Bridge Inspection with Aerial Robots and Computer Vision: A Japanese National Initiative

Jacob J. Lin[a], **Amir Ibrahim**[a], **Shubham Sarwade**[a], **Mani Golparvar-Fard**[a]
Yasushi Nitta[b], **Hirokuni Moirkawa**[b], **and Yoshihiko Fukuchi**[c]

[a] University of Illinois at Urbana-Champaign, USA
[b] Ministry of Land, Transport, Infrastructure and Tourism, Japan
[c] Autodesk Inc., USA

jlin67@illinois.edu

Abstract -
Fast and accurate inspection of elevated structures is imperative for sustaining the increasing traffic flow on deteriorating bridges. Despsite their importance, today's inspection processes still require dedicated equipment, impact traffic flow, and expose inspection personnel to safety concerns. The advent of Unmanned Aerial Vehicles (UAVs) and the ability to position a camera close to elevated highway structures presents an opportunity to perform inspections quickly, safely and effectively. Towards this goal, we present an end-to-end system for robotic bridge inspection. Our system is structured around integrated methods to (a) create UAV flight missions; (b) evaluate accuracy and completeness of data collection plans; (c) generate 3D models of the structures of interest; (d) detect surface distresses in 3D; and (e) generate inspection reports in compliance with the requirements of highway agencies. We present results on validating each algorithm and the system as a whole. We also share lessons learned from an owner's perspective on deploying this system for bridge inspection in Japan particularly around procedures for documenting, communicating, and following up on bridge inspectors' recommendations.

Keywords -
Bridge Inspection; Deep Learning; Structure from Motion; Aerial Robots; Computer Vision

1 Introduction

The discovery of fractures in steel members of Kisogawa Ohashi Bridge and Honjo Ohashi Bridge in 2007, yet again raised alarms about the deterioration and fragile state of the infrastructure in Japan. By March 2019, deficient or functionally obsolete bridges constitute 9.6% of all the 700,000 road bridges in Japan [1]. Japan's geography creates additional risks associated with natural disasters. Thus, evaluating as-is conditions of Japan's bridges and infrastructure assets is more important than many other places around the world.

Today's bridge inspection practices are expensive and can be disruptive to ongoing traffic. Onsite documentations are also time-consuming and assessments can be inconsistent. This means additional data collection may be necessary in many cases, while resource allocation is always challenging. The advent of agile aerial platforms with the ability to carry digital cameras, along with a greater awareness of the technology has created an cost-effective alternative to conduct bridge inspection in a quick, safe and effective manner. Application of aerial platforms resolves issues of access and traffic disruptions. Also, cloud-based analysis lowers the need for onsite engineer visits, reducing cost associated with inspection. As such, the Ministry of Land, Infrastructure, Transport and Tourism (MLIT) in Japan launched an initiative in 2015 with the aim of leveraging advanced robots and artificial intelligence technologies to improve productivity in construction and infrastructure inspection tasks.

Because of the MLIT initiative as well as similar ones in the United States, a large body of work has emerged from the industry and academia which focuses on computer vision and visualization methods to process, analyze and share inspected drone data. Much of these efforts has focused on improving one step in the process, rarely offering an insight or recommendation on how various techniques can be applied in an integrated manner to streamline the data collection, analytics, and reporting in an end-to-end fashion. While there are promising computer vision methods such as 3D reconstruction, image classification, object detection and semantic segmentation that can been applied to bridge inspection processes, yet their adaptability as part of an end-to-end system has not been investigated.

To address the existing gaps in knowledge, we present an integrated end-to-end system for robotic bridge inspection consisting of five integrated methods: (a) creation of flight missions for data collection; (b) evaluation of flight missions based on requirements of inspection and visual quality of collected data; (c) 3D reconstruction of elevated structures; (d) automated detection and localization of surface distresses in 3D; and (e) report generation in compliance with owner requirements. We validate each underlying algorithm and the end-to-end system as a whole. We also share best practices on documentation, communication and following up on bridge inspection recommendations based on deploying our system on 30 bridge inspection projects in Japan and the United States.

2 Related Work

A large body of work in the literature has focused on robotic infrastructure inspection. In this section, we discuss the most recent methods and their gaps in knowledge:

2.1 Model-driven Data Capture using UAVs

Visual data collection with the UAVs has many applications including surveying, inspections, safety and progress monitoring. Based on whether models are used as *a priori*, these methods can be categorized into two classes: (1) Conventional methods that focus on ensuring minimum overlap between collected images to generate complete 3D point clouds; and (b) model-driven methods which use 3D CAD or BIM as *a priori* to simulate and evaluate the 3D visual coverage of structures and in turn reduce safety risks associated with the UAVs. Flight plans with waypoints sampled at a safe offset distance from the structure [2] and the utilization of 4D BIM for path planning in construction monitoring use-cases [3] are examples of these model-driven approaches. Our method builds on these model-driven path planning methods, but it also enables verifying and planning for requirements such as preserving line-of-sight to UAVs, or orthogonality of camera viewpoints against the structure during data capture.

