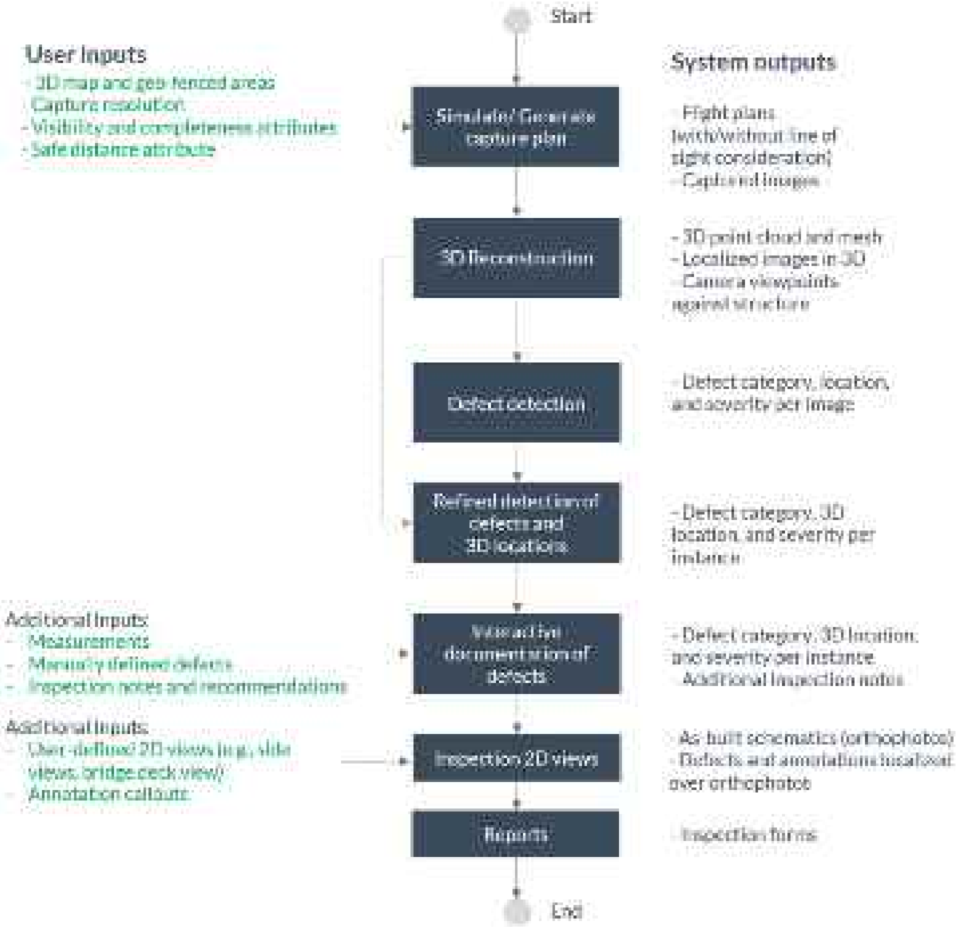

Start
User inputs
- 3D map and geo-fenced areas
- Capture resolution
- Visibility and completeness attributes
- Safe distance attribute
System outputs
Simulate/ Generate capture plan
- Flight plans (with/without line of sight consideration)
- Captured images
3D Reconstruction
- 3D point cloud and mesh
- Localized images in 3D
- Camera viewpoints against structure
Defect detection
- Defect category, location, and severity per image
Refined detection of defects and 3D locations
- Defect category, 3D location, and severity per instance
Additional inputs:
- Measurements
- Manually defined defects
- Inspection notes and recommendations
Interactive documentation of defects
- Defect category, 3D location, and severity per instance
- Additional inspection notes
Additional inputs:
- User-defined 2D views (e.g., side views, bridge deck view)
- Annotation callouts
Inspection 2D views
- As-built schematics (orthophotos)
- Defects and annotations localized over orthophotos
Reports
- Inspection forms
End

Fig. 3. End-to-end system streamlines the process of bridge inspection with the user inputs and expected outputs.

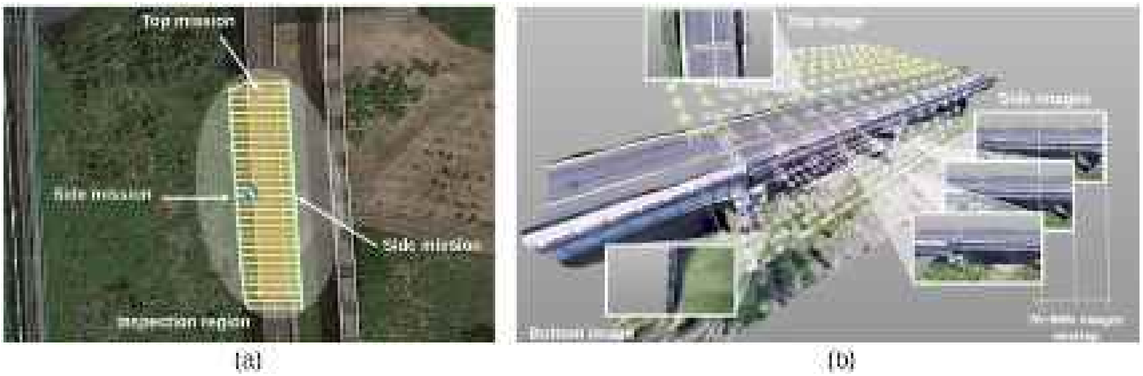


Fig. 4. (a) Defining the bridge inspection region using a 2D satellite map, the 2D flight plan is overlaid on the map to show the top and side missions; and (b) 3D flight plan generated from the 2D inspection region and the user-defined parameters around a 3D mesh model. 3D mesh is automatically generated from synthetic images extracted from a 3D map platform. (Images by authors.)

section) are based on sampled waypoints on the structure's bounding box or generated using greedy algorithms for covering the structure's topography. Accordingly, the later methods are not practical because they are highly dependent on the accuracy and completeness of the initial prior-model, and in addition they require site surveying tasks to georeference the prior model, i.e., using ground control points. Accordingly, our method allows for creating data collection missions with basic information about the underinspection structure. To allow for simple and yet accurate localization of the data collection missions, the inspection region is manually defined by means of a 2D satellite map (Fig. 4); in such cases the user can define any convex shape in 2D by drawing a polygon outlining the structure's orthographic projection in the map. The user also draws polygons of the bridge's piers on the 2D map, which are considered as obstacles so that the generated waypoints will navigate around them. Using a 2D map also provides a georeference to the data collection plan without the need for site visits. The user is only required to input basic information about the structure to automatically generate a 3D flight plan. These details or collection parameters are set based on the inspection project and the requirements of the data collection, which are the bridge's deck underside and topside altitudes from the ground, the waypoints offset distance from the bridge deck, drone and camera settings, allowed field of view to keep line-of-sight during flight execution, and the required overlap between the data frames. Using these user-defined parameters, a 3D flight plan is automatically generated. Fig. 4 shows the 3D capture path, and camera locations and trajectories (waypoints) that are automatically defined on the path. The generated waypoints are then sampled on the data collection path that is generated at an offset from the user-defined inspection region. The sampling distance is based on the offset distance and the image overlap percentage. We also provide the users with guidelines about the best-practices for setting the capture parameters, e.g., setting the overlap percentage to 60%–80% for 3D reconstruction and inspection requirements. The waypoints trajectories are set canonical to the structure to observe all the structure's elements with good resolution except at transitions between different sides of the bridge, and we introduce a gradual transition of waypoint trajectory, which was proven throughout our experiments to provide better completeness and accuracy of the 3D reconstructed models from the collected data. To account for flight execution safety, we introduce the line-of-sight requirement, which is constrained by the US FAA, to our flight planning method; accordingly, the flight plan is sliced into missions to account for the continuous visibility of the drone to the ground operator and also create missions that do not exceed a single battery flight duration so the drone will return to base after each mission for battery replacement. Based on the later requirements, each mission is defined by concatenating consecutive waypoints on the capture path until the mission's flight duration reaches a single battery life or the line-of-sight between all the waypoints in the mission reach the maximum allowed value. The line-of-sight is calculated by measuring the maximum angle between all waypoints in the mission assuming the drone operator at the vertex of such angle. Using vertices of the capture region, we visually guide the operators to stand at specific locations that provide the best lines-of-sight.

To evaluate and communicate the flight plan, we use our 3D virtual environment to render the flight plan missions along with BIM/reality models of the bridge. This visualization is useful for fine-tuning the flight plan by the bridge inspector or drone operator. Adjustments can be made to the plan's waypoints and missions (i.e., editing, adding, and removing waypoints) to improve the visual coverage and safety of the data collection execution. To support effective improvements to the flight plan, we provide a method for automatic feedback and visualization of the quality of the data collection plan, as we show in the following section. The execution of flight missions is supported via an iOS mobile application. To do so, we adapted Ibrahim et al.'s (2017b) application by replacing BIM with 3D point clouds a priori and communicate the data collection waypoints with DJI drones using the DJI Software Development Kit (SDK).

Quality Evaluation and Feedback of Data Collection Plan

The visual quality of bridge inspection data is defined by four main metrics depicting the requirements of bridge inspection data: (1) complete visual coverage to the inspection region; (2) redundant observation of bridge elements across the data frames; (3) target pixel resolution or GSD, which is required for accurate defects detection; and (4) canonical orientation of the images against bridge elements. Complete visual coverage to the bridge elements with redundant observations is important for complete and accurate 3D reconstruction. It assures the detection of several redundant visual features for successful SfM and bundle adjustment processes. High pixel resolution is important for successful automatic defect detection and classification. For instance, if a crack's width is less than the target pixel resolution, then such crack will not be detected in the collected images. The canonical camera views to bridge elements also improve the pixel resolution and minimize distortion of the defects in the images. Due to errors during execution as a result of inaccurate GPS-based navigation, the redundancy of elements' observations in the collected data also contributes to increasing the probability of complete visual coverage. It also improves the reliability of detection and prediction of defects in an end-to-end system by running the detector over each frame separately and then cross-reference the redundant detections across the frames.

For the visual evaluation process, we need to simulate the flight plan with respect to a priori 3D model of the bridge; however, such models usually do not exist for bridges undergoing UAV-driven inspection. To address this issue, we use synthetic images for generating an approximate 3D model of the existing bridge. Specifically, we recreate the simulated data collection flight plan in a 3D terrain map platform (e.g., OpenStreetMap or Google earth) and extract 2D synthetic images from the 3D environment accordingly. These images are collected according to requirements for complete and accurate 3D reconstruction (e.g., ensuring there is 70%–80% overlap between images). The images are then passed to the SfM pipeline (described in Section 3D modeling) to produce the *a priori* model used in evaluation (Fig. 4). This a priori 3D mesh model is divided into fine fragments where each fragment is used to indicate a visual evaluation element. The simulation undergoes four stages to estimate: (1) visibility of each elements in each image, (2) redundant visibility of elements across all images, (3) the average pixel resolution of each element in every image, and (4) the relative orientation of each element to each image.

To detect the observation of a bridge element in each image, we first calculate the number of element's pixels back-projected to the frame. We render the elements in a simulated WebGL scene and set a simulation camera with the parameters of the actual camera used for data collection. Next, we automatically color-encode the model's elements with unique colors based on the element index in the model. This color-encoding process allows the association of elements IDs to the back-projected pixels. A graphics engine is used for fast parallel-thread rendering of the color-encoded elements in the camera frames. For an element to be considered visible in an image, the number of back-projected pixels have to exceed a user-defined visibility threshold, which is set to assure a minimum number of element pixels exist in the frame. Next, we cross-reference the visible elements across the images to detect the redundant visibility of each element.

