IDEA

Innovations Deserving
Exploratory Analysis Programs

NCHRP IDEA Program

Fatigue Crack Inspection Using Computer Vision and Augmented Reality

Final Report for
NCHRP IDEA Project 223

Prepared by:
Jian Li, Caroline Bennett, William Collins
The University of Kansas

Fernando Moreu
University of New Mexico

November 2022

TRB TRANSPORTATION RESEARCH BOARD
The National Academies of
SCIENCES · ENGINEERING · MEDICINE

Innovations Deserving Exploratory Analysis (IDEA) Programs Managed by the Transportation Research Board

This IDEA project was funded by the NCHRP IDEA Program.

The TRB currently manages the following three IDEA programs:

- The NCHRP IDEA Program, which focuses on advances in the design, construction, and maintenance of highway systems, is funded by American Association of State Highway and Transportation Officials (AASHTO) as part of the National Cooperative Highway Research Program (NCHRP).
- The Safety IDEA Program currently focuses on innovative approaches for improving railroad safety or performance. The program is currently funded by the Federal Railroad Administration (FRA). The program was previously jointly funded by the Federal Motor Carrier Safety Administration (FMCSA) and the FRA.
- The Transit IDEA Program, which supports development and testing of innovative concepts and methods for advancing transit practice, is funded by the Federal Transit Administration (FTA) as part of the Transit Cooperative Research Program (TCRP).

Management of the three IDEA programs is coordinated to promote the development and testing of innovative concepts, methods, and technologies.

For information on the IDEA programs, check the IDEA website (www.trb.org/idea). For questions, contact the IDEA programs office by telephone at (202) 334-3310.

IDEA Programs
Transportation Research Board
500 Fifth Street, NW
Washington, DC 20001

The project that is the subject of this contractor-authored report was a part of the Innovations Deserving Exploratory Analysis (IDEA) Programs, which are managed by the Transportation Research Board (TRB) with the approval of the National Academies of Sciences, Engineering, and Medicine. The members of the oversight committee that monitored the project and reviewed the report were chosen for their special competencies and with regard for appropriate balance. The views expressed in this report are those of the contractor who conducted the investigation documented in this report and do not necessarily reflect those of the Transportation Research Board; the National Academies of Sciences, Engineering, and Medicine; or the sponsors of the IDEA Programs.

The Transportation Research Board; the National Academies of Sciences, Engineering, and Medicine; and the organizations that sponsor the IDEA Programs do not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the object of the investigation.

Fatigue Crack Inspection Using Computer Vision and Augmented Reality

NCHRP IDEA Program Final Report

IDEA Project NCHRP-IDEA 223

Prepared for

The NCHRP IDEA Program
Transportation Research Board
National Academies of Sciences, Engineering, and Medicine

by

Jian Li, Principal Investigator

Caroline Bennett, Co-Principal Investigator

William Collins, Co-Principal Investigator

The University of Kansas

Lawrence, KS

Fernando Moreu, Co-Principal Investigator

University of New Mexico

Albuquerque, NM

November 2022

ACKNOWLEDGEMENT

The research reported herein was performed under NCHRP IDEA 20-30/IDEA 223 Project by the Department of Civil, Environmental and Architectural Engineering at the University of Kansas (KU), in collaboration with the Department of Civil, Construction and Environmental Engineering at the University of New Mexico (UNM). Dr. Jian Li of KU was the principal investigator. The other authors are Dr. Caroline Bennett and Dr. William Collins, both of KU, and Dr. Fernando Moreu of UNM.

Cavin Reed of Kansas DOT, Loren (LJ) Dickens of HNTB, Jason Schneider of Collins Engineers, Inc., and Eric Martinez of Los Alamos County of New Mexico served as the industry and government collaborators of this project. Their collaboration and support are greatly appreciated. Dr. Farhad Ansari of University of Illinois at Chicago served as the IDEA advisor and lead of the advisory panel. Other members of the advisory panel of this project include Dr. Sreenivas Alampalli of Prospect Solutions, LLC, Dr. Soundar Balakumaran of Virginia DOT, Dr. Mart Hurt of Kansas DOT, Dr. Christopher Peterson of Air Force Research Lab, and Anne Rearick of Indiana DOT. The comments and valuable input from the advisory panel are greatly appreciated. The research team would also like to thank Dr. Inam Jawed, Senior Program Officer of NCHRP IDEA program, for offering invaluable guidance, assistance and support during the course of this project.

Graduate students Rushil Mojidra of KU and Ali Amohammadkhorasan of UNM made significant contributions to this research. The research team also gratefully acknowledges Don Whisler, Jim Leaden, John Culbertson, Kevin Walker, and Justin Shaw, all of Kansas DOT, for providing the extensive field support.

EXPERT REVIEW PANEL
Farhad Ansari, *University of Illinois at Chicago*
Soundar Balakumaran, *Virginia DOT*
Mark Hurt, *Kansas DOT*
Anne Rearick, *Indian DOT*
Christopher Petersen, *Air Force Research Laboratory*
Sreenivas Alampalli, *Prospect Solutions, LLC*

TABLE OF CONTENTS

Acknowledgement .... **i**

Executive Summary **1**

IDEA Product **3**

Concept and Innovation **4**

Investigation **6**

Development of computer vision algorithms for fatigue crack detection **6**

Displacement-based Fatigue Crack Detection Algorithm 8

Distance-based Fatigue Crack Detection Algorithm 25

Development of Augmented Reality Software for human-centered fatigue crack inspections **37**

Laboratory and Field Implementations **46**

Laboratory Implementation 46

Field Implementation 50

Plans for Implementation **57**

Conclusions **58**

References **59**

Appendix A: Research Results **63**

LIST OF FIGURES

Figure 1 Fatigue crack inspection using computer vision and AR....4
Figure 2 Overview of the fatigue crack detection method based on a moving camera 9
Figure 3 Illustration of geometric transformation for GMC 10
Figure 4 Web-gap region of a bridge girder (a) with three different planes highlighted (b) 10
Figure 5 Bundled camera paths (a) video frames divided into grid cells (b) camera paths of grid cells (x-axis translation over time) 11
Figure 6 Flowchart of the proposed extended bundled camera paths method for GMC of 3D videos....12
Figure 7 Illustration of the proposed crack detection algorithm using a cracked steel plate under fatigue loading....15
Figure 8 Test setup for 2D videos: (a) C(T) specimen; and (b) loading scheme 16
Figure 9 Schematic of the test setup including the camera position 16
Figure 10 Bridge girder test setup for 3D videos (unit: mm)....18
Figure 11 Out-of-plane fatigue crack and video recording using the HL2 headset 18
Figure 12 The initial frame of the 2D video with the selected ROI (a) and all feature points detected by the Shi-Tomasi algorithm (b). Note that the brightness of image in (b) is enhanced to highlight the feature points 19
Figure 13 In-plane fatigue crack detection result of a C(T) specimen using the raw unstable 2D video 19
Figure 14 Comparison between the mean of all frames in the 2D video before (a) and after (b) the GMC 20
Figure 15 In-plane fatigue crack detection result of a C(T) specimen using the global motion compensated 2D video....20
Figure 16 Surface motion tracking of the 2D video: (a) feature points for cracked and uncracked regions; (b) differential displacement before GMC; (c) differential displacement after GMC 20
Figure 17 The initial frame of the 3D video with ROI (a): and all feature points detected by the Shi-Tomasi algorithm (b). Note that the brightness of image in (b) is enhanced to highlight the feature points....21
Figure 18 Out-of-plane fatigue crack detection result of the bridge girder specimen using unstable 3D video 22
Figure 19 Bundled camera paths of the two translation parameters 23
Figure 20 Mean of all frames in the 3D video before (a) and after (b) GMC....24
Figure 21 Out-of-plane fatigue crack detection result of the bridge girder specimen using the global motion compensated 3D video 24
Figure 22 Surface motion tracking of the 3D video: (a) feature points for cracked and uncracked regions; (b) differential displacement before GMC; (c) differential displacement after GMC 25
Figure 23 Overview of the distance-based fatigue crack detection method....26
Figure 24: Crack detection algorithm 27
Figure 25: Clustering of detected feature points 28
Figure 26: Ground truth labeling....29
Figure 27: Illustration of different values of IOU 29
Figure 28: The initial frame of the 2D video with the selected ROI (a) and all feature points detected by the Shi-Tomasi algorithm (b). Note that the brightness of image in (b) is enhanced to highlight the feature points 30
Figure 29: In-plane fatigue crack detection result of a C(T) specimen....30
Figure 30: Feature point distance history for crack and uncracked region 31
Figure 31: Quantification of detected crack in the C(T) speicmen using the displacement-based method (a) detetcted crack; (b) clustered crack; and (c) ground truth and clustered crack 31
Figure 32: Quantification of detected crack in the C(T) speicmen using the distance-based method (a) detetcted crack; (b) clustered crack; and (c) ground truth and clustered crack 31
Figure 33: The initial frame of the 3D video with ROI (a) and all feature points detected by the Shi-Tomasi algorithm (b). Note that the brightness of image in (b) is enhanced to highlight the feature points....32
Figure 34: Out-of-plane fatigue crack detection result of the bridge girder specimen 32