2.2 Image-based 3D reconstruction

Structure from Motion (SfM) and Simultaneous Localization and Mapping (SLAM) algorithms have been widely used as 3D reconstruction techniques to generate high fidelity 3D point cloud and mesh models using images or videos collected with aerial robots. These techniques allow for better organization of collected data by identifying locations and viewpoints of images relative to 3D models. While these techniques have matured over the past decade, nevertheless they can still result in *incomplete* and *inaccurate* models. To address these issues, recent efforts such as [4] propose simulators or empirical metrics to examine quality of data collection and their impact on 3D reconstruction. Nonetheless, these studies have been validated in contexts which are not relevant to data collected from varying heights and viewpoints relative to elevated structures. Tools that can generate elevation or under-deck orthoviews are also required for inspection reports, however current 3D reconstruction pipelines do not produce such deliverables.

2.3 Metrics for visual quality assurance

The percent overlap between collected images, the resolution of bridge elements in each image, and the quality of 3D reconstruction are three key parameters that influence the accuracy of automated condition assessment system. Model-driven flight paths have been deployed to improve the quality of collected visual data [2] but these methods lack quality assurance capabilities. [5, 6] show parameters such as proximity of camera to the bridge elements and orientations of the collected images, which can impact accuracy of defect detection and appearance-based classification of structural components. To evaluate accuracy of 3D reconstruction in simulation environments, metrics such as (a) visibility per element and (b) redundancy in visibility have been introduced by [7]. Metrics such as Ground Sampling Distance (GSD) have also been employed to guarantee resolution of 3D reconstruction before image capture is conducted [8]. In this study, we build upon previously studied visual quality assurance metrics and investigate new metrics focusing on defect detection such as perpendicularity of image viewpoints while achieving a specific GSD per inspection target.

2.4 Defect Recognition, Severity Assessment and Mapping

Computer vision methods can be used on site images to detect and classify surface defects such as crack, spalling, exposed rebar, efflorescence and corrosion. Particularly, deep learning-based approaches have shown promising results in the recent years. Deep Convolutional Neural Network (CNN) architectures such as VGG, AlexNet, ResNet and MetaQNN for single and multi-class defect classification [8] have resulted in exceptional results (e.g., VGG architecture achieved 70.61% for multi-class defect classification) but these method do not predict defect regions and at best, they only focus on a Single-class defect detection. [9, 10, 11, 12] in among the few works that has investigated feasibility of an end-to-end framework for bridge inspection, however no previous work has considered multi-class defect detection and localization within a 3D context. Localizing defect in 2D and 3D is essential for assessing seriousness of defects and to prioritize maintenance operations.

2.5 Virtual inspection and report generation

There is plenty of work in the literature on generating 3D point clouds of bridges and detection of multiple type of defects; nevertheless there are still many components missing to enable an end-to-end robotic inspection; i.e., (a) Comparative views of point clouds and images over time, (b) interactive orthoviews generated automatically, and (c) ability to view images overlaid on 3D point clouds through an interactive interface, similar to ones implemented for project controls [13], are much needed. 3D measurements and annotations could also be enabled through a visual interface for defect severity assessment purposes. These tools and systems can automate and significantly improve how owner-compliance reports are generated. The work in

[9, 14] demonstrate methods to generate reports utilizing BIM of bridges. However, creating BIM from point clouds is not a trivial task and is often cost-prohibitive for large number of bridges in need for inspection.

In the following, we present an end-to-end solution that addresses these gaps in knowledge. Our solution has been deployed by the MLIT in Japan and other bridge owners and operators in the United States.

3 Method

We propose a new end-to-end system for robotic inspection (see Figure 1). Each step in our system's is discussed in the following sections:

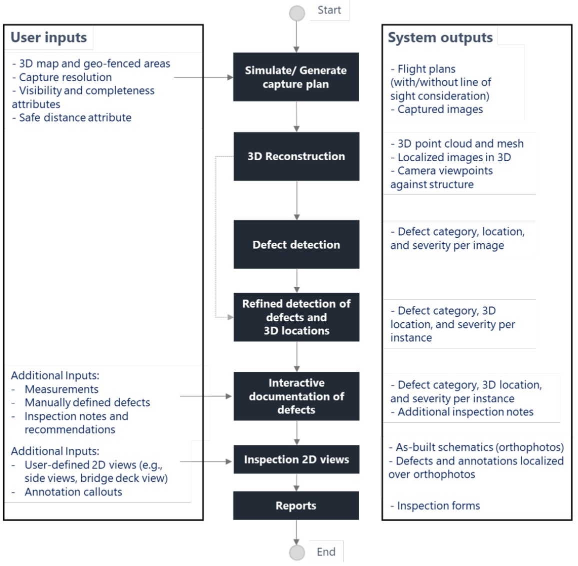

Start
User inputs
- 3D map and geo-fenced areas
- Capture resolution
- Visibility and completeness attributes
- Safe distance attribute
Additional Inputs:
- Measurements
- Manually defined defects
- Inspection notes and recommendations
Additional Inputs:
- User-defined 2D views (e.g., side views, bridge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