A similar approach is utilized to estimate the pixel resolution of an element in each data frame. First, we render the bridge elements using a depth shader graphics program that renders the bridge elements as color intensities based on their depth from the camera. During simulation, we use the same color-coding used for visibility detection to segment the depth pixels of each element in each frame and retrieve the average depth of each element using Eq. (1). We then estimate the average pixel resolution of each element using Eq. (2)

$$D_e^f = C_{near}^f + (1 - I_e^f) \times (C_{far}^f - C_{near}^f) \tag{1}$$

$$R_e^f = \frac{2 \times \tan(\frac{F}{2}) \times D_e^f}{H} \tag{2}$$

where D_e^f = average depth (in meters) of element e in data frame f; I_e^f = average color intensity of element e in data frame f; C_{near}^f and C_{near}^f = near and far planes of the camera at data frame f; R_e^f = average resolution of element e in data frame f; F = camera's vertical field of view; and H = image height in pixels.

The final simulation estimates the relative orientation of each element in each data frame. We use a surface normal shader graphics program to render the elements' surface normals as RGB colors and then we use Eq. (3) to decode the average relative orientation of each element in each image from the rendered RGB color channels

$$N_e^f = \cos^{-1}\left(\begin{bmatrix} -1+2^*R_e^f \\ -1+2^*G_e^f \\ -1+2^*B_e^f \end{bmatrix} \cdot \vec{N}^f\right) \tag{3}$$

where N_e^f = average orientation of element e with respect to data frame f; R_e^f, G_e^f, and B_e^f = average red, blue, and green color intensities of the element e in frame f; and $\vec{N}^f$ = camera's normal vector at frame f.

The results of the evaluation are presented to the user in the form of visual feedback by color-coding the bridge elements in the metaphor of traffic light colors. Using the visual feedback, the user can make adjustments to the flight plan by modifying the existing waypoints' locations and trajectories and/or adding new waypoints at locations with poor visual quality.

Image-Based 3D Modeling

In the previous section, we discussed how to assure the visual quality of the input data for 3D reconstruction and defect detection requirements. Additionally, we developed several 3D reconstruction methods to prevent incomplete and inaccurate point cloud. We specifically focus on issues that appear more frequently during 3D reconstruction of bridges; these issues include incompleteness, misregistration, and curvature (drift). Our 3D reconstruction pipeline is built on top of the typical pipeline developed by Golparvar-Fard et al. (2015), which includes the SfM algorithm followed by patch-based multiview reconstruction and mesh modeling steps. However, the repetitive structure of bridges often causes ambiguity in the feature matching process for unordered images, and eventually leads to misregistration in the point cloud. Taking the capture process as a priori, we sequentially perform feature matching in a range of n images to prevent mismatch between images with features that look alike but not correct. We developed an image cluster mechanism to group the images based on rough GPS coordinates extracted from the image metadata. The image cluster mechanism groups the images based on its GPS information into several clusters. The system then performs feature detection and matching separately for each cluster, and only connects feature tracks between each neighbor cluster to prevent mismatching of similar features that are at different locations. The cluster number is a manual input from the user based on the number of bridge piers and spans where misregistration occurs frequently. This mechanism also avoids potential errors caused by GPS with the use of sequential feature matching. Finally, the pose of the camera is then estimated and triangulated using Nister's 5-point algorithm followed by the direct linear transform (DLT) using the random sample consensus (RANSAC) algorithm. These two strategies can efficiently prevent misregistration of the point clouds. To address the curvature of the point clouds, Wu (2014) has discovered the impact of radial distortion ambiguity on the SfM process. While we found this issue could generally be prevented by an optimized capture plan, we have also developed a method that estimates the radial distortion coefficient by using the model in Brown (1971). A complete 3D

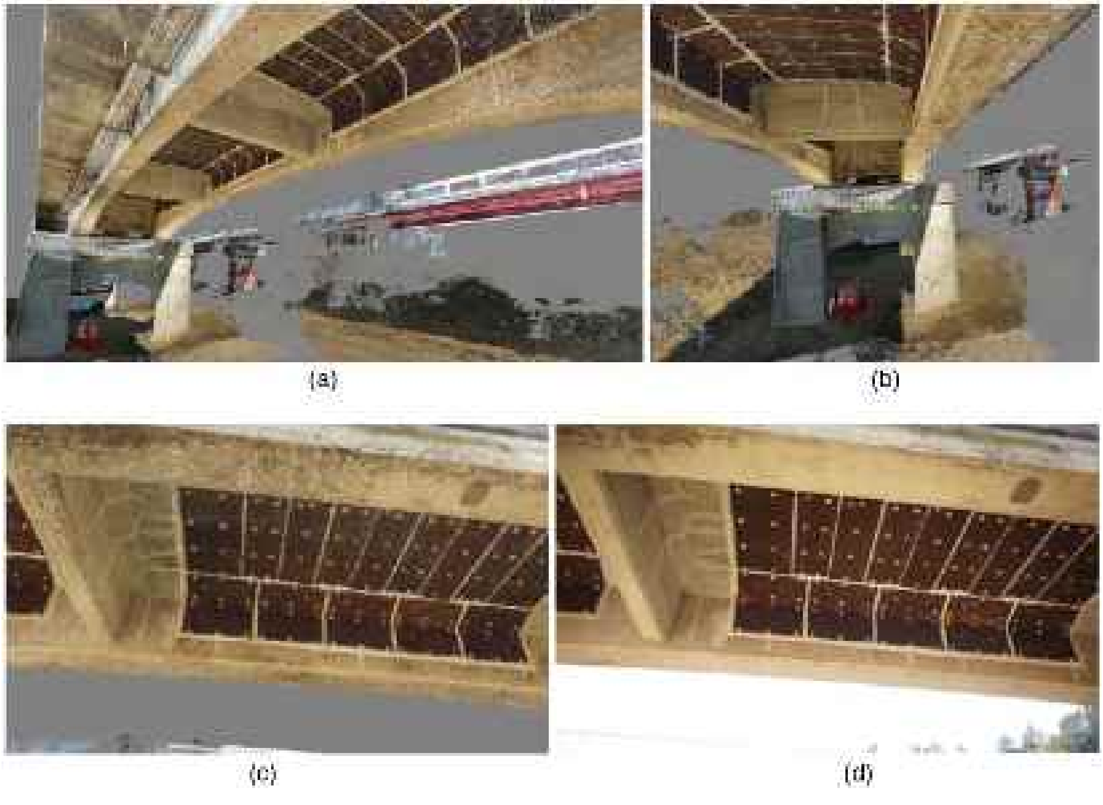


Fig. 5. (a and b) show the complete 3D reconstruction of the bridge using the optimized flight plan and the 3D modeling approach in different perspective; (c) the bridge point cloud underdeck view; and (d) the same perspective with an image overlay on the point cloud. (Images by authors.)

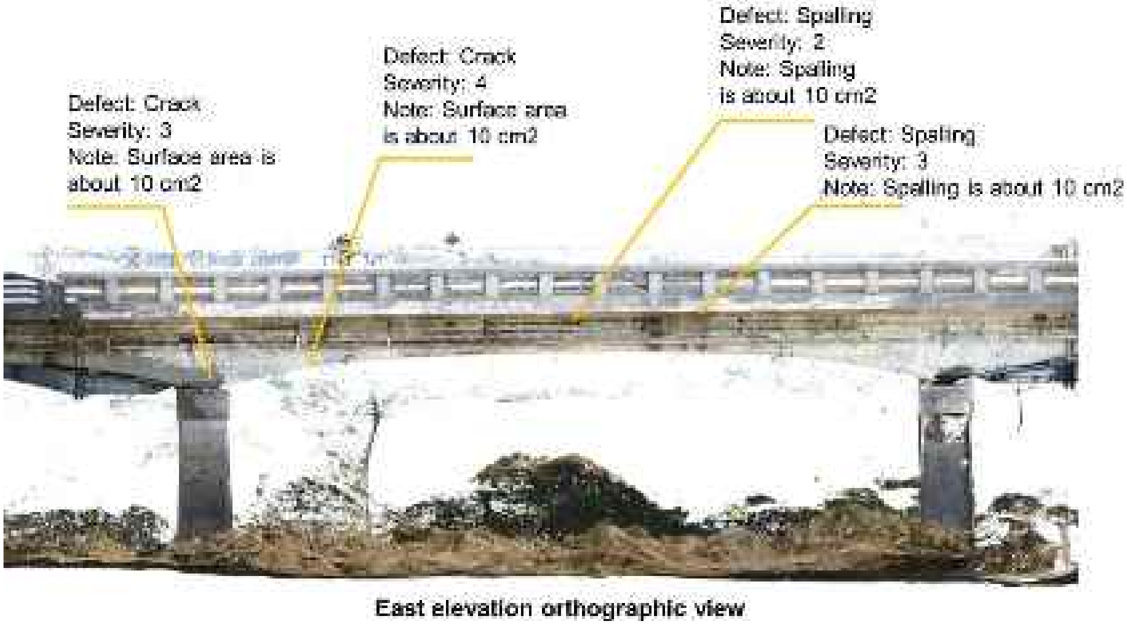

Defect: Crack
Severity: 3
Note: Surface area is about 10 cm2
Defect: Crack
Severity: 4
Note: Surface area is about 10 cm2
Defect: Spalling
Severity: 2
Note: Spalling is about 10 cm2
Defect: Spalling
Severity: 3
Note: Spalling is about 10 cm2
East elevation orthographic view

Fig. 6. Example east elevation orthographic view of one of the inspected bridges. The orthophotos are inserted into inspection reports and are automatically annotated with the result of defect detection and user-driven notes, significantly cutting down FTE time spent on report generation. (Image by authors.)

reconstruction of a bridge is shown in Fig. 5. Once the 3D mesh is generated, it is transformed into a digital depth model from any user-defined view necessary for inspection (e.g., under the deck view or side view). In our approach, these digital depth models are in the form of a raster image with pixels, arranged in rows and columns. Each pixel represents the depth value. A digital depth model is texture-mapped using images that are corrected for radial and relative depth distortions and then stitched together along seam lines to create an orthographic view. Fig. 6 shows an example from a user-defined viewpoint. Here the 3D annotations (resulting from automated defect detection or manual user-driven input, which are discussed later) are transformed from 3D point cloud into the specific orthoview. Inspection information is also organized using callouts and user-defined reporting attributes.

Damage Detection, Localization, and Mapping

We use Faster-RCNN architecture (Ren et al. 2017) and present a new algorithm to detect, localize, and determine the 3D spatial mapping of each type of damage. Our overall network consists of the feature pyramid network (FPN) in a Faster-RCNN system and a 3D spatial mapping algorithm. The FPN and the Faster-RCNN are integrated into a unified network that outputs the predicted label with its bounding box and score. In our new method, these outputs are back-projected into the 3D point clouds to analyze the intersected area and then fed into the 3D spatial mapping algorithm to determine the final labels for each fragment in the 3D point cloud (Fig. 7).