Figure 35: Feature point distance history for crack and uncracked region 33
Figure 36: Quantification of the detected crack in the bridge girder speicmen using the displacement-based method (a) detetcted crack; (b) clustered crack; and (c) ground truth and clustered crack.........................33
Figure 37: Quantification of the detected crack in the bridge girder speicmen using the distance-based method (a) detetcted crack; (b) clustered crack; and (c) ground truth and clustered crack.........................34
Figure 38: Two camera angles, view1 and view 2, for the distortion-induced fatigue test setup 34
Figure 39: Crack detection results with the ground truth from videos with View 1: (a) LC1, (b) LC2, (c) LC3, (d) LC4, (e) LC5, f) LC6, (g) LC7, (h) LC8, (i) LC9, and (j) LC10...35
Figure 40: Crack detection results with the ground truth from videos with View 2: (a) LC1, (b) LC2, (c) LC3, (d) LC4, (e) LC5, f) LC6, (g) LC7, (h) LC8, (i) LC9, and (j) LC10...36
Figure 41: Architecture of the developed AR software for fatigue crack inspection.................................37
Figure 42 Microsoft HoloLens 2nd generation (HL2) 38
Figure 43 The SQL database established to connect different software's components 38
FIGURE 44 Illustration of the Auto Anchoring System (AAS) 39
Figure 45 Virtual menu of the AR software for fatigue crack inspection..39
Figure 46 Visual mesh button reveals the HL2's representation of real-world surfaces steel girder test....40
Figure 47 Digital indicator's accuracy test (a) test setup (b) HL2's view 41
Figure 48 Virtual threshold functions automatically generated based on the crack detection results.........42
Figure 49 (a) In-plane flaw benchmark (schematic) (b) actual benchmark (photo)..................................42
Figure 50 Hologram of the crack detection result using in-plane crack benchmark.................................42
Figure 51 (a) Out-of-plane flaw benchmark (schematic) (b) actual benchmark (photo) 43
Figure 52 Hologram of the crack detection result using out-of-plane crack benchmark 43
Figure 53. The preliminary field experiment (a) inspector visually inspects the simulated crack using the AR software (b) simulated fatigue crack (c) hologram anchored on the simulated crack 43
Figure 54 Hardware and software needed to use the developed AR software..44
Figure 55 (a) Wi-Fi router (b) Hotspot device 44
Figure 56 MATLAB software (a) before running (b) waiting to receive new uploaded video from HL2 ..45
Figure 57 Virtually pushing the "Video Upload" button 45
Figure 58 Hardware used in AR-based fatigue crack inspection 46
Figure 59 Inspector opening the windows menu of the HL2 at the inspection site 46
Figure 60 Inspector opening the AR software 47
Figure 61 Inspector selecting video recording length using the slide bar 47
Figure 62 Inspector moving the AR software menu at a convenient location for inspection 47
Figure 63 Inspector collecting a video for crack detection 48
Figure 64 MATLAB crack detection algorithm running on the laptop/server...48
Figure 65 Inspector selecting the zero-threshold value to see the whole region analyzed for crack detection 49
Figure 66 Inspecting the initial feature point cloud detected from the inspected region 49
Figure 67 Inspector selecting a higher threshold value...49
Figure 68 Inspecting the detected crack with a higher threshold value 49
Figure 69 Inspector selecting the final threhosld value...50
Figure 70 Inspecting the detected crack with the final threhosld value 50
Figure 71 The joint KU-UNM team under the test bridge in the field..50
Figure 72 Span layout of the I-70 highway bridge with Span 3 selected for testing 51
Figure 73 (a) Distortion-induced out-of-plane fatigue crack at the web-gap region, and (b) in-plane fatigue crack on the web 51
Figure 74 Bridge inspector preparing for the inspection process..51
Figure 75 Opening the HL2 menu 52
Figure 76 Opening the AR software 52
Figure 77 Moving the AR software menu to a convenient location 52
Figure 78 Selecting video recording length using the slide bar 53
Figure 79 Hitting the video upload button to start recording..53

Figure 80 Recording a video for fatigue crack detection using the HL2 53
Figure 81 MATLAB crack detection algorithm running on the laptop/server simultaneously 53
Figure 82 Enabling the threshold button after the MATLAB processing is done 54
Figure 83 Hitting the button with zero threshold value to see the whole region analyzed for crack detection 54
Figure 84 Inspecting the initial feature points cloud detected from the inspected region...........................54
Figure 85 Selecting the threshold value of 0.2..55
Figure 86 Inspecting the crack detection result for threshold value of 0.2 55
Figure 87 Selecting the threshold value of 0.38..55
Figure 88 Inspecting the detected crack area with threshold value of 0.38 55

LIST OF TABLES

Table 1: IOU results for parametric study variations of out-of-plane fatigue crack in bridge girder specimen.35
Table 2 Test results for the digital indicator's accuracy..41
Table 3 Required software and hardware 44

GLOSSARY

AAS	Automatic anchoring system
AI	Artificial intelligence
AR	Augmented reality
ASCE	American Society of Civil Engineers
ASTM	American Society for Testing and Materials
CNN	Convolution neural network
CV	Computer vision
DBSCAN	Density-based spatial clustering of applications with noise
DOT	Department of transportation
FAST	Features from accelerated segment test
FHWA	Federal Highway Administration
GB	Gigabyte
GMC	Global motion compensation
GMM	Gaussian mixture model
HD	High definition
HL2	HoloLens2
IOU	Intersection over union
KDOT	Kansas department of transportation
KLT	Kanade-Lucas-Tomasi
LAN	Local area network
LC	Load case
LCR	Local circular region
LTE	Long Term Evolution
MRTK	Mixed Reality Toolkits
NDT	Nondestructive testing
RAM	Random-access memory
RANSAC	Random sample consensus
ROI	Region of interest
SHM	Structural health monitoring
SIFT	Scale-Invariant Feature Transform
SQL	Structured Query Language
SSD	Sum of squared distance
SURF	Speeded up robust features
UAV	Unmanned aerial vehicle
XAMPP	X-operating system, Apache, Mysql, Php, Perl

EXECUTIVE SUMMARY

Fatigue cracks developed under repetitive traffic loads are a major threat to maintaining the structural integrity of steel bridges. In fact, it is believed that 50~90% of all mechanical failures in metallic structures are fatigue related. The collapse of the Silver Bridge and the Mianus River Bridge are among the most well-known bridge failures directly caused by fatigue. Human visual inspection is currently the de facto approach for fatigue crack detection because as of today, administrations strongly rely on the experience of the experts to maintain their bridges. However, due to human limitations and the complex nature of bridge structures, fatigue crack inspections are time consuming, labor intensive, and lack reliability. Although non-destructive testing (NDT) techniques such as ultrasonic testing and acoustic emission have been used as supplemental methods to human visual inspection, they require complex testing equipment, and thus are not broadly used. As a result, inspecting the large steel bridge inventory in the United States remains a great challenge due to the lack of a human-centered, efficient and cost-effective methodology for detecting, tracking, and documenting fatigue cracks. On the other hand, if crack inspections could inform the inspector in the field, more reliable, efficient, and accurate assessment of the inventory could be achieved and documented.