Figure 1. A new end-to-end system of bridge inspection, with relevant user inputs and expected outputs.

3.1 Model-driven Visual Data Inspection

Automatic visual data collection is the first step in our end-to-end bridge inspection system. To achieve this, we developed a web-based virtual environment with client-server architecture for fast and easy access to aerial data collection missions and automatic 3D flight plans generation. The user sets the inspection region with a 2D satellite map and specified the data collection parameters to generate the 3D flight plan. The parameters considered the requirements of the data collection, such as altitudes of bridge deck underside and topside, waypoints offset distance from the structure, drone and camera settings, the field of view to keep line-of-sight during flight execution and the required overlap between image frames. The generated flight plan is sliced into missions to account for line-of-sight requirements and the drone's battery life.

The 3D virtual environment is used to render the flight mission alongside BIM/reality models of the bridge for evaluation and communication. This allows adjustments to the flight mission parameters for improving visual coverage and safety. Execution is supported via an iOS mobile application adapted from a previous study [3].

3.2 Quality Evaluation and Feedback on Data Collection Plan

We developed four metrics to define the visual quality of bridge inspection data with regards to the requirements of bridge inspection data: (1) visual coverage completeness of inspection region; (2) redundancy of observations of bridge elements across images needed for accurate 3D reconstruction; (3) target pixel resolution or GSD needed for precise defect detection; and (4) canonical orientation of camera against bridge elements for minimal distortion and required pixel resolution. Redundancy in observations is used for cross-referencing detection across images, which improves detection reliability.

A priori 3D model needs to be generated to perform the flight mission simulation. Without previous capture, we perform a 3D reconstruction from synthetic images extracted from 3D terrain map platforms to create *a priori* model. Using this model, the simulation can estimate: (1) the visibility of each element in each data frame; (2) redundant visibility of elements across all data frames; (3) the average pixel resolution of each element in every frame; and (4) the relative orientation of each element to each frame. The elements are rendered in the virtual environment by a simulated camera with the parameters of the actual camera used for data collection. In the evaluation of visibility and redundant visibility of elements, an element ID color-coding scheme is implemented, and an element's visibility is determined by whether the number of back-projected pixels of the corresponding color exceeds a user-defined threshold in a frame.

A similar approach is applied to estimate pixel resolution of an element in each frame with color-coding based on the depth of the element. The relative orientation of each element to each data frame is estimated in the final simulation. We utilize the surface normal shader to render elements' surface normals as RGB colors. The results of the simulation are provided to the user visually for flight plan adjustment.

3.3 Image-based 3D modeling

We developed different 3D reconstruction methods to address the issues that appear more frequently for bridges. These issues include incompleteness, misregistration, and curvature (drift). The 3D reconstruction pipeline follows an extension of previous studies [15], which include Structure from Motion (SfM) algorithm, patch-based multiview reconstruction and mesh modeling steps. However, point cloud misregistration often occurs in bridges due to the repetitive bridge structures in typical SfM process. To resolve the ambiguity in the feature matching process, we introduced a feature clustering mechanism that utilizes GPS coordinates extracted from image metadata to reduce the misregistration. To mitigate point clouds curvature caused by the radial distortion ambiguity, we implemented the camera model [16] that estimates the intrinsic matrix including the distortion coefficient. In the final mesh modeling process, a *digital depth model* is generated to allow users to create orthographic views from any user-defined view necessary for inspection (e.g., under the deck view or side view). The *digital depth model* is presented in the form of a raster image with pixels representing depth value arranged in rows and columns. The images are stitched together along the seam lines with the correction of radial and depth distortion utilizing the digital depth model.

3.4 Damage Detection, Localization and Mapping

We use the Faster-RCNN architecture [17] and present a new algorithm to detect, localize, and determine the 3D spatial mapping of each type of damage. The detection output is a predicted label with its bounding box and score. In our new method, these outputs are projected into the 3D point clouds and back-projected to all related images to analyze intersected area and then fed into the 3D spatial mapping algorithm to determine final labels for each fragment in the 3D point cloud (Fig. 2).