The Faster R-CNN method includes a new region proposal network (RPN) that replaces the selective search approach for finding region proposals in R-CNN and Fast R-CNN. This method improved the detection speed significantly for the Faster-RCNN architecture, and in addition, the RPN shares the backbone of CNN architecture with the object detection network Fast-RCNN to form a unified network. The input bridge images are first fed to the backbone CNN architecture, which is usually initialized with a pretrained network (e.g., VGG, Alexnet, and ResNet) to extract the feature maps. The RPN takes in the feature maps and generates regional proposal by sliding a network with fully-connected $n \times n$ spatial windows. These windows are then output into a lower-dimensional vector (512 for VGG; 1,024 for Resnet) and fed into the box-regression and box classification layers. The translation- and scale-invariant output is achieved by the k-number of anchor boxes that are set as grid and simultaneously predicting k-regions. In our method, the anchors are generated at the center of the sliding window and associated with three scales and three aspect ratios. The loss function for training the RPN is computed by assigning each anchor box a binary class label where positive labels are anchors with the most intersection of union (IoU) with the ground truth (more than 70% IoU with the ground truth), as follows:

$$L(\{d_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum L_{cls}(d_i, d_i^*) + \lambda\frac{1}{N_{reg}}\sum d_i^* L_{reg}(t_i, t_i^*) \quad (4)$$

where i = index of the anchors; d_i = corresponding predicted probability and the ground truth label d_i^* is a binary classification of the anchor; t_i = coordinates of the bounding boxes where t_i^* denotes the ground truth box associated with the positive anchor; L_{cls} = classification loss; and L_{reg} = regression loss, while these two terms are normalized to N_{cls} and N_{reg} by the batch size and the total anchor locations.

The FPN leverages the deep CNN pyramidal features hierarchy from low to high levels with bottom-up, top-down pathways, and lateral connections to build a semantically rich feature pyramid. The network takes an arbitrary size input image and outputs a collection of features computed in multiple scales from different levels of the backbone CNN architecture. The lower level features extracted during the bottom-up pathway contain better localization information while the higher-level features from the top-down pathway hold better semantics knowledge. These two different resolution features are then linked via the lateral connections, and the network could learn the spatial and semantic information through

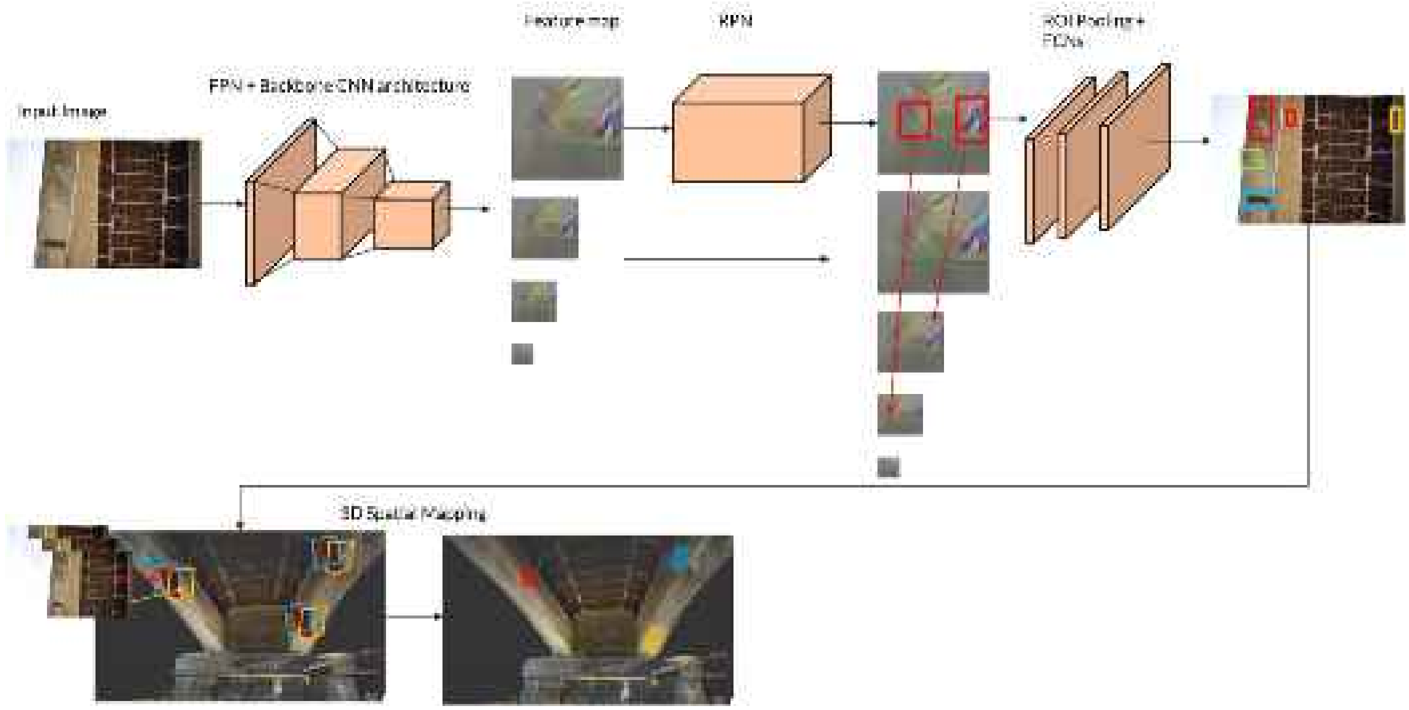


Fig. 7. Network architecture. Starting from the FPN in a Faster-RCNN system, the feature map extracted from the Resnet50 backbone and inputted into the RPN. Through the classifier, the network output the bounding boxes with the labels and scores. These outputs are projected to the 3D environment for a spatial mapping and re-injected to the algorithm to finalize the prediction. (Images by authors.)

the shared weights across different layers and scales. The FPN is independent of the backbone convolutional architecture because it is designed to use shared classifiers/regressors and fixed output channels. In this paper, we have tested with Resnet50 and Resnet-101 architectures, and we present details of our experiments in the validation section.

3D Spatial Mapping

The 3D spatial mapping leverages computer graphics techniques and the estimated cameras projection matrices from 3D reconstruction to increase the precision of defect detection by eliminating annotation and prediction inconsistencies. This step is important to ensure the completeness and consistency of the detection result among all the collected overlapped images. We first project all defects' bounding boxes from the 2D images to the 3D space by means of ray-casting to retrieve the vertices of the boxes in 3D, then we back-project these 3D bounding boxes to all the cameras that contain the corresponding defect. For each image, this process calculates all the defect bounding boxes that are detected in the spatially close images (based on distance and orientation of the cameras) and append these boxes to the original image. Ideally, these back-projected boxes should be all aligned in the same location; however, due to errors in detecting bounding boxes annotations and projection errors, these boxes are not perfectly aligned.

Accordingly, we developed algorithm 1, which eliminates these errors and increases detection accuracy by identifying the most probable back-projected boxes for each defect. This method is applied to both ground truth data and prediction results. For ground truth, we first use the IoU with a threshold t_{iou} to identify bounding boxes clusters, where only images within a user-defined 3D spatial distance d will be selected for the projection to limit reprojection error. Any box that does not pass the IoU check against the original ground truth box is removed, while we keep all the ground truth boxes even those without any intersection with reprojected boxes. Clustered reprojected boxes that have a count over the threshold t_{rp} will be further processed by performing a greedy nonmaximum suppression (NMS) to choose the most probable one. The later technique tries to improve the recall of the ground truth data, labeled by exports, because it has high precision compared to recall. We also repeat the previous process after the detection of predicted bounding boxes. First, we cluster the boxes using the same IoU threshold t_{iou} with the same allowed 3D spatial distance between frames. Only detected boxes that pass the IoU and reprojection count checks with the reprojected boxes are kept. Moreover, if there is no detection in a specific frame but there are clustered reprojected boxes of a number more than the reprojection count threshold, we assume there is a detection at that cluster and we perform greedy NMS to add a new box. Finally, if there is only an original detection in a specific frame but no reprojected boxes or the reprojected boxed do not pass the reprojection count check, then the original detection is removed. The later process was validated (as we will show later) to add missing annotations for the ground truth and predictions, and removes false positives from the predictions.

Algorithm 1. Improved defect detection using 3D spatial mapping

```
Input: A set of bounding boxes $bbox$ with spatial distance $d$ in
       image $i$
       Ground truth bounding boxes $bbox_{gt}$
       Detection bounding boxes $bbox_{det}$
Result: Reprojected bounding boxes $bbox^*_{gt}$, $bbox^*_{det}$
$bbox_{cluster} = \mathrm{IoU}(bbox) > t_{iou}$
  for $bbox_i$ in $bbox_{cluster}$ do
      if $bbox_i$ is in $bbox_{gt}$ then
          if $bbox_{gt}$ not in $bbox_{cluster}$ and $bbox_i$ count $> t_{rp}$ then
              perform NMS and add the result $bbox_i$ to $bbox^*_{gt}$
          else
```

```
            add $bbox_{gt}$ to $bbox^{*}_{gt}$
    else
        if $bbox_{det}$ in $bbox_{cluster}$ and $bbox_i$ count $>t_{rp}$ then
            add $bbox_{det}$ to $bbox^{*}_{det}$
        else if $bbox_{det}$ not in $bbox_{cluster}$ and bbox$x$ count $> t_{rp}$
          then
            perform NMS and add the result $bbox_i$ to $bbox^{*}_{det}$
```

Interactive Web-Based Viewers for Inspection

The interactive web-based viewer is extended from Lin and Golparvar-Fard (2018) with added components specifically developed for the purpose of bridge inspection. In our system, the octree structure stores a point cloud in different hierarchical levels of detail, which in turn enhances the efficiency of point cloud rendering. This representation is leveraged to only render details within the user's field of view. This allows the system to effectively deal with limited memory and bandwidth of a commodity platform, yet provide opportunities for point cloud interactions. As the user approaches, the system automatically loads the points inside the view via a lazy scheme to reduce the loading speed. The visual depth that controls the visible point size of each node is calculated as

$$vd_{node} = \frac{\sum n_i^{*} w(d_i)^{*} d_i + n^{*}_{node} w(d_{node})^{*} d_{node}}{\sum n_i^{*} w(d_i) + n^{*}_{node} w(d_{node})} \qquad (5)$$

where n_i = number of points at node i; d_i = depth of node i; and $w(d_i)$ = level weight for node i. Index i goes over all nodes and ensures the point size of each depth remains consistent while changing views. In addition to using the environment for planning visual data collection and evaluating capture quality, the virtual walkthrough for bridge inspection satisfies another main bridge inspection requirement, which is conducting measurements. Various measurements such as length, area, and volume are needed to support the bridge inspection workflow, and we have developed measurement tools for the 3D point clouds and 2D images. The measurement tools for 3D point clouds utilizes ray-casting to retrieve the user-clicked point in 3D through the intersection of the ray with the point cloud. We improved the ray-casting accuracy by implementing a cylindrical ray to find the intersected point. The cylinder is defined by the origin and the target points with a radius r, and it returns the first hit with the cylinder with a moving direction. This approach provides an accurate result when the point cloud is sparser and with varying depth. The image-based measurement (Fig. 8) tools utilized the camera projection matrix "calcula ted" during the SfM process to project 2D points in the images into 3D point clouds. We use intrinsic and extrinsic camera parameters and ray-casting to project the 2D points the user selects in the image to the 3D space for measurement purposes.