Recently, computer vision has shown great potential as a non-contact, low-cost, and versatile platform for structural health monitoring (SHM). However, most computer-vision-based crack detection methods rely on still images to extract edge features of cracks. As a result, distinguishing real fatigue cracks from crack-like surface features, such as scratches, corrosion marks, and structural boundaries remains a major challenge. Moreover, compared with cracks in concrete structures and pavements, which are often easily visible to human eyes, fatigue cracks in steel bridge components are much smaller and more difficult to see, and can even be invisible upon crack closure, presenting a bigger challenge for crack detection using computer vision. In addition, inspectors currently lack an effective way to efficiently interact with new and historic inspection data. Such human-centered ability has been identified as one of the top interests of bridge inspectors, as it not only improves inspection quality but also facilitates decision-making in the field.

To overcome the above challenges, as an entirely new concept, we propose to integrate computer-vision-based motion tracking and augmented reality (AR) techniques to empower bridge inspectors to perform robust fatigue crack detection, characterization, tracking, and documentation in the field. First, our computer vision algorithm differs from prior ones in that it does not rely on edge features of images – a major advance in terms of accuracy and usability for fatigue crack detection. Instead, it is based on recording a short video of the structure under fatigue loading, tracking the surface motion of the structure through the proposed computer vision algorithm, and analyzing the surface motion pattern to reveal the 'breathing' of fatigue cracks. In addition, crack width can be quantified with sub-millimeter accuracy using the tracked surface motion. Second, to overcome the limitation of the technique in the field for the inspectors, this research will integrate computer vision with Augmented Reality (AR), so the inspector can see the crack information such as the crack geometry, realized via holograms overlaid on top of the bridge surface. Microsoft Holographic lenses, popularly known as HoloLens, have the capacity to track eye movements, listen to voice commands, and follow hand gestures, and are currently used in areas such as medicine. The wearable AR device will greatly increase bridge inspectors' ability to perform accurate and reliable on-site inspection in a human-centered manner. Moreover, inspectors will be able to interactively manage inspection results and compare with historic data for efficient decision-making.

The main findings of the project are summarized as follows:

1. Videos recorded from non-stationary platforms such as mixed-reality headsets have camera induced global motion. Laboratory testing based on compact, C(T) and bridge girder specimens demonstrated that GMC is necessary to detect the fatigue crack successfully using feature point displacement-based method with unstable videos. The proposed GMC methods were effective at removing camera motion

from 2D and 3D videos. The developed displacement-based crack detection algorithm successfully detects in-plane and out-of-plane fatigue cracks. However, the computation time for crack detection using GMC and displacement-based method is as high as 1.75 seconds per frame for machine with intel i9 processor and 64 GB of RAM, making it challenging for near real-time fatigue crack inspection.

2. To facilitate near real-time crack detection, a novel feature point distance-based approach was developed. The differential motion due to crack movement is detected by analyzing distance between feature pairs on the surface, thus avoiding the need for GMC. The proposed method was validated in laboratory testing for near real-time fatigue crack detection in 2D and 3D videos. The computation time for the distance-based approach is 0.2 seconds per video frame, making it suitable for near real-time fatigue crack inspection. This approach was integrated with the AR software for fatigue crack detection.

3. A method was developed to quantify the detected crack using discrete feature points over continuous fatigue crack. The proposed crack detection approach had an IOU score as high as 0.77 implying the robustness of the crack detection algorithm in laboratory setting.

4. AR software integrating computer vision-based crack detection algorithm was developed for human-machine collaboration. A novel AAS system was developed through an invisible rectangle to overcome challenges in HL2's 3D mapping and to anchor the hologram in the right location and orientation. The AAS system is practical for inclined surfaces as well as surfaces perpendicular to user's line of sight. The AAS system was validated in both laboratory and field tests.

5. The developed AR software was successfully tested in the laboratory on a half-scale bridge girder specimen. This proof-of-concept testing shows that the developed AR-based fatigue crack inspection tool integrated with computer vision techniques is functioning seamlessly and smoothly, achieving the originally proposed goals.

6. The field test demonstrated the seamless flow of the developed AR-based software integrated with computer vision algorithms and the many features to support human-centered bridge inspections. The result proves the effectiveness of the proposed crack detection concept by detecting differential surface motions. The result also highlights the challenge for field implementation especially for fatigue cracks that experience extremely small openings under operational loadings. Further research is needed to continue to improve the AR-based software and the crack detection algorithm.

IDEA PRODUCT

The IDEA product developed in this project is a non-contact, human-centered, and interactive fatigue crack inspection technique for steel bridges enabled by advanced computer vision algorithms and wearable augmented reality (AR) techniques. The goal is to enable bridge inspectors to detect, localize, and measure fatigue cracks using AR devices in the field in near real-time, simplifying on-site documentation, tracking, and decision making through AR holograms. Using advanced computational imaging and AR, this innovation will greatly improve the accuracy and efficiency of bridge inspection, especially for fatigue cracks, which are difficult to detect and measure compared with other types of cracks in civil infrastructure. This project is focused on developing a human-centered diagnostic technology to enhance field detection and quantification of cracks in highway infrastructure, which is a high priority need of state highway agencies.

The current state of practice for fatigue crack detection relies on human visual inspection. However, fatigue cracks are challenging to detect due to small crack lengths and openings, especially in early stages of crack propagation. Various nondestructive testing (NDT) methods have been used to supplement human inspection, but they still require complex equipment and extensive human involvement for data collection. Using the AR device, this innovation will enable bridge inspectors to accurately detect and quantify fatigue cracks by tracking motions of the structural surface by recording short videos and uncovering differential motion patterns produced by the ‘breathing’ of cracks under traffic loading. Therefore, it is able to distinguish real fatigue cracks from crack-like features such as corrosion marks, scratches, and structural boundaries, which pose a major challenge to traditional computer-vision-based crack detection methods that rely on edge features. In addition, through the holographic images and human-centered environment of AR, bridge inspectors can interact with quantitative crack data for automated damage tracking, documentation, and decision-making for bridge maintenance in the field during inspection.

The proposed innovative solution is economically feasible and will be easy to use in practice. It requires one set of a wearable AR device such as the Microsoft HoloLens to provide computational imaging for crack inspection as well as holograms overlaid on top of the bridge surface for documentation, tracking, and interactive decision-making. The world-facing camera on the HoloLens can be used for videotaping if cracks are directly accessible by the HoloLens. Otherwise, a hand-held camera can be used. A user-friendly software program is created to enable the described functionalities. Adopting the proposed solution in bridge inspection for fatigue cracks will achieve major cost savings by reducing inspection time as well as documentation efforts, avoiding bridge closures and the need for expensive equipment such as NDT devices. The developed tool is hands free and very intuition driven, so is easy to train bridge inspectors to use. It enables inspectors to identify and quantify the crack on the actual structural surface they are inspecting.

CONCEPT AND INNOVATION

As an entirely new concept, we propose to integrate computer-vision-based motion tracking and augmented reality (AR) techniques to empower bridge inspectors to perform robust fatigue crack detection, characterization, tracking, and documentation in the field. First, our computer vision algorithm differs from prior ones in that it does not rely on edge features of images – a major advance in terms of accuracy and usability for fatigue crack detection. Instead, it is based on recording a short video of the structure under fatigue loading, tracking the surface motion of the structure through the proposed computer vision algorithm, and analyzing the surface motion pattern to reveal the 'breathing' of fatigue cracks. In addition, crack width can be quantified with sub-millimeter accuracy using the tracked surface motion. Second, to overcome the limitation of the technique in the field for the inspectors, this research will integrate computer vision with Augmented Reality (AR), so the inspector can see the crack information such as the crack geometry, realized via holograms overlaid on top of the bridge surface. Microsoft Holographic lenses, popularly known as HoloLens, have the capacity to track eye movements, listen to voice commands, and follow hand gestures, and are currently used in areas such as medicine. The wearable AR device will greatly increase bridge inspectors' ability to perform accurate and reliable on-site inspection in a human-centered manner. Moreover, inspectors will be able to interactively manage inspection results and compare with historic data for efficient decision-making. Figure 1 illustrates the proposed framework, which consists of the following steps:

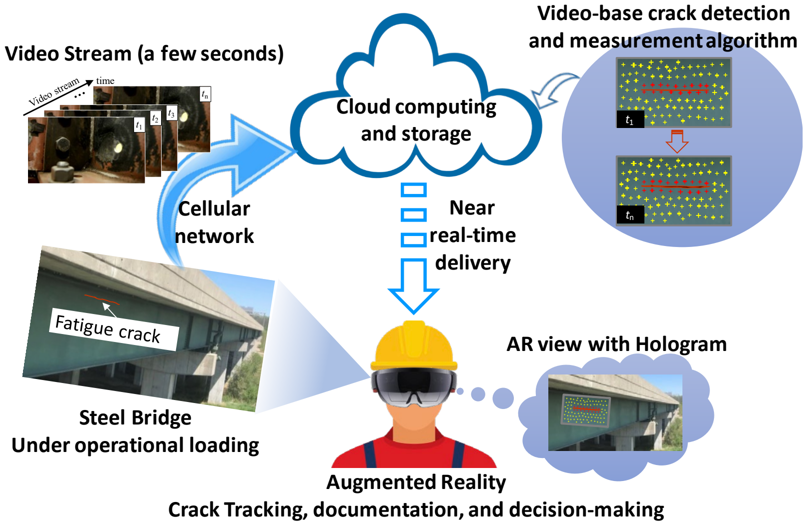

Video Stream (a few seconds)
Video stream
time
t_1
t_2
t_3
t_n
Cellular network
Cloud computing and storage
Video-base crack detection and measurement algorithm
t_1
t_n
Near real-time delivery
Fatigue crack
Steel Bridge
Under operational loading
AR view with Hologram
Augmented Reality
Crack Tracking, documentation, and decision-making

Figure 1 Fatigue crack inspection using computer vision and AR

Step 1: A short video stream of the target structural surface under operational loading is recorded using the world-facing camera in the HoloLens or a hand-held camera depending on the accessibility of the crack location. The video is typically several seconds long, capturing multiple fatigue cycles. The video is then uploaded to the cloud server for processing through the 4G LTE cellular network, which would take a few seconds.

Step 2: The camera, either hand-held or integrated in the HoloLens, will inevitably introduce extraneous motion into the video, which may contaminate the surface motion data to be extracted later. Therefore, camera motion compensation will be applied to stabilize the video.

Step 3: With the stabilized video, a dense set of feature points, which correspond to the pixels with high intensity gradient, are detected in each video frame. These feature points serve as the natural targets on the structural surface to accurately track the surface motion.

Step 4: Subsequently, the pattern of the surface motion is analyzed. For regions with no cracks, the feature points will move in a similar way. However, for regions with fatigue cracks, the crack breathing under fatigue loading will cause differential movement patterns among the feature points around the crack.

Step 5: By highlighting the feature points exhibiting significant differential movement patterns, the fatigue cracks can be localized. Moreover, using the accurately tracked surface movement, the crack opening width, crack length, and positions can be accurately recorded and documented.

Step 6: Having the feature points that highlight the crack and the calculated dimensions, the hologram for the crack can be programmed in the computer server, which can be (1) anchored and (2) displayed on top of the crack so the inspectors can see the quantified crack features in the field through the HoloLens. Bridge inspectors can also interact with the hologram to perform tasks such as documentation and tracking for decision-making.

Since this method detects fatigue cracks by capturing crack breathing under traffic loading, false positive results due to other crack-like surface features can be avoided. No special lighting and surface preparation are necessary to perform the motion tracking. The use of AR for the field inspector enables for the first time the ability of interactively quantifying and observing the crack opening in the field. This project also aims at achieving near real-time video processing through algorithm optimization so inspectors can receive almost instantaneous results during the inspection.

INVESTIGATION

The investigation is conducted in three tasks. Task 1 is focused on creating and validating the core computer vision algorithms for video-based fatigue crack detection. Task 2 develops the augmented reality (AR) interface for interactive crack inspection. Task 3 integrates the computer vision algorithm and AR interface to achieve near real-time human-centered crack inspection and performs laboratory and field testing. The following sections present the detailed investigations and findings from the three tasks.

DEVELOPMENT OF COMPUTER VISION ALGORITHMS FOR FATIGUE CRACK DETECTION

This section presents the creation and validation of the core computer vision algorithms for video-based fatigue crack detection. The main challenge addressed in this task is reliable fatigue crack detection using a short video recorded by a moving camera, as the AR headset would inevitably introduce camera movement during inspection. Two solutions are proposed and investigated in this section. The first method is based on post-processing the unstable video to remove the camera motion through global motion compensation (GMC) algorithms, in which both 2D and 3D video scenes are considered. While the GMC-based method produced satisfactory crack detection results, its high computation cost makes it challenging to achieve near real-time inspection. Therefore, a distance-based crack detection method is proposed which achieved high accuracy without the need for GM. The second approach is finally chosen to be integrated with the AR interface for the laboratory and field validations.

Literature Review

Recent reports on infrastructure surveys (ASCE 2021, ARTBA 2020) reveal that 42% of all bridges in the United States are at least 50 years old, and 46,154 out of the 617,000 of the bridges of the bridges are in poor condition and structurally deficient. Fatigue cracks in steel bridges develop under repetitive loading and are very small in their early stages. However, depending on connection details and secondary load effects, fatigue cracks can grow rapidly and lower the structural integrity, leading to catastrophic failure (Haghani et al., 2012). Thus, timely detection and monitoring of fatigue cracks becomes important to prevent excessive damage to steel bridges and other metallic civil infrastructure. The current practice of fatigue crack inspection mainly relies on human vision, but this approach is labor intensive, error-prone, and time-consuming. A study conducted to evaluate effectiveness of visual inspection revealed that early-stage fatigue cracks have a low probability of detection (Campbell et al., 2020). Moreover, in a bridge inspection study conducted in Virginia and Pennsylvania by the Federal Highway Administration (FHWA), only two of 49 bridge inspectors correctly identified the fatigue crack (Holusha and Chang, 2007). Various structural health monitoring (SHM) techniques have been developed for damage detection and monitoring structural health conditions (Jang et al., 2013; Sirca and Adeli, 2012; Amezquita-Sanchez and Adeli, 2016; Asadollahi and Li, 2017). Non-destructive testing (NDT) techniques such as acoustic emission (Roberts and Talebzadeh, 2003) and piezoelectric sensors (Inh and Chang, 2004) were used to monitor fatigue cracks. Fatigue cracks have also been detected using sensors by monitoring strain changes (Glišić et al., 2016; Kharroub et al., 2015; Kong et al., 2016, Taher et al., 2022). However, these techniques generally require complex system setups including a data acquisition system and sensor installation, which often once installed are not easily moveable. Moreover, such sensors and instrumentation can deteriorate and malfunction due to harsh outdoor environments.

Recent advancements in computer vision have automated many human visual tasks. Computer vision-based algorithms are increasingly being used in SHM as they offer non-contact, low-cost solutions. Some early studies using computer vision for crack detection include extracting edge-like features using filters (Abdel-Qader et al., 2003), image contrast enhancement and filter application for crack segmentation (Iyer and Sinha, 2006), percolation model to extract texture for crack detection (Yamaguchi et al., 2008), multisequential image filtering for crack detection (Nishikawa et al., 2012), and morphological features with support vector machines to identify crack thickness (Jahanshahi and Masri, 2013). However, these methods use traditional image processing to identify intensity changes and edge features of cracks, making it challenging to differentiate real cracks from crack-like features such as corrosion marks, scratches, surface textures, and structural boundaries. For example, Yeum and Dyke (2015) proposed a technique for automated crack detection in steel bridges using several computer vision techniques and reported that machine-generated scratches were also detected as cracks.

Recently, advancements in artificial intelligence (AI) have triggered significant interest in crack detection using deep learning and computer vision. Cha and Choi (2017) proposed to use convolutional neural networks (CNN) to accurately detect cracks in images of concrete components under various conditions including bright lighting, shadow, blur, and close-up. Dung et al. (2019) trained a CNN classifier to detect fatigue cracks in steel bridge gusset plates. The dataset images were divided into smaller images for training that do not include crack-like features, making the trained network susceptible to inaccurate predictions in presence of crack-like features. Deep learning-based crack detection methods are strongly dependent on the data used for training, and may also face challenges in distinguishing crack-like features from true cracks. Dong et al. (2021) trained an encode-decoder network based on the modified U-net structure to segment fatigue crack pixels in the image. The trained network was detecting marker curves, edges of weld lines, and handwriting as small fatigue crack segments and concluded that the model might give wrong predictions when the test image has a large variation in surface texture, crack width, and lighting condition from the training dataset.