Faster-RCNN includes a Region Proposal Network (RPN) and an object detection network Fast-RCNN, which shares a backbone CNN to form a unified network. The backbone CNN architecture is initialized with a pretrained network (e.g., VGG, Alexnet, ResNet) and fed with input images to extract feature maps. The RPN utilizes the feature maps to generate region proposals by sliding a network with fully-connected $n \times n$ spatial windows, which are output as a lower-dimensional (512 for VGG, 1024 for Resnet) vector and fed into box-regression and box-classification layers. Translation and scale-invariant outputs are achieved using k anchor boxes set as a grid, and k regions are predicted simultaneously. Anchors are associated with three scales and three aspect rations and are generated at the center of the sliding window. Loss function for training of RPN is computed using eq. 1 with a binary class label assigned to each anchor box, where positive labels have Intersection over Union (IoU) over 70% with ground truth. The Feature Pyramid Network (FPN) utilizes deep CNN pyramidal feature hierarchy to build a semantically rich feature pyramid from low to high levels to both learn local and spatial features.

$$\begin{aligned} L\{d_i\}, \{t_i\} = \frac{1}{N_{cls}} \; L_{cls} d_i, d_i^* \\ +\lambda \frac{1}{N_{reg}} \; d_i^* L_{reg} t_i, t_i^* \end{aligned} \tag{1}$$

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

3.5 3D Spatial Mapping

The 3D spatial mapping developed in this study helps eliminate the annotation and prediction inconsistencies. We utilize computer graphics techniques and camera projection matrices estimated in 3D reconstruction to localize the same damages. All defect bounding boxes from each data frame are projected to 3D space and back-projected to any images that contain the corresponding region. Ideally, all such back-projected boxes should perfectly align with the same defect, but due to errors in projection and defect detection, that is not the case. To address such errors and increase detection precision, we present an algorithm to identify the most probable back-projected boxes for each defect.

Three user-defined parameter thresholds are used for this purpose. An IoU threshold is used to identify bounding box clusters within a user-defined 3D spatial distance to limit reprojection error. Clustered bounding boxes over a threshold count are then processed by performing a greedy Non-maximum suppression (NMS) to choose the most probable box. This approach is first applied to the ground-truth bounding boxes obtained from annotations and then to the predicted detection bounding boxes. Ground-truth bounding box clusters with lower box count than the threshold are kept, and detection box clusters with lower box count than the threshold and lower IoU than the threshold are removed. Similar approach is also applied to the detection process.

3.6 Interactive Web-based Viewers for Inspection

The interactive web-based viewer is extended from [13] 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 for efficient rendering. The system automatically loads the points inside the user's field of view via a lazy scheme to reduce the loading speed.

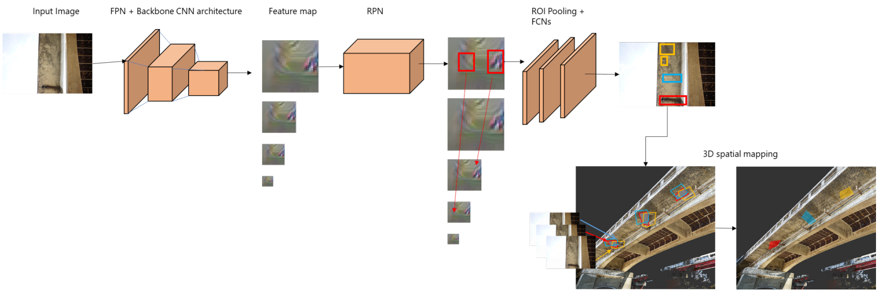


Figure 2. The network architecture. Feature maps extracted from backbone Resnet50, FPN are input to RPN. Bounding boxes with labels and scores as output from classifier mapped in 3D environment is re-injected to finalize prediction.

In addition to visual data collection planning and capture quality evaluation, the virtual walkthrough provides the necessary measurement capabilities for inspection. The image-based measurement tools utilize raycasting to find the closest point and backproject the point using the camera intrinsic, extrinsic to the image. We improved the raycasting accuracy by implementing a cylindrical ray with a radius r to get accurate results when the point cloud is sparser with varying depth. The web-based viewer also provides a non-linear transformation from 3D point clouds to 2D orthophoto using feature matches, estimated depth, and gravity direction from 3D reconstruction. A 4D (3D+time) point cloud timeline is formed chronologically by the capture date with the images, point clouds. Side-by-side or overlaid image comparison interface over time is also enabled by finding the closest image and transform it into the same coordinates using McMillan & Bishop warping method.

Per-pixel distance values are used to warp pixels to their correct location for the current camera position. Side-by-side or overlay views enable observation of defect progression over time and severity assessment for the inspectors.