To support inspectors via virtual walkthroughs, the web-based viewer provides a nonlinear transformation from 3D point clouds to 2D orthophoto. The 2D orthophoto is generated automatically by using the feature matches, estimated depth, and gravity direction from 3D reconstruction. The camera locations of the capture are projected to the 2D orthophoto based on the estimated 3D pose,

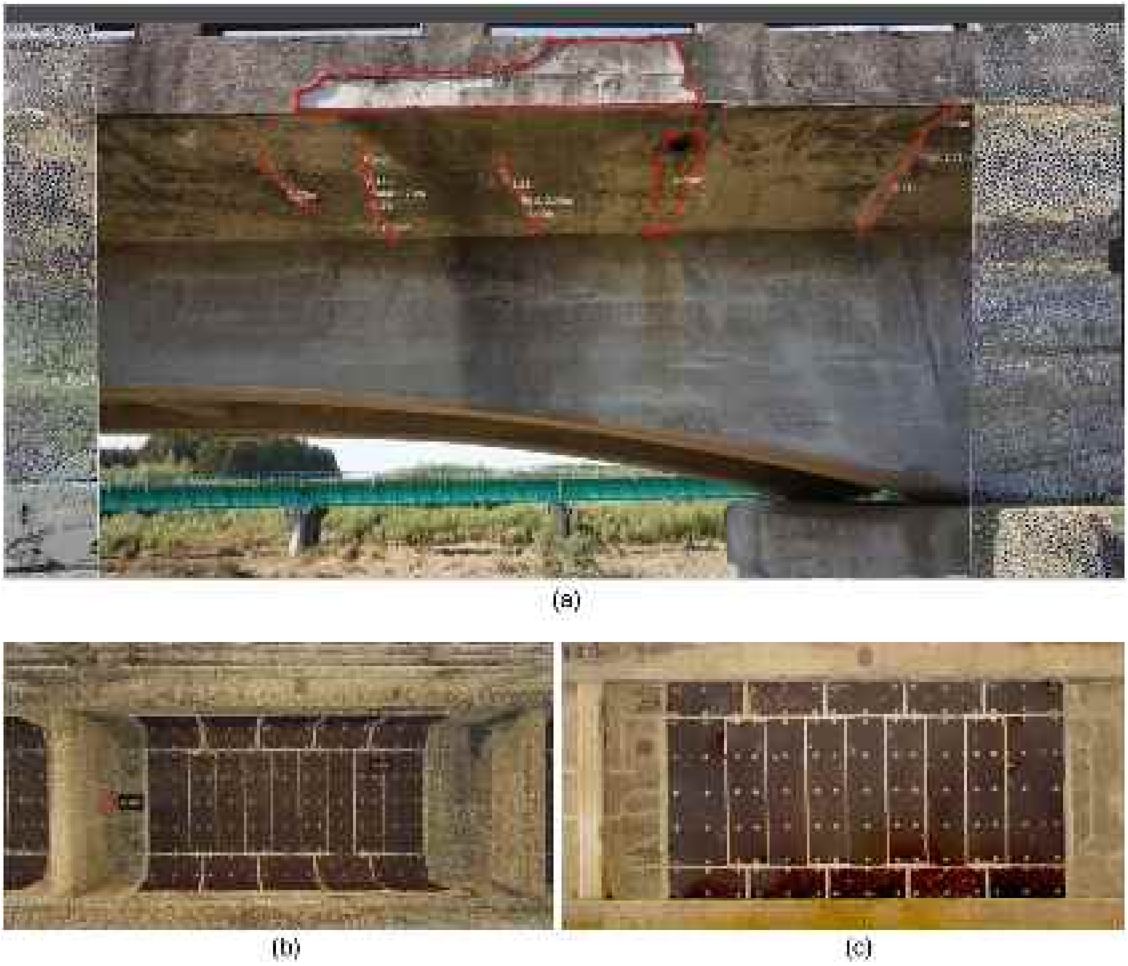


Fig. 8. (a) Image-based measurements using improved ray-casting to point cloud; (b) pin tools that can be used to document defect information during virtual inspection; and (c) example of mapping 2D orthoview from the bridge underdeck. (Images by authors.)

and the user can easily switch between different views according to the user needs. Using all images and point clouds associated with their capture date, a 4D (3D+time) point cloud timeline is formed chronologically to provide complete documentation throughout the bridge lifecycle. Each capture can be analyzed against a previous capture via side-by-side or overlaid image comparison interfaces. To do so, we query the closest image viewpoint and location from the previous capture and transform that into the new 3D view using the following McMillan & Bishop warping Eq. (6)

$$x_2 = \delta(x_1)P_2^{-1}(c_1 - c_2) + P_2^{-1}P_1x_1 \qquad (6)$$

where the first term represents the move pixels from x_1 to x_2 based on camera optic center c_1 to c_2; and P_1 to P_2 = camera projection matrices. The second term is for texture mapping. The per-pixel distance values are used to distance the warp pixels to their correct location for the current camera position. Using similar views (side-by-side or overlay), the inspectors can easily observe the progression of the defect over time and assess the severity.

Report Generation

Each inspection includes essential reporting forms that: (1) present schematic views to various bridge components and relative location and severity of defects (e.g., an under deck view with clear mapping of various defects); and (2) defect images as well as their associated inspection info. To minimize the time necessary for creating these reports, our process starts with leveraging defects and associated properties (such as severity, notes, and inspector's name) that are already mapped in the point cloud through the 3D virtual inspection. These properties can also be customized and color-coded to directly correlate to the fields in the inspection report. A pin placed on an image has the image attached automatically for the generation of the reports that depict defect images and their associated information. This information is then transformed automatically into several excel sheets with hyperlinks that directly connect to the 3D location in the viewer upon request. The information includes the images, associated properties, and the GPS location of each pin when available. Inspectors can easily access all the information that needs to be used to assess the defect from these excel sheet that follow owner reporting guidelines.

Another essential tool that assists inspectors in generating inspection reports is the surface mapping tool. Our tool creates 2D orthophotos from different viewpoints or by unfolding the structure from different shapes (such as bridge elevation or under the deck view). This tool detects the surface of the structure and maps the surface into a 2D orthophoto, which is more informative compared to schematic drawings that are provided in these reports. This 2D orthophoto automated the process of creating schematic drawings. It also provides an overview of the inspected structure and enables analyzing the overall severity quickly with the location of all pins (Fig. 1).

In conclusion, our method streamlines the process of automated bridge inspection including data collection, processing, organizing, analytics, and report generation. The contribution of each module and submodules are summarized in Table 1.

Technical Validation

We validate the end-to-end bridge inspection and defects reporting pipeline via 30 bridge inspection projects in the United States and Japan. Here, we focus on showing results from two inspection projects for simplicity (Table 2). One span of each bridge is shown here focusing on the underside and lateral-sides of the bridge's deck. In addition, the two piers supporting the deck are shown as well. The inspection was conducted to detect and localize the five defects classes discussed previously in the method section. Detailed results for each stage of the inspection workflow are presented in the following sections.

Table 1. Comparison of our method to related work

Modules	Morgenthal et al. (2019)	Ibrahim et al. (2017b)	Mundt et al. (2019)	Our method
Automatic flight plan	v	v	—	v
D flight plan	v	v	—	v
Flight plan evaluation	v	v	—	v
Bridge-based 3D reconstruction	—	—	—	v
Defect detection and classification	v	—	v	v
Multiple defect detection	—	—	v	v
Defect localization	v	—	—	v
Viewer	—	v	—	v
Multiple point clouds	—	v	—	v
Measurements	—	v	—	v
Report generation	—	—	—	v

Table 2. Inspection projects used in validation

Structure information	Inspection project 1	Inspection project 2
Structure's type	Concrete beam bridge	Concrete beam bridge
Structure's age	29 years	16 years
Inspection length	36.0 m	200.0 m
Inspection width	6.0 m	22.5 m
Deck's top altitude	6.8 m	19.4 m
Deck's bottom altitude	4.5 m	18.2 m
No. of inspected piers	2	3

Table 3. Flight plan parameters

Parameter	Value
Top offset	5 m
Bottom offset	3 m
Sides offset	3 m
Images overlap	75%
Drone battery life	20 min
Drone speed	25 km/h
Line-of-sight FOV	45°
Aerial platform	DJI Phantom4

Data Collection

To create a flight plan, the developed web-interface was used for select the inspection region on the 2D map. The selected area and geofenced zones are defined by drawing polygons over the bridge section that will undergo inspection. Next the user sets the data collection parameters based on the inspection region dimensions, bridge's geometry, safe flight distance away from the bridge deck's sides, line-of-sight requirements for safe navigation, and the utilized aerial platform. Table 3 provides all the defined parameters for the inspection flight plan in one of these inspected bridges.

Using the defined inspection polygon and parameters a 3D flight plan was automatically generated with four missions covering the top, bottom, and both sides of the bridge. While we mainly focus on the underdeck and sides of the bridge, the top waypoints are needed for complete 3D reconstruction of the bridge. Furthermore, to exclude the underside region occupied by the bridge piers from the flight plan, we manually define the locations of the piers using the 2D map and accordingly all waypoints lying inside the bridge piers regions are removed from the bottom mission. The waypoints are set to canonical trajectories with the bridge sides except for the top row of the side missions, which are set to 45° nadir to account for smooth image transition between top and side missions; this transition strategy is needed for accurate 3D reconstruction. The total number of generated waypoints are 1,309 with 217 waypoints for the top mission, 693 waypoints for both side missions, and 399 waypoint for bottom mission. Such distribution of waypoints is based on the user-defined offsets, camera parameters, and image overlap percentage, which are used to calculate the distance between consecutive waypoints. The total flight distance is 1,871 m, which requires around 64 min of flight accounting for the transition between waypoints and capturing an image at each waypoint. Accordingly we informed the drone operator that 4 drone batteries will be needed for completing the flight and collecting all the data. Fig. 9 shows the generated 3D flight plan and waypoints for each mission visualized with the latest reality model of the bridge in the developed 3D virtual environment.