While the studies summarized above are primarily focused on processing individual static images which provide only spatial information, utilizing a video stream which contains both spatial and temporal information of the subject presents new opportunities. For example, a video-based approach for dynamic displacement monitoring of civil infrastructure has been extensively studied in SHM through target-based (Wahheh et al.,2003); Park et al, 2010; Feng et al. (2015) and targetless approaches (Khuc and Catbas, 2017, Caetano et al. 2011, Feng and Feng 2017, and Yoon et al., 2016). However, these are focused on tracking global structural responses rather than detecting localized damage such as cracks.

Kong and Li (2018) proposed a novel video-based approach for fatigue crack detection through structural surface motion tracking. A short video stream is recorded using a consumer-grade camera to capture a region of interest responding under a few fatigue load cycles. Then, salient features are extracted to track the surface motion and a fatigue crack is detected by identifying differential motion patterns induced by the crack opening and closing under fatigue load that exceed a predefined threshold. This method does not rely on edge features of cracks and performs well in the presence of dust, rust, corrosion, other crack-like features, and even can detect cracks that are invisible to human eyes. It has been shown to be robust under various lighting conditions and camera positions. However, the original design of the approach works only for videos taken by a fixed camera, and selecting the appropriate threshold values for different scenarios could be challenging. Furthermore, it is not always practically possible to set up a stationary support such as a tripod for the camera to videotape the fatigue prone regions of a structure. In such cases, video must be taken from a moving camera such as one carried by hand, boat, unmanned aerial vehicle (UAV), or wearable devices. As a result, as will be illustrated in this paper, the tracked surface motion will be contaminated by camera movement, leading to failed crack detection results.

Various methods have been developed in the field of computer vision to remove unwanted global motion induced by camera movement. The process of removing camera-induced motion is referred to as global motion compensation (GMC), which is similar to video stabilization. In general, GMC is focused on removing the global motion of the camera, while video stabilization aims at removing high frequency ‘jitters’ for a smoothed video. A video scene can be broadly classified into 2D and 3D scenes. A scene is considered 2D if it is planar, and/or is sufficiently distant from the camera, or the variations of the distance from the scene to the camera are small relative to the overall distance to the camera (Irani and Anandan, 1998). Lee et al. (2019) used digital image correlation technique with camera modeling to estimate homography between 2D scene. Darrell and Pentland (1991) decomposed a scene into multiple layers for motion compensation using the 2D GMC technique. Burt et al. (1991) provided insights into the role of pyramid-based approach in the stabilization algorithm. Bergen et al. (1992) developed hierarchical model-based motion estimation approach that has been proven to be very robust, even in presence of moving objects. Therefore, the latter method is adopted in this study to perform GMC for 2D videos and is discussed in more detail in later sections.

For GMC of 3D videos, early studies (Lawn and Cipolla, 1994; Torr and Murray, 1994; Torr et al.,1995) attempted to interpret global motion in terms of 3D camera motion components, but with low success for videos with high parallax effects. In the past decade, numerous methods have been developed for 3D video stabilization to remove camera jitter (e.g., Liu et al., 2011; Goldstein and Fattal, 2012; Grundmann et al., 2012). Some of the recent deep learning-based video stabilization methods include those reported by Yu and Ramamoorthi (2020), Shi et al. (2022), and Lee et al. (2021). These deep learning models need to be retrained on specific video datasets which is computationally expensive, and creating and labeling the dataset is time consuming. Safdarnejad et al. (2015) proposed an approach for 3D GMC in the presence of predominant foreground. However, this method only estimates a single transformation matrix which is incapable of describing differential motion due to parallax effects. Liu et al. (2013) proposed the bundled camera paths method for removing jitters from 3D videos. By dividing a 3D scene into a grid of smaller scenes and estimating individual camera paths for each cell, this method is able to handle parallax effects. For GMC of 3D videos for fatigue crack detection in bridges, our approach adopts the bundled camera paths method as the basic framework and made improvements for removing the global camera motion.

In this report, we first extended the approach developed by Kong and Li (2018) to detect fatigue cracks in videos recorded with a moving camera through GMC. In particular, videos containing 2D and 3D scenes are both considered. For 2D videos, the hierarchical model-based motion estimation approach by Bergen et al. (1992) is adopted to perform 2D GMC. For 3D videos, a GMC strategy is developed by extending the bundled camera paths method by Liu et al. (2013) with high-pass filtering to handle parallax effects. In addition, an adaptable threshold scanning strategy is proposed to address the challenge of threshold selection for identifying appropriate differential surface motion for crack detection. The proposed methodology is validated through laboratory testing, demonstrating its ability to reliably detect realistic fatigue cracks through a moving camera for both 2D and 3D scenarios.

One limitation of the above displacement-based crack detection approach which requires GMC is the high computation cost, making it difficult to achieve near real-time fatigue crack detection. To enable the proposed AR-based fatigue crack inspection, a novel distance-based approach was developed to address the computational challenges of the displacement-based method. Instead of relying on the absolute displacements of feature points, the distance-based approach computes distance change between feature points to determine differential surface motion due to fatigue crack opening and closing, thus avoiding the need for GMC. The proposed methodology is validated through laboratory testing, demonstrating its ability to reliably detect realistic fatigue cracks through a moving camera for both 2D and 3D scenarios in near real-time. In the following sections, both methods are investigated and presented.

Displacement-based Fatigue Crack Detection Algorithm

A new computer vision-based algorithm was developed to detect fatigue cracks in videos recorded from a non-stationary camera. Figure 2 illustrates the major steps involved in the developed fatigue crack detection method. First, a short video of a fatigue crack prone region in the structure is recorded under several fatigue load cycles. The video can be recorded with various types of nonstationary devices such as a handheld camera, mixed reality head-mounted device, and unmanned aerial vehicle (UAV). Camera motion is introduced in the recorded video because camera was not fixed during videotaping. Consequentially, the video frames do not share the same coordinate system. To capture the true structural surface motion for fatigue crack detection, camera motion needs to be compensated. The process of removing camera-induced motion is referred to as global motion compensation (GMC).

The second step in the proposed approach is to minimize the camera induced motion in the recorded video by applying GMC. Generally, for typical steel girder bridges, in-plane fatigue cracks occur in planar surfaces such as the girder's web or flange, and out-of-plane fatigue cracks develop at web-gap regions, such as a crossframe to girder connection. These situations result in videos that are considered either 2D or 3D, and different algorithms are used to perform GMC.

The next step after camera motion compensation is to detect salient features within the region of interest (ROI) in the first video frame. Movement of each feature point is then tracked through the video stream (Figure 2c). The crack detection algorithm analyzes the displacement pattern of the features to detect and localize a fatigue crack. The features' movement pattern is evaluated against nearby features within a local circular region (LCR). For an LCR that does not contain a fatigue crack, the feature points share a similar movement pattern as they reflect the rigid-body movement of the uncracked surface; however, when a fatigue crack is present, feature points within the LCR show two different displacement patterns due to crack opening/closing under the fatigue load. By scanning all feature points within the ROI using the LCR and detecting differential movement patterns that exceed a threshold value, feature points from the LCRs that contain a fatigue crack are isolated and highlighted. In the end, the highlighted feature points collectively trace the location of the fatigue crack. Details of each major component of the proposed methodology are described in the following subsections.