3.7 Report Generation

Bridge inspection results are summarized in necessary reporting forms that (a) present schematic views of various bridge components and relative location and severity of defects; and (b) defect images as well as their associated inspection info. Our process leverages defect information and associated attributes already mapped in the 3D point cloud to speed up the generation of such reports. An image with a pin is automatically attached to the reports with associated information. These are exported as excel sheets with hyperlinks to the 3D location in the viewer. The related information includes properties such as severity, notes, inspector's name, and GPS location. Inspectors can easily access all the information that needs to be used to assess the defect from these excel sheets that follow owner reporting guidelines (fig. 3).

Our surface mapping tool creates 2D orthophoto from different viewpoints, facilitates the schematic drawing creation with more information than the traditional. The 2D orthophoto with the analysis of the inspected structure provides the overall status of the bridge in a short glance.

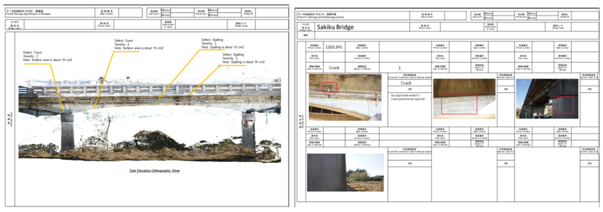

Defect: Crack
Severity: 3
Note: Surface area is about 10 cm2
Defect: Crack
Severity: 4
Note: Surface area is about 10 cm2
Defect: Spalling
Severity: 2
Note: Spalling is about 10 cm2
Defect: Spalling
Severity: 3
Note: Spalling is about 10 cm2
East Elevation Orthographic View
Sakiku Bridge
1203.JPG
Crack
1
Crack
Our approved vendor's crack sealant to be applied.

Figure 3. Examples of inspection forms (left) that require bridge elevation view to illustrate defect location and (right) detailed description of each defect with additional notes

4 Technical Validation

We validated the developed end-to-end system through 30 bridge inspection projects in Japan and the United States. In this section we present a couple of these projects for the purpose of demonstrating our results, mainly we show the qualitative and quantitative findings of deploying the new system to two Japanese bridge inspection projects. For each of the later projects, we focused on automatic defects detection for one of the bridge's spans covering the underside and lateral sides of the inspection region. Accordingly, detailed results for each stage of the inspection workflow are presented as follows.

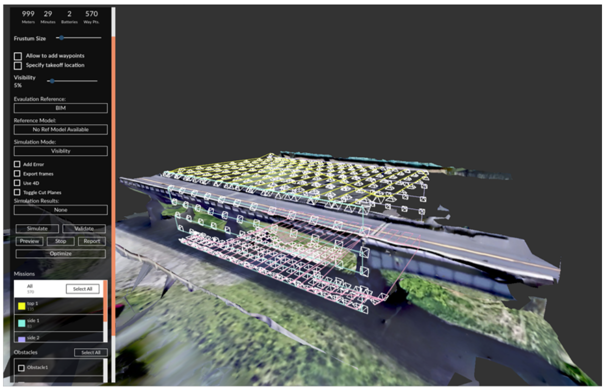
4.1 Data Collection

The first step of bridge inspection system is using the developed web-interface to select the inspection region for each of the bridges, such region is selected by navigating a 2D map to the project location and using satellite images to define the boundary of the region. Next we set the data collection parameters as discussed in the method section, for instance, we provide an example of the parameters set for one of the bridges in Table 1.

Table 1. Flight plan parameters

Parameter	Value
Top offset	6 m
Bottom offset	3 m
Sides offset	4 m
Images overlap	80%
Drone battery life	20 mins
Drone speed	25 kph
Line-of-sight FOV	90°
Aerial platform	DJI Phantom4

The developed flight planning interface automatically generates a 3D flight plan with missions covering the top, lateral and under sides of the bride structure. The total flight distance and flight-time for the plan are 2,200 meters and 79 minutes respectively. We provide a visual representation for the later flight plan in Figure 4 which shows the 3D missions along with as-is point cloud model of the bridge.



999 Meters 29 Minutes 2 Batteries 570 Way Pts.

Visibility 5%

Evaluation Reference: BIM

Reference Model: No Ref Model Available

Simulation Mode: Visiblity

Simulation Results: None

Figure 4. Generated 3D flight plan with top, bottom and one of the side missions visualized in unique colors.

4.2 Visual Quality Evaluation

Feedback for the visual quality of the data collection plan is presented in Figure 5 which provides an example of the visual feedback for visibility, resolution and orientation evaluations. For such task, the bridge mesh was divided into fine fragments that are color-coded to show the visual quality value for each fragment.