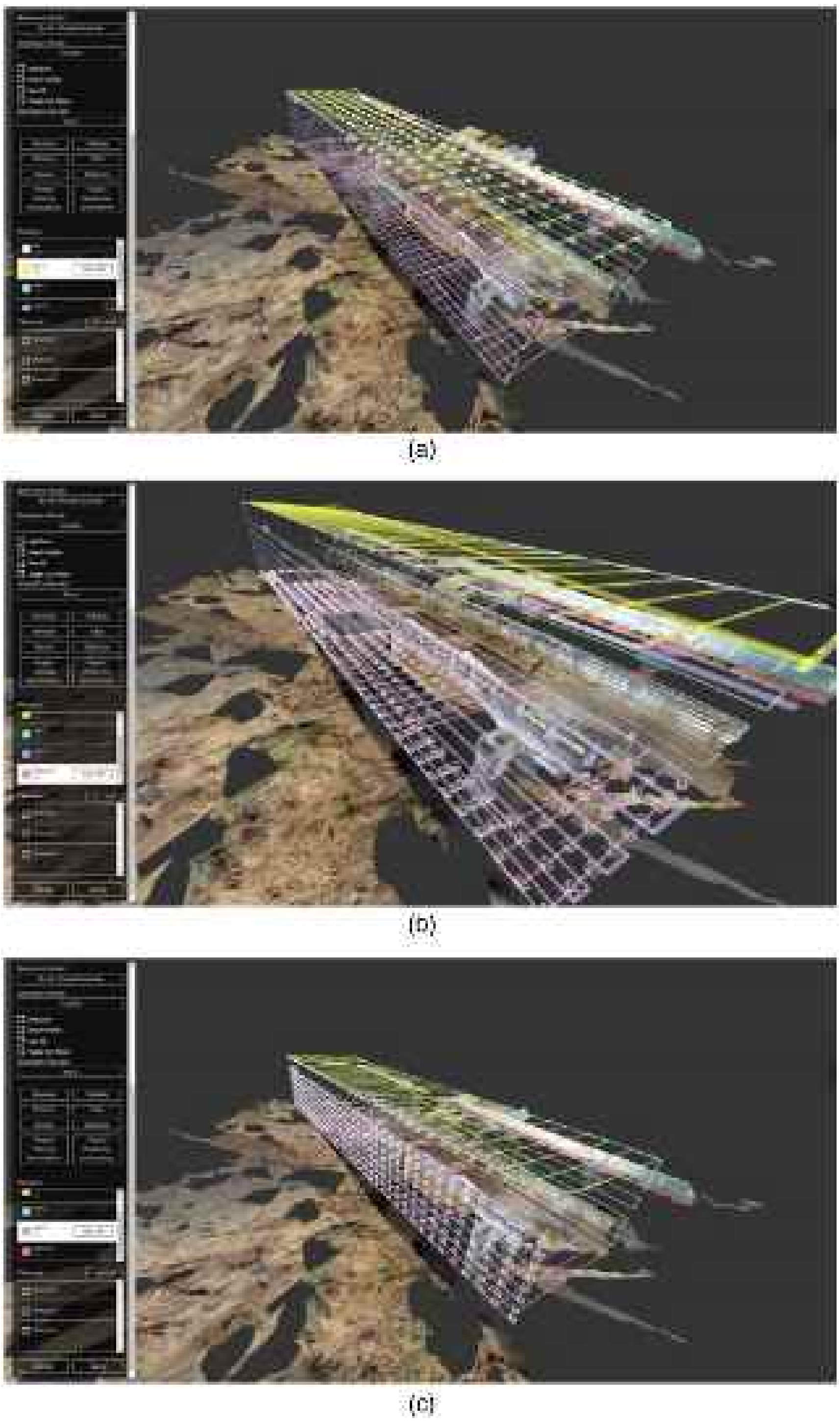


Fig. 9. Generated 3D flight plan with (a) top mission; (b) bottom mission; and (c) one of the side missions. (Images by authors.)

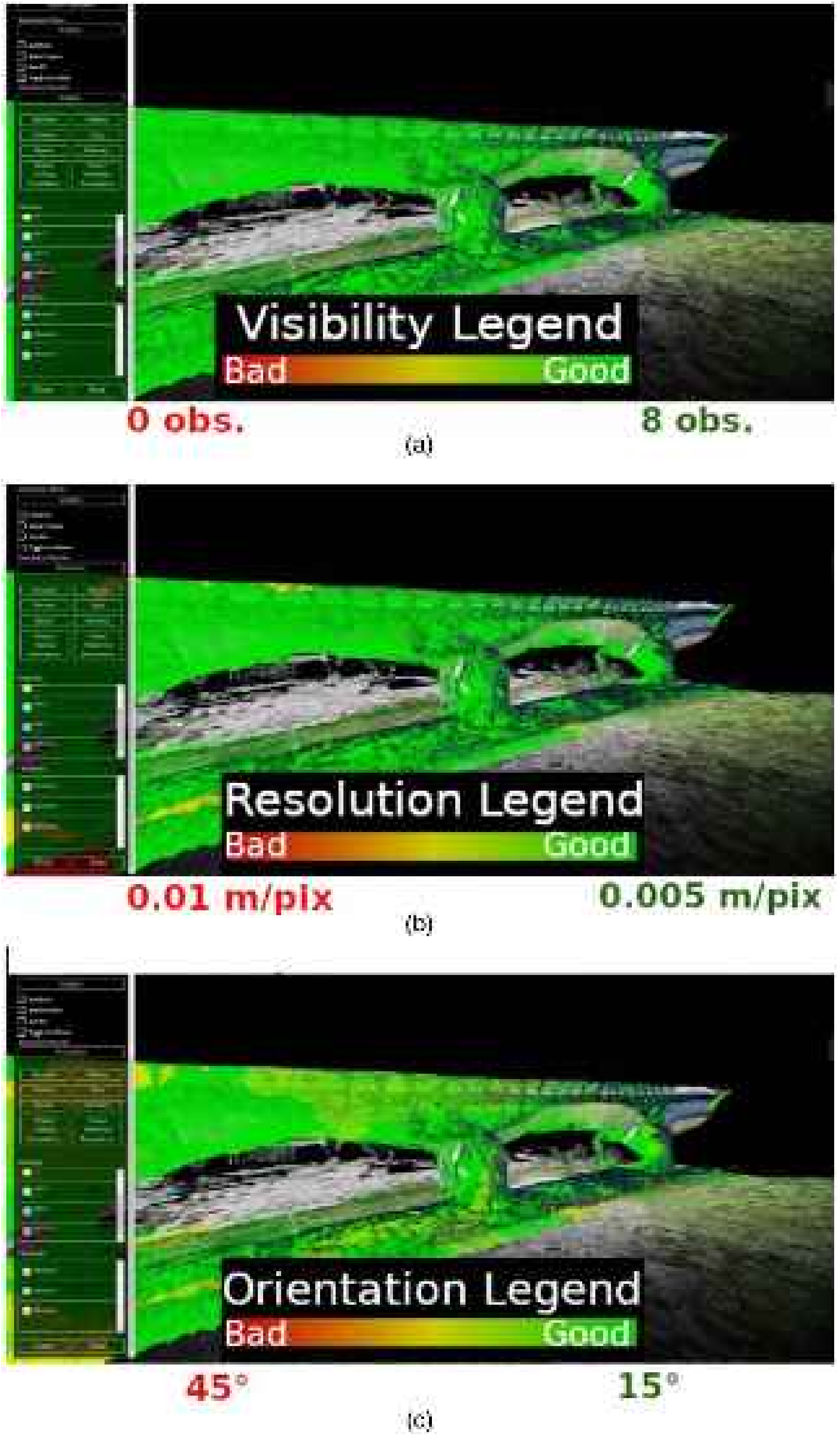


Fig. 10. Color-coded visual feedback for data collection plan: (a) visibility evaluation; (b) resolution evaluation; and (c) orientation evaluation. (Images by authors.)

Visual Quality Evaluation

Once a flight plan is generated, the data collection evaluation is executed to provide feedback to assess the quality of the data collection plan. Fig. 10 shows an example visual feedback for visibility, resolution, and orientation evaluations. The bridge mesh was divided into 45 elements (fragments), which are colored using traffic-color metaphor (red, yellow, and green) to show the visual quality per element, where elements colored in green are satisfying the evaluation criteria, red colored elements are not satisfying the quality requirements, and elements colored in the yellow range between green and red are acceptable and are evaluated subjectively. For instance, the resolution of some elements on the top edges of the bridge are on the yellow range, but it is not required to detect small cracks at these edges. Table 4 shows evaluation results within the acceptable range of each quality metric. These results show that a complete visibility of all the inspection regions was achieved with

Table 4. Flight plan evaluation results

Metric	Criteria	No. of elements	% of elements
Visibility	Not met (<3 observations)	0	0.00
	Acceptable (3–8 observations)	2	4.44
	Satisfactory (≥8 observations)	43	95.56
Resolution	Not met (>0.01 m/pix)	3	6.67
	Acceptable (0.005–0.01 m/pix)	3	6.67
	Satisfactory (≤0.005 m/pix)	39	86.67
Orientation	Not met (>45°)	0	0.00
	Acceptable (15°–45°)	14	31.11
	Satisfactory (≤15°)	31	68.89

the required redundancy in observations per element except for three elements that were observed but were not redundant in at least eight images. The visual feedback is used to locate those elements that were found at one of the ends of the bridge mesh outside the inspection span so the overall visibility evaluation was satisfactory.

The satisfactory range for resolution metric was set subjectively based on the requirements of the inspection report. Accordingly, for our experiment complete and accurate automatic inspection requires the detection structural cracks that are at least 5 mm in thickness, so the satisfactory resolution was set to 5 mm per pixel. We also set a maximum acceptable resolution of 1 cm per pixel to account for observing other defects types and thicker cracks. Because we also provide visual feedback, the regions with acceptable but not satisfactory resolution were addressed subjectively according to the importance of detecting small cracks at those regions; for instance, 5-mm cracks at the top edges of the bridge's deck are not critical to the structural integrity of the bridge. Results of resolution evaluation show that 86.67% of inspection fragments are visible with a maximum of 5 mm per pixel resolution in at least one data frame and 6.67% of the elements are still in the acceptable resolution range. According to the later evaluation values, more focused analysis was conducted using the visual feedback to determine the regions with bad resolution. It was found that three inspection fragments are outside the acceptable range for resolution, which happen to be on the ground and at one of the ends of the bridge outside the inspection region. We also detected another three fragments that have acceptable but not satisfactory resolution (between 5 mm and 1cm of thickness). But those fragments were detected to be at the top edges of the bridge's deck where 5-mm cracks are not required to be detected. Based on the previous analysis, we concluded that the resolution evaluation is satisfactory. In future work, we plan to remove the ground and regions outside the inspection span from resolution and orientation evaluation while keeping them during the visibility evaluation because the ground below the bridge is still needed for complete 3D reconstruction.

Evaluating the orientation of the simulated bridge elements shows that 68.89% of the elements are almost canonical at a relative angle of 15° or less to one frame at least; the remaining 31.11% of the elements were still in the acceptable range and they are located on the ground, at one of the ends of the bridge and at the top edges of the bridge deck. Similar to resolution evaluation, because both orientation and resolution metrics are used for improving defects detection model, we did not need to improve the flight plan to capture the ground or the region outside the inspection span with high resolution or canonical camera orientations.

Damage Localization, Detection, and Mapping

Dataset Preparation and Annotations

The dataset for our scientific validation was collected using the automatically generated and optimized flight plans for the two bridges. The captured images were carefully labeled into the five common defect types: crack, spalling, efflorescence, corrosion stains, and exposed reinforcement rebar. A web-hosted annotation tool extended from the computer vision annotation tool (CVAT) was developed for the annotation process.

Annotation

The collected inspection images were curated and hosted on the web-based annotation tool. Annotation tasks were split into batches of 10 images per annotation job for better organization, easier access, and workload management. To maintain uniformity in annotations from different annotators, common guidelines were created and shared with the annotators showing examples of defects found in images and corresponding annotations. A total of 28 engineering experts served as annotators in this study. The steps followed for annotating are: (1) all defects belonging to one class are annotated first in an image before moving to the next class; (2) a specific geometric shape for the corresponding class label is selected to annotate defect area before making the annotations (geometric shapes used for annotating defect area are polylines for Cracks and polygons for the other four defect types); (3) after finishing annotating one class, the annotator moves to annotating the next class; and finally (4), each completed annotation job is reviewed and validated by two engineering professionals similar to the quality assurance/control process of Liu and Golparvar-Fard (2015). An example of the annotations is shown in Fig. 11 and the resulted dataset attributes are detailed in Table 5.

Dataset Statistics

Most images are observed to have multiple types of defects with defect regions overlapping in some cases. The choice of structures for image acquisition has resulted in some defect class annotations being more prevalent than others. This has been observed to not impact the detection experiments since ample number of images were available for each class. The dataset is split into training and testing sets using the 80/20 convention. In total, 653 images (14,302 defect instances) were used for training and 89 images (4,804 object instances) were used for testing.