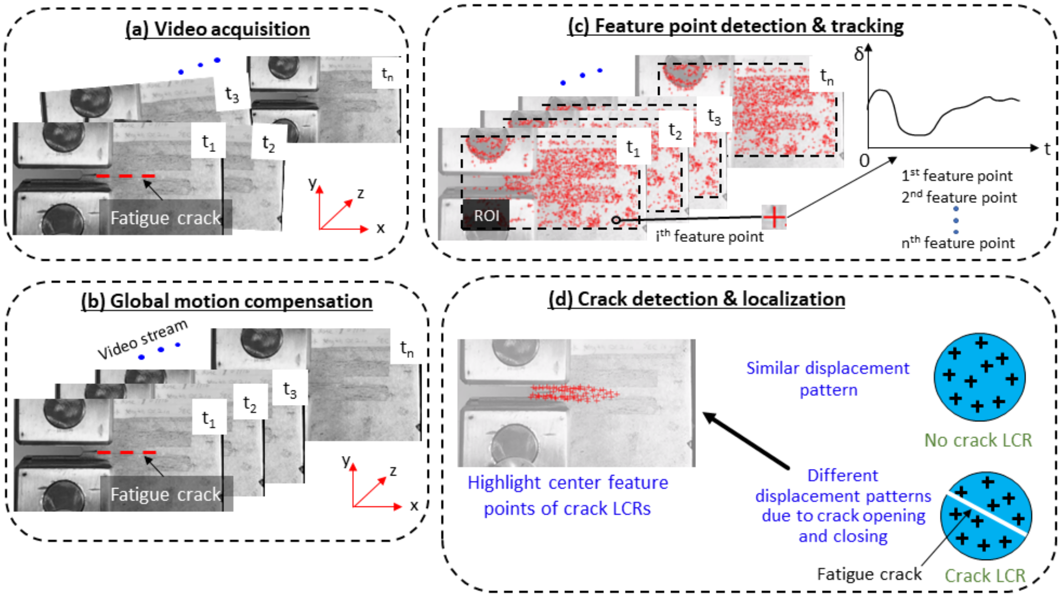

(a) Video acquisition
t3
tn
t1
t2
Fatigue crack
y
z
x
(c) Feature point detection & tracking
δ
tn
t3
t2
t1
ROI
ith feature point
0
t
1st feature point
2nd feature point
nth feature point
(b) Global motion compensation
Video stream
tn
t3
t2
t1
Fatigue crack
y
z
x
(d) Crack detection & localization
Similar displacement pattern
No crack LCR
Highlight center feature points of crack LCRs
Different displacement patterns due to crack opening and closing
Fatigue crack
Crack LCR

Figure 2 Overview of the fatigue crack detection method based on a moving camera

Global Motion Compensation (GMC)

GMC removes unwanted camera motion from the video through geometric transformation between two video frames such that all frames share the same coordinate system. Figure 3 demonstrates how GMC realigns video frames to share a single coordinate system. Figure 3a shows the first frame of a video with its local coordinate system which is aligned with the global coordinate system. Due to camera motion, for other frames, such as the second frame, their local coordinates are not aligned with the global coordinates, as illustrated in Figure 2b using a pure rotation as an example. Note that other deviations such as translation, shear, and scaling can also be compensated. Consequently, the coordinate system of the second frame is oriented differently than the first frame. After performing GMC, the coordinate system of the second frame is oriented identical to the first frame as demonstrated in Figure 3c.

As explained in the introduction, GMC methods can be broadly categorized into two groups: (i) 2D techniques, and (ii) 3D techniques. A scene is called a 2D scene if it is planar or contains a flat 2D surface, and/or is sufficient distant from the camera, or the deviations from a planar surface are small relative to the overall distance of the scene from the camera (Irani and Anandan, 1998). In the context of fatigue crack detection in steel bridges, a typical scenario of 2D scene is a fatigue crack that propagates in the flange or web of a steel girder. For 2D scenes, global motion can be modeled by a single global 2D transformation matrix between the two frames as follows:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \alpha \begin{bmatrix} a_1 & a_2 & b_1 \\ a_3 & a_4 & b_2 \\ v_1 & v_2 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \qquad (1)$$

where $(x, y, 1)$ and $(x', y', 1)$ are the coordinates before and after the 2D transformation, respectively. a_1, a_2, a_3, a_4 are pure affine components, b_1 and b_2 are the translation components, v_1 and v_2 are pure projective components of the camera motion, and α is the scaling factor that holds the third component of the coordinates 1. Affine transformations are used to describe scaling, skewing and rotation between two images. Projective transformations do not preserve parallelism, length, and angle, while affine transformations preserve parallelism. When the scene contains multiple planes and is not viewed from a sufficient distance, significant depth variations occur. In this case, a single geometric transformation is no longer sufficient to remove the camera-induced motion. Such a scene is called a 3D scene or multiplanar scene. One typical example of a 3D scene, shown in Figure 4a, is the web-gap region of a bridge girder with a fatigue crack developed in the weld between the web and connection plate. The three different planes are highlighted in Figure 4b including the girder web, flange, and the connection plate. All different members are at different distances from the camera lens and are also oriented in different directions. When the camera is translating while recording a 3D scene, regions within the scene closer to the camera would appear to be moving faster than those further away, and these multiple planes have substantially different 2D motions, a phenomenon called the parallax effect, or 3D parallax motion (Longuet-Higgins and Prazdny, 1980; Rieger and Lawton, 1985). Note that parallax motions are only induced by camera translation and 3D scene variations, and camera rotation or zoom do not cause parallax (Irani and Anandan, 1998). Due to the difference between 2D and 3D scenes, different GMC strategies are necessary, which will be discussed as follows.

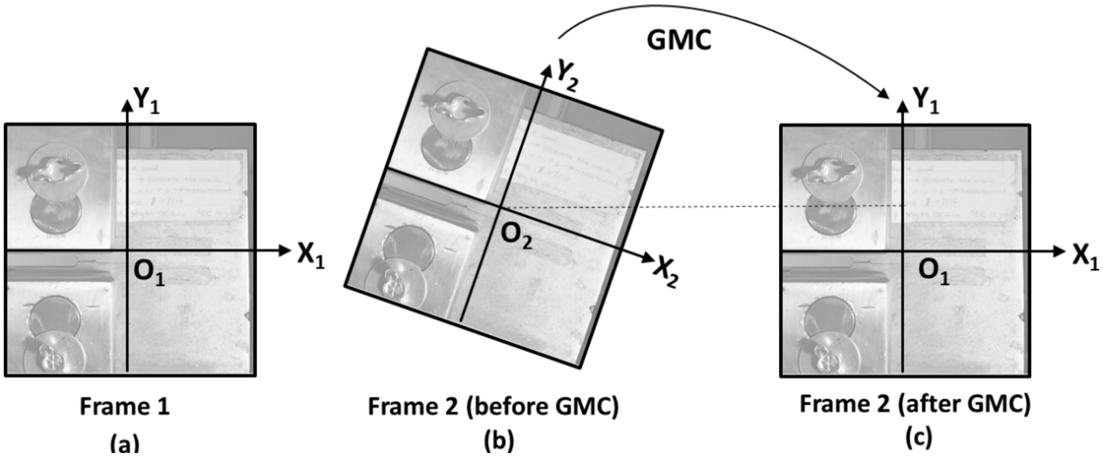


Figure 3 Illustration of geometric transformation for GMC

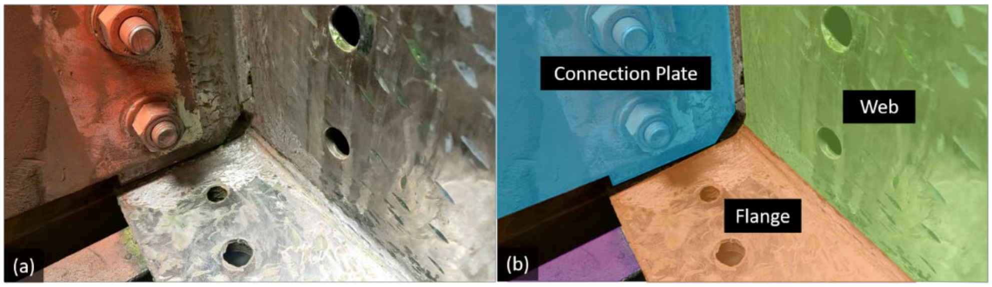


Figure 4 Web-gap region of a bridge girder (a) with three different planes highlighted (b)

GMC for 2D Videos

In-plane fatigue cracks usually develop in the web or flange of a bridge girder. The video of bridge girder web with fatigue cracks recorded with a narrow field of view can be considered as a 2D video as the web is a planar flat surface. For 2D scene videos, we propose to use the hierarchical estimation framework proposed by Bergen et al. (1992) that provides accuracy and efficiency for global motion compensation. The basic components of this framework include: 1) pyramid construction, 2) motion estimation, 3) image wrapping, and 4) coarse-to-fine refinement.