The visual evaluation criteria are set according to the requirements for bridge inspection, the later criteria and results of the evaluation feedback are shown in Table 2. 13 fragments found to be outside the acceptable criteria which happen to be on the ground and outside the inspection scope. In future iterations of the tool, we plan to remove these elements from the resolution evaluation process, thus, revisions and further modifications to the flight plan were not needed.

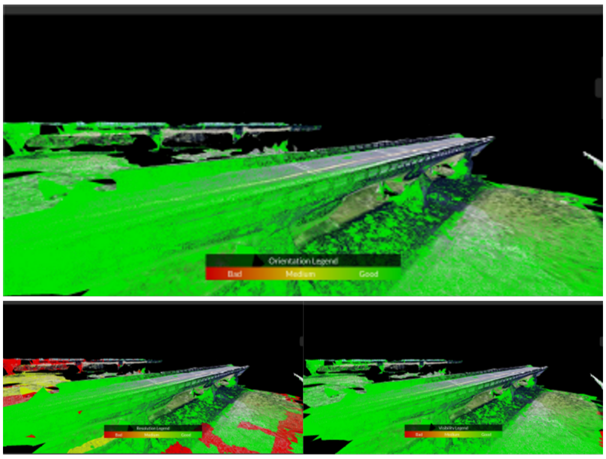


Figure 5. Color-coded visual feedback for data collection plan. (top) visibility evaluation, (bottom-left) resolution evaluation, and (bottom-right) orientation evaluation. The red region, being outside scope of evaluation, is ignored.

Table 2. Flight plan evaluation results

Metric	Criteria	# of elements	% of elements
Visibility	Not met (< 3 observations)	13	16.88%
	Acceptable (3 − 8 observations)	3	3.90%
	Satisfactory (≥ 8 observations)	61	79.22%
Resolution	Not met (> 0.01 m/pix)	11	14.29%
	Acceptable (0.005 − 0.01 m/pix)	0	0.00%
	Satisfactory (≤ 0.005 m/pix)	66	85.71%
Orientation	Not met (> 45°)	1	1.30%
	Acceptable (15 − 45°)	2	2.60%
	Satisfactory (≤ 15°)	74	96.10%

4.3 Damage Localization, Detection and Mapping

4.3.1 Dataset preparation and annotations

The image dataset was created from inspection images captured for the two bridges, the defects in these images were labelled into five common defect classes, namely: crack, spalling, efflorescence, corrosion stains and exposed rebar. For the annotation process, we extended Computer Vision Annotation Tool (CVAT) by integrating the collected dataset and specifying new labels. The inspection images were annotated by experts and validated by two reviewers per each annotation job following the quality assurance/control process in [18]. A total of 30 engineering experts served as annotators and reviewers in this study. Attributes of the resulted dataset are detailed in Table 3. For defect detection, the dataset was split into training and testing datasets with the 80/20 convention. A total of 653 images with 14,302 defect instances were used for training and 89 images with 4804 object instances for testing.

Table 3. 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

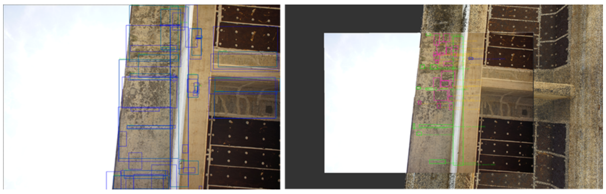


Figure 6. (Left): Results of the localization bounding-boxes for defects in green with projected boxes from other frames -generated through 3D mapping process- colored in blue; (Right) the final localized bounding boxes -after 3D mapping process- overlaid on registered image in 3D point cloud context.

4.3.2 Experimental Results and Discussions

FasterRCNN architecture was implemented using pytorch framework and was fine-tuned with MS-COCO object detection model to expedite initial training with six object classes including defects mentioned earlier and the background. The model was trained from scratch for each layer with the Stochastic Gradient Descent (SGD) optimizer. The learning rate was set using cosine annealing schedule cycling between 1e-5 to 1e-2 in every 10 epochs to avoid local minimum and overfitting. Results of the detection are shown in Fig. 6.

Average Precision (AP) is then calculated by dividing count of true positive detections over all positive detections and is used to evaluate the detection results. The average precision values from the detection are summarized in Table 3. NMS removes some detection boxes not meeting threshold earlier defined leading to reduced total detections. NMS parameters were fine-tuned to balance trade-off for best result.

3D spatial mapping was found to increase the AP for crack and efflorescence by more than 4% with improvements in all defect types (Table.3). Detection results are also continuously improved through user inputs provided through the interactive web viewer such as verification of correct defects and adding missing detections for documentation and reporting by expanding dataset and detection model retraining.