Experiment

We implemented FasterRCNN architecture in Pytorch with five object classes including crack, spalling, efflorescence, corrosion stains, and exposed reinforcement rebar. The Faster-RCNN model is fine-tuned with the MS-COCO object detection model with six output classes, including the background. Because the MS-COCO dataset is very different from our dataset, we only utilized the fine-tuned model to expedite the initial training process and still trained the model from scratch for every layer. The object detection model is trained with the stochastic gradient descent (SGD) optimizer with a momentum of 0.9. We set the learning rate using a cosine annealing schedule that cycles between 0.00001 to 0.01 in every 10 epochs. The learning rate setting prevents the training from getting stuck in local minima and stops improving the model. We have also used data augmentation to increase the variation in the dataset.

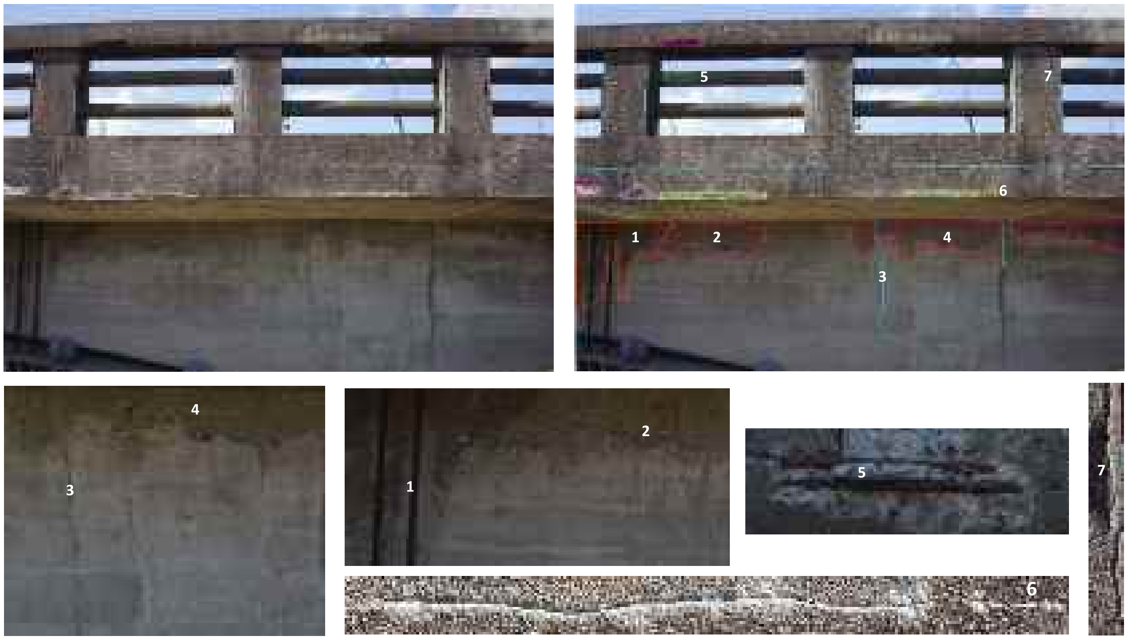


Fig. 11. Dataset example. (Images by authors.)

Table 5. Results from the annotation process and detection

Defect class	Number of annotations	AP (%)	w/mapping AP (%)
Spalling	5,013	84.6	85.3
Exposed rebar	1,545	84.5	85.2
Corrosion stains	3,310	74.1	74.9
Efflorescence	1,944	57.6	61.1
Crack	5,779	49.2	53.2

Several different techniques were used, including flipping, rotation, shearing, and changing brightness and contrast. The cropping augmentation is based on the defect locations and sizes to retain each defect's original high resolution in the image. The loss functions of the object detection are the object classification and L1 bounding box regression losses. The object detection model uses the NMS algorithm to remove overlap boxes inferring the same object. This process is vital for damage detection as much damage could cover a large area with smaller defects, and the model without NMS could predict various sizes of boxes overlap with each other. We used the top 80 predictions before and after the regional proposal network and a 0.2 NMS threshold for the final predictions. The detection results are shown in Fig. 12.

The average precision (AP) is used to evaluate the result for object detection. Precision is calculated by dividing the correctly detected bounding boxes (True Positives) to all the detected bounding boxes (True Positives + False Positives). The bounding box is only correctly predicted when: (1) the IoU of the prediction and the ground truth is over 0.5; (2) the predicted class of the box and the ground truth class is the same; and (3) the score of the prediction is above 0.3.

Experiment Results and Discussion

The Faster-RCNN object detector with Resnet50 and FPN system reaches 84.6% AP for spalling, 84.5% for exposed rebar, 74.1% for corrosion stains, 57.6% for efflorescence, and 49.2% for crack as shown in Table 5. To rigorously evaluate performance, we conducted error analysis inspired by Hoiem et al. (2012) to better understand the network's performance. The analysis produces a Precision-Recall (PR) graph with six types of error curve for examining the false positive errors. These six types include: (1) PR curve at IoU = 0.75 (C75); (2) PR at IoU = 0.50 (C50); (3) PR at IoU = 0.1 (Loc) where the localization error is ignored, which indicates that the damage is detected, but the localization is incorrect; (4) Sim, which only applies when a super-category is used in the detection; (5) Oth, i.e., PR after the classes are removed, which means detection without classification; (6) BG, i.e., similar to the previous one with also removing the background class; and the False Negatives (FN) that shows the PR after all the errors are removed. The area under the curve represents the corresponding AP for each type. With the analysis, we observe that the main improvement can be made by improving the FN of the model (Fig. 13). The FN area shows that some of the damages were not detected in some settings. The NMS setting for the model has a trade-off of the number of detections versus the overlapping detections; we have chosen to reduce the overlapping detections. This trade-off also reduces the total amount of received detections. We fine-tuned the NMS settings with various tests to balance the trade-off for the best result. Also, the detection result is being continuously improved through the virtual inspection process in the interactive web viewer. The users verify the correct defects and adding the missing ones simultaneously for documentation and reporting; the system then collects these results to expand the dataset and retrain the detection model.

The 3D spatial mapping is the next in the algorithmic pipeline after the detection model. We used this mapping for both the ground truth and detection to increase the consistency of the labels. We set the t_{iou} to 0.1 to identify any box that has contact with the original ground truth or detection box; this threshold allows us to eliminate the projection errors and always keep the original ground truth or detection. The projection count threshold t_p is set to 2, where it only keep the boxes if it appears more than two times. The results show that the 3D spatial mapping can efficiently increase the AP of crack and efflorescence by more than 4% and

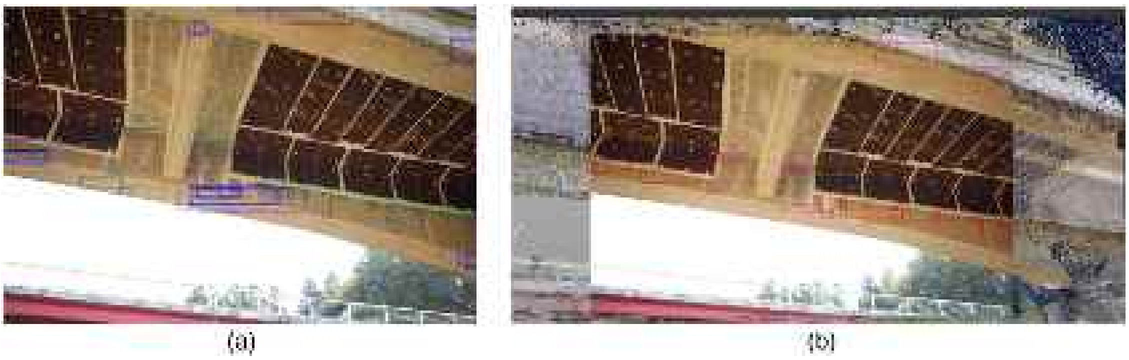


Fig. 12. (a) The detected bounding boxes are shown with boxes mapped from other images during the 3D mapping process; and (b) the detected bounding boxes are shown in the web viewer with the 2D image overlay on top of the point cloud after the 3D mapping. (Images by authors.)

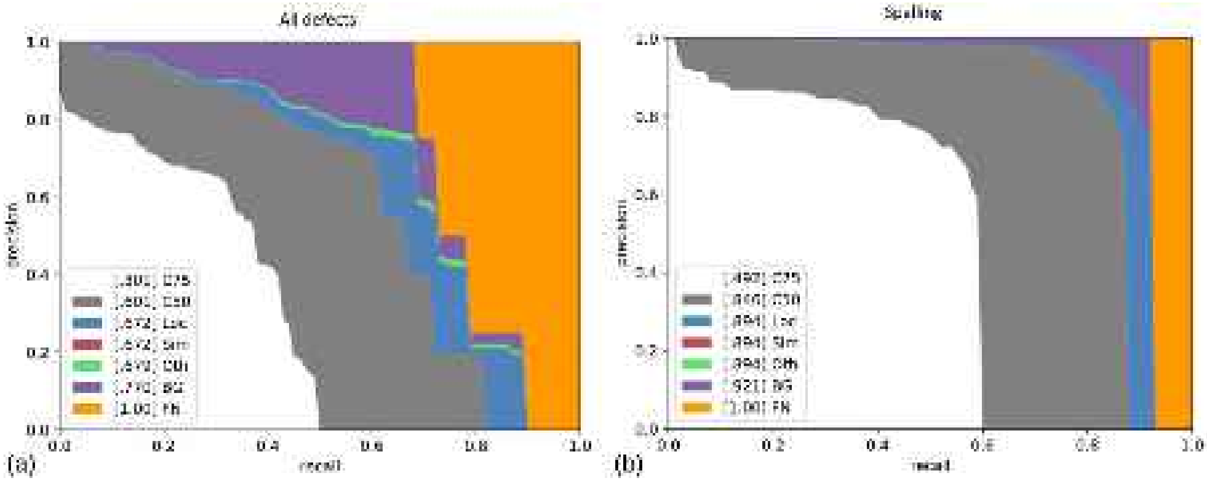


Fig. 13. Trained faster-RCNN detector error analysis: (a) all object instances; and (b) all spalling instances.

shows an improvement for all the defect types (Table 5). However, the improvements are more notable for crack and efflorescence, while these two types have relatively lower AP without 3D spatial mapping compared to other types. We have observed that these two types of defects are generally harder to annotate with the same standard across multiple images. The annotators tend to miss the same cracks and efflorescence in different images. The 3D spatial mapping has successfully eliminated the inconsistency between the ground truth images. On the other hand, it has also improved the detection result. The 3D spatial mapping can be improved by optimizing the threshold selection process. The current threshold of nonmaximum suppression has a large range of values to select from and differs from the detection model. However, this could also be achieved through an NMS threshold learning process in the detection framework. We plan to explore a learning approach in the future to reduce the optimization time.

Web-Based Viewer and Report

We followed the Charette test (Clayton et al. 1998) to evaluate the user feedback on the web viewer and the reporting process. The Charette test validates the process by measuring several essential aspects when new workflows are introduced to engineering domain. These aspects include effectiveness, repeatability, and reliability. The effectiveness is measured by the increase of productivity when conducting a virtual bridge inspection in the web viewer and generating reports. We measured the repeatability by observing if different users can perform the same inspection tasks under different projects. Also, the reliability is measured by the likelihood of the system providing similar results when the user interacts with different inspection conditions. These are measured qualitatively by observing and interviewing real world practitioners interacting with the system.