In this method, GMC is performed by keeping the first video frame as the reference frame and other frames as target frames. The first frame is paired with other frames in the video individually. For each pair, a multi-scale representation of images, termed the pyramid, is created by generating images with different levels of resolution. If the displacement of an object between the two images is larger than 1-2 pixels, computing such displacement can be expensive for a matching process. However, computation time can be reduced significantly if the large displacements can be computed using low resolution. Subsequently, the accuracy of the displacement estimation is improved by incrementally estimating small displacements using higher resolution images. Motion estimation between two images involves minimizing the sum of squared distance (SSD) using Gauss-Newton minimization. In our case, motion is defined as affine motion which includes scaling, translation, rotation, and shear. So, it is usually possible to approximate the motion of a 2D surface as an affine transformation. Equation (2) is used to estimate the motion parameters between the reference image and target image.

$$\left[\sum \boldsymbol{X}^T (I_{\mathbf{x}})(I_{\mathbf{x}}^T)\, \boldsymbol{X}\right] \delta a = -\left[\sum \boldsymbol{X}^T\, (I_{\mathbf{x}})(I_{\mathrm{t}})\right] \qquad (2)$$

In the above motion estimation equation, δa is an unknown vector of affine parameters $[a_1\ a_2\ b_1\ a_3\ a_4\ b_2]^T$, I_{t} is the temporal derivative of image intensities, $\boldsymbol{X}$ is a matrix of pixel coordinates, and $I_{\mathbf{x}}$ is the image gradient vector $[I_x\ I_y]^T$, where, I_x and I_y are image spatial derivatives in x and y direction, respectively. This linear equation can be solved to compute the affine motion parameters between two images.

The third component of this framework is image warping. The estimated motion parameters are applied to the target image to remove the camera motion. It is carried out using bicubic interpolation since it produces smoother images and results in fewer interpolation artifacts. The final step is coarse-to-fine refinement, which is basically propagation of current estimated motion estimation from one level to the next level of the pyramid. Motion estimation is performed in the image with the lowest resolution which is at the top level of the pyramid. Then, the global motion parameters are transmitted to the next level of pyramid, where they are used as the initial estimate. The entire propagation process is carried out to the bottom level of the pyramid to reach the original resolution of the images. More details of the 2D hierarchical GMC method can be found in Bergen et al. (1992).

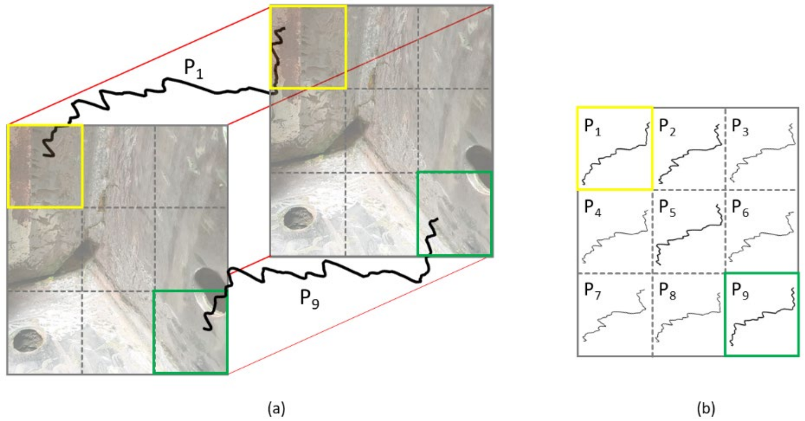


Figure 5 Bundled camera paths (a) video frames divided into grid cells (b) camera paths of grid cells (x-axis translation over time)

GMC for 3D Videos

As discussed previously, when multiplane or 3D scenes are present, a single geometric transformation matrix is no longer able to remove the camera-induced motion. To this end, we have adopted the bundled camera paths method by Liu et al. (2013) and extended it to perform GMC from 3D videos. The key to the bundled camera paths method is to divide each video frame by a mesh grid into multiple cells and estimate the camera motion of each grid cell individually. Figure 5 illustrates this concept using the 3D scene shown in Figure 4 divided into a 3 × 3 mesh grid as an example. The camera motions associated with the cells are different due to their different distances to the camera, hence are estimated individually. Therefore, this approach can estimate the 3D parallax motion present in the video. However, the original method was aimed at removing high-frequency jitter to generate a smoothed video with low camera shaking, rather than removing the global motion. In this study, we have extended the bundled camera paths method to further remove the global camera motion through high-pass filtering, producing global motion compensated video. Figure 6 shows the flowchart of the proposed GMC strategy based on the bundled camera paths method with the extension highlighted in the dashed box.

The first step of the bundled camera paths method is to divide the video into multiple grid cells. The number of grid cells depends on the complexity of the geometry present in the scene. For example, a larger number of grid cells should be selected if the video is recorded very close to the scene to accommodate severe differential motion due to the parallax effect. Once the video frames are meshed into grid cells, feature points are identified and matched in each grid cell to estimate the spatially varying 2D homographies. This type of motion model is between the global homography and per-pixel optical flow field. The outliers or incorrectly matched features are eliminated by using the Random Sample Consensus (RANSAC) algorithm applied at coarse scale (the whole image) and fine scale (4 × 4 sub-images). Specifically, features with fitting errors exceeding 6% of the image width at the coarse scale and 2% of the image width at the fine scale are discarded.

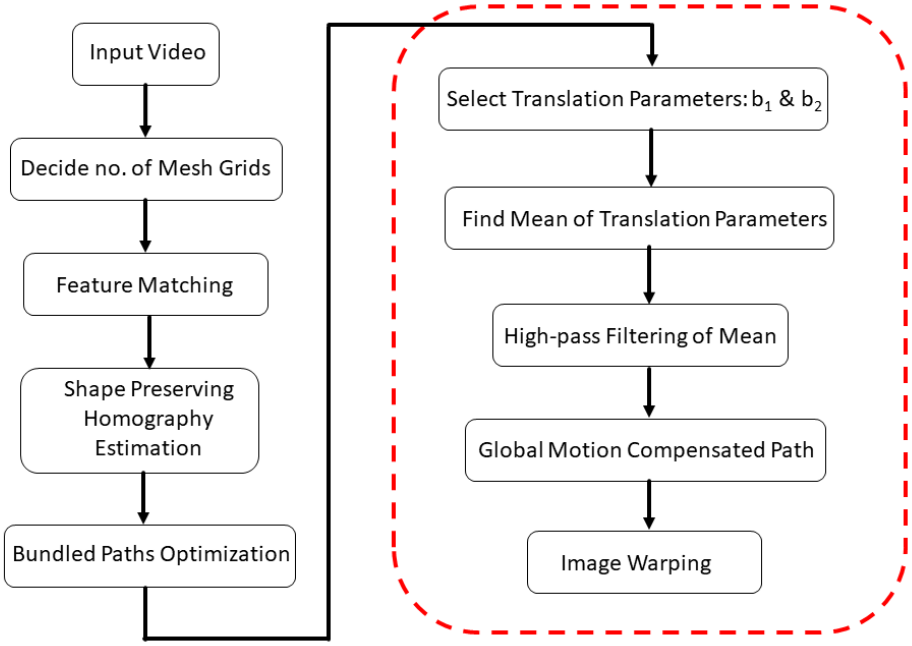

Input Video
Decide no. of Mesh Grids
Feature Matching
Shape Preserving Homography Estimation
Bundled Paths Optimization
Select Translation Parameters: b_1 & b_2
Find Mean of Translation Parameters
High-pass Filtering of Mean
Global Motion Compensated Path
Image Warping

Figure 6 Flowchart of the proposed extended bundled camera paths method for GMC of 3D videos

It may be possible that features are not detected in some grid cells due to the lack of texture/gradient in the region. To estimate homography in those grid cells, an ‘as-similar-as-possible’ constraint (Igarashi et al. 2005, Schaefer et al. 2006) is applied, which estimates the position of cell vertices without features by minimizing the distortion among cells. The series of estimated homographies over the video length of a grid cell is called camera path. Once the camera paths of all grid cells are estimated, the bundled camera paths are smoothed and optimized together to remove the camera jitter by minimizing the following objective function:

$$\sum_{i} \mathcal{O}\left(\{P_i(t)\}\right) + \sum_{t} \sum_{j \in N(i)} || P_i(t) - P_j(t) ||^2 \qquad (3)$$

where P_i is the camera path function of the i^th grid cell to be optimized, and includes the eight neighbors of the i^th grid cell. The first term in Eq. (3) is the objective function for each single camera path to stabilize the path while minimizing geometric distortions, and the second term is a regularization term to ensure smoothness between adjacent paths. Detailed descriptions of the objective function can be found in Liu et al. (2013). The method adopts a Jacobi-based iteration to solve the minimization problem. The optimized camera paths are typically obtained after 20 iterations.