4.4 Web-based Viewer and Inspection Reports

We adopted Charette test method [19] to evaluate the user's feedback about the web-viewer and report generation process. The validation works by evaluating *effectiveness*, *repeatability*, and *reliability* of our method, which measure productivity improvement, performance of system under different inspection conditions and users. These were measured qualitatively through interviews and observations from MLIT practitioners interacting with the system. We have provided training and guidelines for bridge inspectors, transportation professionals and civil engineers to perform virtual inspection tasks. Users expressed through the training that time required for defect localization and creating reports using corresponding images was significantly reduced. Users also expressed that the output generated can be easily transformed to actionable format which is vital for effective bridge maintenance.

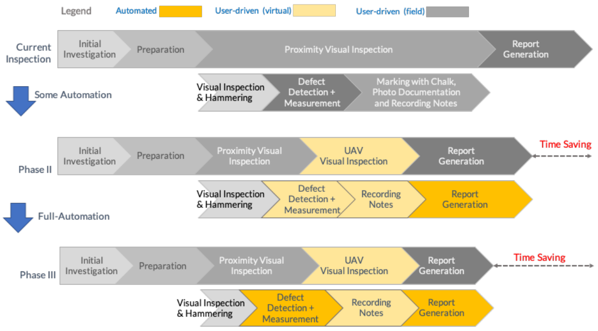


Figure 7. The MLIT in Japan has outlined a three-phase path towards a fully automated bridge inspection process, where more automation is introduced one step at a time. Currently, Phase II is in full testing mode with the solution presented in this paper.

5 Discussion - Gradual Transition to a Fully Automated System

The components of the presented system were deployed on bridge inspection projects in Japan and the United States. Based on discussion and interviews with bridge inspectors, Japan MLIT has decided to engage in a gradual transition from their current workflow to full automation, due to the many components, bridge owners, and engineering personnel involved. Fig. 7 shows MLIT's plan and perceived benefit from this transition.

In the current workflow, inspectors spend significant time measuring, capturing and organizing damages and related information. Phase II will involve automating data collection and report generation processes consisting of the robotic solutions that collect and process the data in the web-based system. The system will also actively improve the automatic defect detection through (a) user input of correcting and adding missing detections, and (b) retraining of detection model on a timely basis over more massive ground truth datasets. Phase III will introduce automated pre-analysis of bridge condition, 3D reconstruction, defect detection, and report generation. Analyzed data will be used for verification of condition and to perform physical examinations on the field. Full adaptation of automation by Phase III is expected to cause substantial time savings (30% based on Charettes) and cost-savings in the inspection process. A reliable defect detection component is an essential step towards Phase III.

We have shown in this study that the data collection process improves the visual quality of collected data to meet MLIT inspection and report generation requirements. Future work will focus on improvements of the flight plan optimization by eliminating user manual inputs, the evaluation process by removing simulated fragments that are not in the inspection scope, the detection model by phased roll-outs of improved models, and expansion of ground-truth dataset with a generalized distribution of defect shapes, colors and incorporating 3D geometry information in the detection process.

6 Conclusion

We presented an end-to-end system to automate robotic bridge inspection process with integration of flight mission generation, assessment of visual quality of collected data, ensuring accuracy and completeness of reconstructed reality model, 3D reconstruction for elevated structures, automated detection and localization of damages and report generation. Our proposed system shows the connections between components and how robotic bridge inspection solutions can be streamlined systematically and gradually. We introduced the phase-based implementation of detection models that enables continuous improvement and active learning over time through dataset expansion and user input. Our holistic approach through the proposed end-to-end system connects the dots between the components of robotic bridge inspection for the MLIT in Japan. We plan to implement this system across 100s of bridges in the coming year.

References

[1] MLIT. Annual Road Maintenance Report, 2019. URL `https://www.mlit.go.jp/road/sisaku/yobohozen/pdf/h30/1-2.pdf`.

[2] Henk Freimuth and Markus König. Planning and executing construction inspections with unmanned aerial vehicles. *Automation in Construction*, 96:540–553, 12 2018. ISSN 09265805. doi:10.1016/j.autcon.2018.10.016.

[3] Amir Ibrahim, Dominic Roberts, Mani Golparvar-Fard, and Timothy Bretl. An Interactive Model-Driven Path Planning and Data Capture System for Camera-Equipped Aerial Robots on Construction Sites. *International Workshop for Computing in Civil Engineering (IWCCE 2017)*, pages 117–124, 2017.

[4] Yuting Chen, Jiansong Zhang, and Byung-Cheol Min. Applications of BIM and UAV to Construction Safety. In *7th CSCE International Construction Specialty Conference, Laval, QC, Canada*, pages 1–7, 2019.

[5] R. S. Adhikari, O. Moselhi, and A. Bagchi. Image-based retrieval of concrete crack properties for bridge inspection. *Automation in Construction*, 39:180–194, 2014. ISSN 09265805. doi:10.1016/j.autcon.2013.06.011. URL `http://dx.doi.org/10.1016/j.autcon.2013.06.011`.