We trained 65 users including bridge inspectors, transportation professionals, and civil engineers to perform the tasks specifically for a virtual inspection on: (1) transitioning from 2D orthoview to 3D view with image overlaid; (2) using measurement tool to analyze the damage; (3) creating pins with additional properties; and (4) creating reports with 2D orthoview and images. Through the training, all users expressed that the process could significantly reduce the time they spent on localizing the defects and going through the corresponding images to create the reports. They were able to locate the area for inspection in 2D views quickly and transition to a 3D view with images to perform detailed inspection tasks such as measurements. Pins are created in both point clouds and images to document the inspection results and generate reports. The users have expressed that even though the excel sheet is not entirely the same as the required report, the output can easily be transformed to actionable format. The following section discusses how the methods presented previously could facilitate the transition from the current practice to a fully automated system.

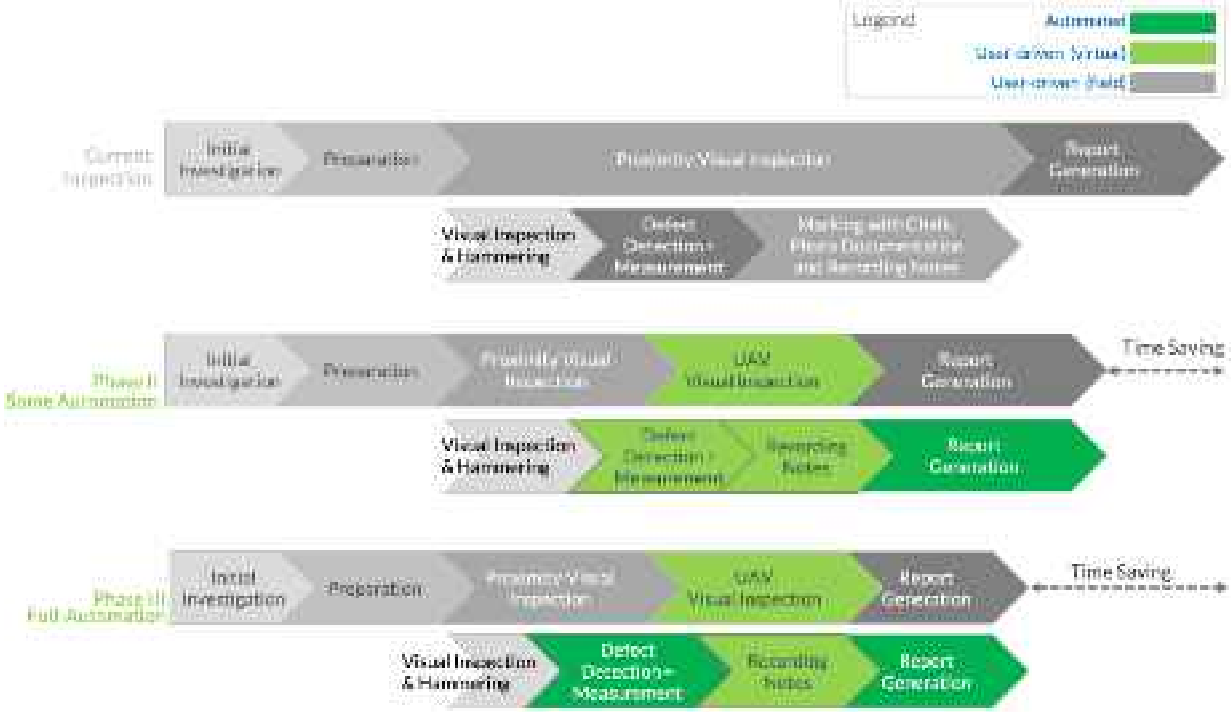


Fig. 14. Transition to a fully automated bridge inspection process can be broken down into three phases where more automation is introduce one step at a time.

Discussion: Gradual Transition to a Fully Automated System

In previous sections, we have presented a system that includes model-driven visual data collection, quality evaluation, feedback of data collection plan, image-based 3D modelling, damage detection, localization, and mapping and interactive web-based viewers for inspection. We also presented an in-depth analysis for each component of the system separately and as a whole. We showed that an accurate and comprehensive defect detection requires complete and accurate visual data collection where the accuracy of the defect inference model relies on the number of pixels the defects occupy per frame and the redundant detection of the defect across different data frames for 3D spatial mapping. The data collection planner evaluates such requirements through a simulation process before collecting the data. Accordingly, the visual evaluation metrics (redundant visibility, resolution, and orientation feedback) are the basis for improving the accuracy of automatic defects detection. As shown in Table 5, redundant high resolution observation and detection of defects led to improvements for all defect types between 0.83% and 8.13% through 3D mapping with the highest improvement in cracks from 49.2% to 53.2% AP. We also showed that high-quality image-based 3D reconstruction is vital for correct localization and mapping of defects in 3D, making it easier to spatially document the defect and conduct successful 3D mapping. Finally, for enabling actionable feedback for bridges maintenance prioritization, the system generates automatic reports supported by 3D visualizations and 2D orthophotos projections.

Additionally, we have learned through deploying the system on 30 bridge inspection projects in the United States and Japan that implementing the fully automated system at once is not possible. Through focusing on documenting, communicating, and following up on inspectors' recommendations, we have developed a process that facilitates the transition from a manual to automatic workflow. The key to automating this process is to gradually transition from the manual workflow to semi-automatic and eventually to fully automatic because it involves many components and field personnel. Based on many discussions and interviews with bridge inspectors, the process to systematically automate bridge inspection can be broken down into three phases, including the current practice, which is Phase I (Fig. 14). This process is being actively implemented in the US by bridge inspectors and in Japan, by the Ministry of Land, Infrastructure, Transport, and Tourism as part of an effort to streamline inspections.

Phase II is to automate the data collection process and the report generation part. In current practice, investigation and preparation happens before the hands-on inspection as the initial step. During the hands-on inspection, the inspector performs visual inspection and physical examination (marking, sounding, measuring, and capturing). Furthermore, all the information collected on the field is compiled into an inspection report in the office. The first transition can be done by automating the data collection process that facilitates the process of measuring, capturing the damages, and organizing data for report generation because the inspector spends a significant amount of time measuring, capturing, and organizing the damages and related information. An automated robotic solution that collects data and a web-based system that processes the data can be the transition phase towards automation. Throughout this phase, the system is also actively improving the automatic detection models and expanding the defect dataset via: (1) user input to correct detected defects or adding the missing ones, conducted through quality assurance/control steps similar to Liu and Golparvar-Fard (2015); and (2) retraining the model on a regular process with the larger ground truth dataset.

In this work we showed that the proposed data collection planning process assures the visual quality of the collected data for bridge inspection and report generation requirements. However, the optimization of the automatically generated flight plans is still manual. Therefore, an optimization model is needed to automatically generate flight plans that satisfy the requirements without further user input. Additionally, the current data size is sufficient with regard to the instance number, and in turn we aim to expand the data through collaborations between academia and the industry. In the future, we will focus more on improving the data collection and evaluation process by automatically optimizing the flight plans and removing simulated fragments that are not needed for each evaluation metric as the ground or elements outside the inspection region. We also plan to improve the results of our models by collecting and annotating more data to capture a more generalized distribution of the defects shapes and colors. In terms of the detection model, we plan to use the 3D geometry information generated through the flight plan and reconstruction process to improve the detection network.

Conclusions

This paper presents an end-to-end system for automating a robotic bridge inspection process with the integration from flight mission generation, assessment for visual quality of the data, assuring accuracy and completeness of the reconstructed reality model, 3D reconstruction for elevated structure, automated damage detection and localization, and report generation. The new system enables a new virtual bridge inspection workflow that collects quality data with complete 3D point clouds for automated damage detection and report generation. This addresses the current challenges of deploying UAVs for bridge inspection and further extends the opportunity to adopt a fully automatic process. The results show that the system can automatically generate a quality ensured flight plan and detect the defects with an AP of 85.3%. The workflow has shown the connections between each component and how to streamline a robotic bridge inspection solution systematically and gradually. The introduced phase-based implementation towards full automation enables continuous and active learning for the detection models through constant inputs collected from the users and enhanced ground truth data. This approach could eventually build a substantial bridge defect dataset that could contribute to a full-scale and reliable field implementation at any national level. Through the validation on several bridges, high-resolution images are collected through an optimized simulated flight plan and used to generate the 3D point clouds. Damages are detected and localized in the 2D and 3D point clouds, and this information is used for generating reports in compliance with bridge owner requirements. The proposed end-to-end system provides a holistic approach that connects the dots between robotic bridge inspection components with validated examples.

Data Availability Statement

Data generated or analyzed during the study that support the findings of this study are available from the corresponding author upon reasonable request (specifically, the labeling software, images, and point cloud models).

Acknowledgments

The authors would like to thank Yoshihiko Fukuchi, Katsunori Yasui, the RAAMAC Lab, the entire PWRI office at Japan MLIT, and all participants of the annotations for their suggestions and support to this research. The support and help of all bridge inspection companies in the US and Japan who were involved in collecting data and implementing the system is greatly appreciated. This material is in part based upon work supported by the National Science Foundation #1446765. The opinions, findings, and conclusions or recommendations expressed are those of the authors and do not reflect the views of the NSF, the companies, or the participants mentioned above.

References

AASHTO. 2018. *35 state DOTs are deploying drones to save lives, time and money*. Rep. No. 1. Washington, DC: AASHTO.

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 (Apr): 180–194. https://doi.org/10.1016/j.autcon.2013.06.011.

Agnisarman, S., S. Lopes, K. Chalil Madathil, K. Piratla, and A. Gramopadhye. 2019. "A survey of automation-enabled human-in-the-loop systems for infrastructure visual inspection." *Autom. Constr.* 97 (Jan): 52–76. https://doi.org/10.1016/j.autcon.2018.10.019.

Almadhoun, R., T. Taha, L. Seneviratne, and Y. Zweiri. 2019. "A survey on multi-robot coverage path planning for model reconstruction and mapping." *SN Appl. Sci.* 1 (8): 1–24. https://doi.org/10.1007/s42452-019-0872-y.

Baik, H., and J. Valenzuela. 2019. "Unmanned aircraft system path planning for visually inspecting electric transmission towers." *J. Intell. Robot. Syst.* 95 (3–4): 1097–1111. https://doi.org/10.1007/s10846-018-0947-9.

Bang, S., H. Kim, and H. Kim. 2017. "UAV-based automatic generation of high-resolution panorama at a construction site with a focus on preprocessing for image stitching." *Autom. Constr.* 84 (Aug): 70–80. https://doi.org/10.1016/j.autcon.2017.08.031.

Bircher, A., K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, and R. Siegwart. 2015. "Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics." In *Proc., 2015 IEEE Int. Conf. on Robotics and Automation*, 6423–6430. New York: IEEE.

Brown, D. C. 1971. "Close-range camera calibration." *Photogramm. Eng.* 86737 (8): 855–866.

Byrne, J., E. O'Keeffe, D. Lennon, and D. F. Laefer. 2017. "3D reconstructions using unstabilized video footage from an unmanned aerial vehicle." *J. Imaging* 3 (2): 15. https://doi.org/10.3390/jimaging3020015.

CBS Chicago. 2019. *Part of railroad bridge collapses near Chicago Skyway—CBS Chicago*. Chicago: CBS Chicago.

Cha, Y.-J., W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk. 2018. "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types." *Comput.-Aided Civ. Infrastruct. Eng.* 33 (9): 731–747. https://doi.org/10.1111/mice.12334.

Chen, Y., J. Zhang, and B.-C. Min. 2019. "Applications of BIM and UAV to construction safety." In *Proc., 7th CSCE Int. Construction Specialty Conf.*, 1–7. Montréal: Canadian Society for Civil Engineering.

Clayton, M., J. Kunz, and M. Fischer. 1998. *The Charrette test method*. Rep. No. TR120. Stanford, CA: Center for Integrated Facility Engineering.

Degol, J., J. Y. Lee, R. Kataria, D. Yuan, T. Bretl, and D. Hoiem. 2018. "FEATS: Synthetic feature tracks for structure from motion evaluation." In *Proc., 2018 Int. Conf. on 3D Vision, 3DV 2018*, 352–361. New York: IEEE.

Dorafshan, S., R. J. Thomas, and M. Maguire. 2018. "Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete." *Constr. Build. Mater.* 186 (Oct): 1031–1045. https://doi.org/10.1016/j.conbuildmat.2018.08.011.

Freimuth, H., and M. König. 2018. "Planning and executing construction inspections with unmanned aerial vehicles." *Autom. Constr.* 96 (Dec): 540–553. https://doi.org/10.1016/j.autcon.2018.10.016.

Girshick, R., J. Donahue, T. Darrell, and J. Malik. 2014. "Rich feature hierarchies for accurate object detection and semantic segmentation."

In Vol. 9 of *Proc., IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, 580–587. New York: IEEE Computer Society.

Golparvar-Fard, M., F. Peña-Mora, and S. Savarese. 2015. "Automated progress monitoring using unordered daily construction photographs and IFC-based building information models." *J. Comput. Civ. Eng.* 29 (1): 04014025. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000205.

Ham, Y., K. K. Han, J. J. Lin, and M. Golparvar-Fard. 2016. "Visual monitoring of civil infrastructure systems via camera-equipped unmanned aerial vehicles (UAVs): A review of related works." *Visualization Eng.* 4 (1): 1. https://doi.org/10.1186/s40327-015-0029-z.

Hoiem, D. 2018. "Maximize measurement accuracy with images overlaid on point clouds." Accessed May 13, 2020. https://medium.com/reconstruct-inc/maximize-measurement-accuracy-with-images-overlaid-on-point-clouds-dca828f4a539.

Hoiem, D., Y. Chodpathumwan, and Q. Dai. 2012. "Diagnosing error in object detectors." In *Proc., European Conf. Computer Vision*, 340–353. Berlin: Springer. https://doi.org/10.1007/978-3-642-33712-3_25.

Ibrahim, A., and M. Golparvar-Fard. 2019. "4D BIM based optimal flight planning for construction monitoring applications using camera-equipped UAVs." In *Proc., Computing in Civil Engineering 2019*, 217–224. Reston, VA: ASCE.

Ibrahim, A., M. Golparvar-Fard, T. Bretl, and K. El-Rayes. 2017a. "Model-driven visual data capture on construction sites: Method and metrics of success." In *Proc., Int. Workshop for Computing in Civil Engineering (IWCCE 2017)*, 109–116. Reston, VA: ASCE.

Ibrahim, A., D. Roberts, M. Golparvar-Fard, and T. Bretl. 2017b. "An interactive model-driven path planning and data capture system for camera-equipped aerial robots on construction sites." In *Proc., Int. Workshop for Computing in Civil Engineering (IWCCE 2017)*, 117–124. Reston, VA: ASCE.

Irizarry, J., M. Gheisari, and B. Walker. 2012. "Usability assessment of drone technology as safety inspection tools." *J. Inf. Technol. Constr.* 17 (12): 194–212.

Isailović, D., V. Stojanovic, M. Trapp, R. Richter, R. Hajdin, and J. Döllner. 2020. "Bridge damage: Detection, IFC-based semantic enrichment and visualization." *Autom. Constr.* 112 (Apr): 103088. https://doi.org/10.1016/j.autcon.2020.103088.

Jahanshahi, M. R., J. S. Kelly, S. F. Masri, and G. S. Sukhatme. 2009. "A survey and evaluation of promising approaches for automatic image-based defect detection of bridge structures." *Struct. Infrastruct. Eng.* 5 (6): 455–486. https://doi.org/10.1080/15732470801945930.

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.

Lin, J. J., and M. Golparvar-Fard. 2018. "Visual data and predictive analytics for proactive project controls on construction sites." In *Proc., Workshop of the European Group for Intelligent Computing in Engineering*, 412–430. Berlin: Springer.

Lin, J. J., K. K. Han, and M. Golparvar-Fard. 2015. "A framework for model-driven acquisition and analytics of visual data using UAVs for automated construction progress monitoring." In *Proc., Computing in Civil Engineering 2015*, 156–164. Reston, VA: ASCE.

Lin, T.-Y., P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. 2016. "Feature pyramid networks for object detection." In *Proc., IEEE Conf. on Computer Vision and Pattern Recognition*, 2117–2125. New York: IEEE.

Lindner, S., C. Garbe, and K. Mombaur. 2019. "Optimization based multi-view coverage path planning for autonomous structure from motion recordings." *IEEE Rob. Autom. Lett.* 4 (4): 3278–3285. https://doi.org/10.1109/LRA.2019.2926216.

Liu, K., and M. Golparvar-Fard. 2015. "Crowdsourcing construction activity analysis from jobsite video streams." *J. Constr. Eng. Manage.* 141 (11): 04015035. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001010.

Liu, Y., J. Yao, X. Lu, R. Xie, and L. Li. 2019. "DeepCrack: A deep hierarchical feature learning architecture for crack segmentation." *Neurocomputing* 338 (Apr): 139–153. https://doi.org/10.1016/j.neucom.2019.01.036.

Majeed, A., and S. Lee. 2019. "A new coverage flight path planning algorithm based on footprint sweep fitting for unmanned aerial vehicle navigation in urban environments." *Appl. Sci.* 9 (7): 1470. https://doi.org/10.3390/app9071470.

MDOT (Minnesota Department of Transportation). 2019. "Bridge inspection quality assurance." Accessed May 13, 2020. https://www.dot.state.mn.us/govrel/reports/2019/2019-bridge-iqa.pdf.

Morgenthal, G., N. Hallermann, J. Kersten, J. Taraben, P. Debus, M. Helmrich, and V. Rodehorst. 2019. "Framework for automated UAS-based structural condition assessment of bridges." *Autom. Constr.* 97 (Jan): 77–95. https://doi.org/10.1016/j.autcon.2018.10.006.

Mundt, M., S. Majumder, S. Murali, P. Panetsos, and V. Ramesh. 2019. "Meta-learning convolutional neural architectures for multi-target concrete defect classification with the concrete defect bridge image dataset." In Vol. 2019 of *Proc., IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*. New York: IEEE.

Nedjati, A., G. Izbirak, B. Vizvari, and J. Arkat. 2016. "Complete coverage path planning for a multi-UAV response system in post-earthquake assessment." *Robotics* 5 (4): 26. https://doi.org/10.3390/robotics5040026.

Nishikawa, T., J. Yoshida, T. Sugiyama, and Y. Fujino. 2012. "Concrete crack detection by multiple sequential image filtering." *Comput.-Aided Civ. Infrastruct. Eng.* 27 (1): 29–47. https://doi.org/10.1111/j.1467-8667.2011.00716.x.

Paine, D. P., and J. D. Kiser. 2012. *Aerial photography and image interpretation*. New York: Wiley.

Reason. 2020. "Policy study: Structurally deficient bridges." Accessed May 12, 2020. https://reason.org/policy-study/24th-annual-highway-report/structurally-deficient-bridges/.

Ren, S., K. He, R. Girshick, and J. Sun. 2017. "Faster R-CNN: Towards real-time object detection with region proposal networks." *IEEE Trans. Pattern Anal. Mach. Intell.* 39 (6): 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031.

Ryan, T. W., J. E. Mann, Z. M. Chill, and B. T. Ott. 2012. "Bridge inspector's reference manual (BIRM)." Accessed May 12, 2020. https://www.fhwa.dot.gov/bridge/nbis/pubs/nhi12049.pdf.

Sacks, R., et al. 2018. "SeeBridge as next generation bridge inspection: Overview, information delivery manual and model view definition." *Autom. Constr.* 90 (Jun): 134–145. https://doi.org/10.1016/j.autcon.2018.02.033.

Seo, J., L. Duque, and J. Wacker. 2018. "Drone-enabled bridge inspection methodology and application." *Autom. Constr.* 94 (Oct): 112–126. https://doi.org/10.1016/j.autcon.2018.06.006.

Siebert, S., and J. Teizer. 2014. "Mobile 3D mapping for surveying earthwork projects using an unmanned aerial vehicle (UAV) system." *Autom. Constr.* 41 (Feb): 1–14. https://doi.org/10.1016/j.autcon.2014.01.004.

Spencer, B. F., V. Hoskere, and Y. Narazaki. 2019. "Advances in computer vision-based civil infrastructure inspection and monitoring." *Engineering* 5 (2): 199–222. https://doi.org/10.1016/j.eng.2018.11.030.

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.

Tuttas, S., A. Braun, A. Borrmann, and U. Stilla. 2016. "Evaluation of acquisition strategies for image-based construction site monitoring." *ISPRS Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* 41 (Jun): 733–740. https://doi.org/10.5194/isprsarchives-XLI-B5-733-2016.

Washer, G., M. Nasrollahi, C. Applebury, R. Connor, A. Ciolko, R. Kogler, P. Fish, and D. Forsyth. 2014. *Proposed guideline for reliability-based bridge inspection practices*. Rep. No. 12-82 (01). Washington, DC: The National Academies Press.

Wells, J., and B. Lovelace. 2018. "Improving the quality of bridge inspections using unmanned aircraft systems (UAS)." Accessed July 21, 2020. http://mndot.gov/research/reports/2018/201826.pdf.

WTKR. 2019. "Concrete bridge railing collapses onto Tennessee interstate, injuring one person." Accessed May 12, 2020. https://wtkr.com/2019/04

/02/concrete-bridge-railing-collapses-onto-tennessee-interstate-injuring-one-person/.

Wu, C. 2014. “Critical configurations for radial distortion self-calibration.” In *Proc., IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, 25–32. New York: IEEE.

Wu, S., W. Sun, P. Long, H. Huang, D. Cohen-Or, M. Gong, O. Deussen, and B. Chen. 2014. “Quality-driven poisson-guided autoscanning.” *ACM Trans. Graph.* 33 (6): 1–12. https://doi.org/10.1145/2661229.2661242.

Xu, Y., Y. Bao, J. Chen, W. Zuo, and H. Li. 2019. “Surface fatigue crack identification in steel box girder of bridges by a deep fusion convolutional neural network based on consumer-grade camera images.” *Struct. Health Monit.* 18 (3): 653–674. https://doi.org/10.1177/1475921718764873.

Yang, J., M. W. Park, P. A. Vela, and M. Golparvar-Fard. 2015. “Construction performance monitoring via still images, time-lapse photos, and video streams: Now, tomorrow, and the future.” *Adv. Eng. Inf.* 29 (2): 211–224. https://doi.org/10.1016/j.aei.2015.01.011.