[6] J.J. Jacob J. Lin, Kevin K.K. Han, and Mani Golparvar-Fard. A Framework for Model-Driven Acquisition and Analytics of Visual Data Using UAVs for Automated Construction Progress Monitoring. In *Computing in Civil Engineering 2015*, pages 156–164, 2015. ISBN 978-0-7844-0794-3. doi:doi:10.1061/9780784407943.

[7] Amir Ibrahim and Mani Golparvar-Fard. 4D BIM Based Optimal Flight Planning for Construction Monitoring Applications Using Camera-Equipped UAVs. In *Computing in Civil Engineering 2019*, pages 217–224, 2019. doi:10.1061/9780784482438.028.

[8] Yahui Liu, Jian Yao, Xiaohu Lu, Renping Xie, and Li Li. DeepCrack: A deep hierarchical feature learning architecture for crack segmentation. *Neurocomputing*, 338:139–153, 4 2019. ISSN 18728286. doi:10.1016/j.neucom.2019.01.036.

[9] Guido Morgenthal, Norman Hallermann, Jens Kersten, Jakob Taraben, Paul Debus, Marcel Helmrich, and Volker Rodehorst. Framework for automated UAS-based structural condition assessment of bridges. *Automation in Construction*, 97:77–95, 1 2019. ISSN 09265805. doi:10.1016/j.autcon.2018.10.006.

[10] Mohamad Alipour, Devin K. Harris, and Gregory R. Miller. Robust Pixel-Level Crack Detection Using Deep Fully Convolutional Neural Networks. *Journal of Computing in Civil Engineering*, 33(6):4019040, 11 2019. doi:10.1061/(ASCE)CP.1943-5487.0000854. URL `https://doi.org/10.1061/(ASCE)CP.1943-5487.0000854`.

[11] Allen Zhang, Kelvin CP Wang, Yue Fei, Yang Liu, Siyu Tao, Cheng Chen, Joshua Q. Li, and Baoxian Li. Deep Learning–Based Fully Automated Pavement Crack Detection on 3D Asphalt Surfaces with an Improved CrackNet. *Journal of Computing in Civil Engineering*, 32(5):4018041, 9 2018. doi:10.1061/(ASCE)CP.1943-5487.0000775. URL `https://doi.org/10.1061/(ASCE)CP.1943-5487.0000775`.

[12] Young-Jin Cha, Wooram Choi, and Oral Büyüköztürk. Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks. *Computer-Aided Civil and Infrastructure Engineering*, 32(5):361–378, 5 2017. ISSN 1093-9687. doi:10.1111/mice.12263. URL `https://doi.org/10.1111/mice.12263`.

[13] Jacob J. Lin and Mani Golparvar-Fard. Visual data and predictive analytics for proactive project controls on construction sites. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 10863 LNCS, pages 412–430. Springer Verlag, 6 2018. ISBN 9783319916347. doi:10.1007/978-3-319-91635-4_21.

[14] Dušan Isailović, Vladeta Stojanovic, Matthias Trapp, Rico Richter, Rade Hajdin, and Jürgen Döllner. Bridge damage: Detection, IFC-based semantic enrichment and visualization. *Automation in Construction*, 112:103088, 4 2020. ISSN 09265805. doi:10.1016/j.autcon.2020.103088.

[15] Mani Golparvar-Fard, Feniosky Peña-Mora, and Silvio Savarese. Automated Progress Monitoring Using Unordered Daily Construction Photographs and IFC-Based Building Information Models. *Journal of Computing in Civil Engineering*, 29(1):04014025, 1 2015. ISSN 0887-3801. doi:10.1061/(ASCE)CP.1943-5487.0000205. URL `http://ascelibrary.org/doi/10.1061/%28ASCE%29CP.1943-5487.0000205`.

[16] Duane C. Brown and Duane C. Brown. Close-range camera calibration. *PHOTOGRAMMETRIC ENGINEERING*, 37(8):855–866, 1971. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.14.6358`.

[17] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 6 2017. ISSN 01628828. doi:10.1109/TPAMI.2016.2577031.

[18] Kaijian Liu and Mani Golparvar-Fard. Crowdsourcing Construction Activity Analysis from Jobsite Video Streams. *Journal of Construction Engineering and Management*, 141(11):04015035, 11 2015. ISSN 0733-9364. doi:10.1061/(ASCE)CO.1943-7862.0001010. URL `http://ascelibrary.org/doi/10.1061/%28ASCE%29CO.1943-7862.0001010`.

[19] Mark Clayton, John Kunz, and Martin Fischer. The charrette test method. Technical report, 1998.