Since the bundled camera path is designed to remove camera jitter, the optimized camera paths still contain the global camera motion. Here we extend the bundled camera paths method to further reduce the global motion through high-pass filtering. As mentioned previously, parallax motion present in 3D videos can introduce differential surface motion, leading to unreliable crack detection. Since parallax effects are only induced by the translation parameters of the camera motion, and rotation or zooming do not cause parallax in videos (Irani and Anandan, 1998), to compensate for the global motion and reduce parallax effects for 3D videos, we apply high-pass filtering to the translation components of the camera motion. However, individual high-pass filtering of each camera path would result in incompatibility and distortion between the grid cells. Therefore, a strategy of high-pass filtering of the mean camera path is adopted. Specifically, the means of the translation components b1 and b2 are first computed in the x and y directions, respectively, which are then high-pass filtered to remove the low-frequency components of the mean camera motions. Subsequently, for each direction, the difference between the mean camera paths before and after the high-pass filtering is then applied to adjust the individual camera path of each grid cell. This strategy ensures compatibility between adjacent grid cells and avoids distortion in the global motion compensated video.

Feature Point Detection and Surface Motion Tracking

Feature point detection and tracking are critical to the success of fatigue crack detection, since accurate displacement time histories of the structural surface are key to identifying differential motion patterns induced by crack opening and closing. Feature points are specific locations in images with high intensity gradient such as peaks, corners, edges, objects etc. Feature point-based techniques have wide applications in computer vision such as image-stitching (Zoghlami et al., 1997; Brown and Lowe, 2007), automated 3D modelling (Beardsley et al., 1996; Schaf-falitzky and Zisserman, 2002; Brown and Lowe, 2003; Snavely et al., 2006), vehicle tracking (Coifman et al., 1998), and object recognition (Ta et al., 2009). In the field of SHM, feature tracking has been used for global displacement estimation of structures (Yoon et al., 2016; Khuc and Catbas, 2017). Some well-known algorithms for feature detection include Harris-Stephens (Harris and Stephens, 1998), scale invariant feature transform (SIFT) (Lowe, 2004), sped-up robust feature (SURF) (Bay et al., 2006), Shi-Tomasi (Shi, 1994), and features from accelerated segment test (FAST) (Rosten and Drummond, 2005). While any of the above algorithms can be used for feature detection, we have adopted the Shi-Tomasi algorithm due to its robust performance.

In feature detection, an ROI is first selected in the first video frame. The ROI can either be the entire image or a specific region in the image. Feature points in Shi-Tomasi algorithm are selected based on the eigenvalues of following matrix:

$$M = \sum w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{4}$$

where $w(x, y)$ is a Gaussian window function, and I_x and I_y are spatial derivatives of pixel intensity in x and y directions, respectively. The two eigenvalues of M are λ_1 and λ_2. Feature points are determined if the minimum of λ_1 and λ_2 is greater than a cutoff threshold. A large threshold leads to a more stringent requirement hence fewer feature points. Therefore, depending on the application, the density of detected feature points can be tuned by changing the cutoff threshold.

With the feature points determined in the first video frame, their displacements can be tracked throughout the video stream. To this end, the Kanade-Lucas-Tomasi (KLT) tracker (Tomasi and Takeo, 1991; Lucas and Takeo, 1981) is adopted. The KLT tracker computes the movement of feature points between the two adjacent frames and the coordinates of each feature point are stored. KLT also uses image pyramids to robustly estimate larger displacements. Five pyramid levels are used in this study. Moreover, forward backward tracking (Kalal et al., 2010) was used to track feature points from a previous frame to the current frame and then from the same feature back to the previous frame. If the error between forward and back tracking is high, then corresponding points are considered invalid. Other conditions for invalid points include points that fall outside of the image or the computed spatial gradient matrix in its neighborhood is singular. In our study, only robustly tracked feature points are considered for crack detection.

Note that the KLT tracker used here is based on sparse optical flow since it does not compute motion of every pixel of the image. Obtaining dense optical flow is computationally expensive, and it increases the computation cost for the crack detection algorithm, which will be introduced in the next subsection. For accurate feature tracking, light conditions should remain relatively constant between consecutive frames. Since the crack detection algorithm relies on a relatively short video with a length of a few seconds, this condition can easily be satisfied in practice.

Crack Detection Algorithm

The fundamental principle of the proposed fatigue crack detection algorithm is to identify the discontinuities in the structural surface motion induced by fatigue crack opening/closing under live load. Fatigue crack induced surface movements are usually very small, which can easily be disguised by larger global rigid-body movement due to camera motion. With global motion compensated videos, it becomes possible to use the tracked displacement history of feature points to analyze small movements and identify the fatigue crack. To this end, the crack detection algorithm by Kong and Li (2018) is adopted and extended in this study. First, we establish the origin of the coordinate system at the top left corner of the first frame of the global motion compensated video. The location of each feature point throughout the video stream is stored during the feature tracking process. The relative movement of feature point *I* between the first frame and frame *t* is given by:

$$x_r^i(t) = x^i(t) - x^i(1) \text{ and } y_r^i(t) = y^i(t) - y^i(1) \tag{5}$$

where subscript *r* denotes relative displacement, $x_r^i(t)$, and $y_r^i(t)$ are the relative displacements of the *i*th feature point in x and y directions, respectively. Thereafter, the total displacement can be obtained by combining $x_r^i(t)$ and $y_r^i(t)$ as:

$$d^i(t) = \sqrt{x_r^i(t)^2 + y_r^i(t)^2} \tag{6}$$

The displacement with respect to the first frame is computed for all detected feature points. A 3rd degree polynomial trend is computed and removed from the displacement histories to eliminate any inaccuracy introduced during the tracking process. To provide a single-value metric for analyzing the variation of tracked displacements, the standard deviation $S^i$of $d^i(t)$ is adopted to quantify the magnitude of the feature point movements under fatigue loading.

To facilitate efficient analysis of the motion of the entire structural surface within the ROI, an LCR is deployed to group feature points for localized analysis. First, a random feature point is selected as the center for the LCR, and a search is performed to gather features within the LCR. Subsequently, the displacement pattern within the LCR is computed using the coefficient of variation *(CV)* of the standard deviations ($S^1, S^2, \dots, S^i$) of all feature displacements within the LCR. No differential movement pattern is detected if the *CV* is lower than a certain threshold, hence no fatigue crack is detected. However, if the *CV* exceeds the threshold, a fatigue crack is considered to exist within the LCR, hence the center feature point of the LCR is retained and highlighted as a part of the final crack detection result. This process is repeated by choosing another feature point as the center of the LCR until all feature points have been analyzed. The final crack detection result consists of all the highlighted feature points for which differential movement pattern was discovered within the associated LCR. Since these highlighted feature points trace along the fatigue crack, they signal the location and extent of the detected crack in an intuitive way.

The crack detection algorithm by Kong and Li (2018) uses a single predefined threshold value for *CV*. As the criteria for determining differential movement pattern, the threshold of *CV* is a critical parameter. The threshold value depends on several factors such as fatigue load level, distance of the camera from the structural surface, and the amount of crack opening/closing. However, the inspector may not have all the information during the field inspection for selecting the most appropriate threshold value. To alleviate this challenge, we have extended crack detection algorithm in this paper by scanning across a range of threshold values and provide a series of crack detection results associated with those thresholds. Compared to Kong and Li (2018) which uses a single predefined threshold, our crack detection algorithm produces a series of crack detection results corresponding to a range of threshold values. Too-low threshold values will result in many feature points spread over a large portion of the structural surface. However, as the threshold increases, feature points that are not associated with a fatigue crack get filtered out, leaving a smaller group of features clustered in one or multiple groups surrounding the actual crack(s). If no fatigue crack is present, feature points will not be clustered but rather distributed sparsely over the ROI. In practical applications, the inspector can start with a large and coarse threshold range, then adjust the range and resolution by observing the crack detection result. The goal is to capture a series of smooth transitions of the clustering result of feature points, which will be further explained in Section 4.

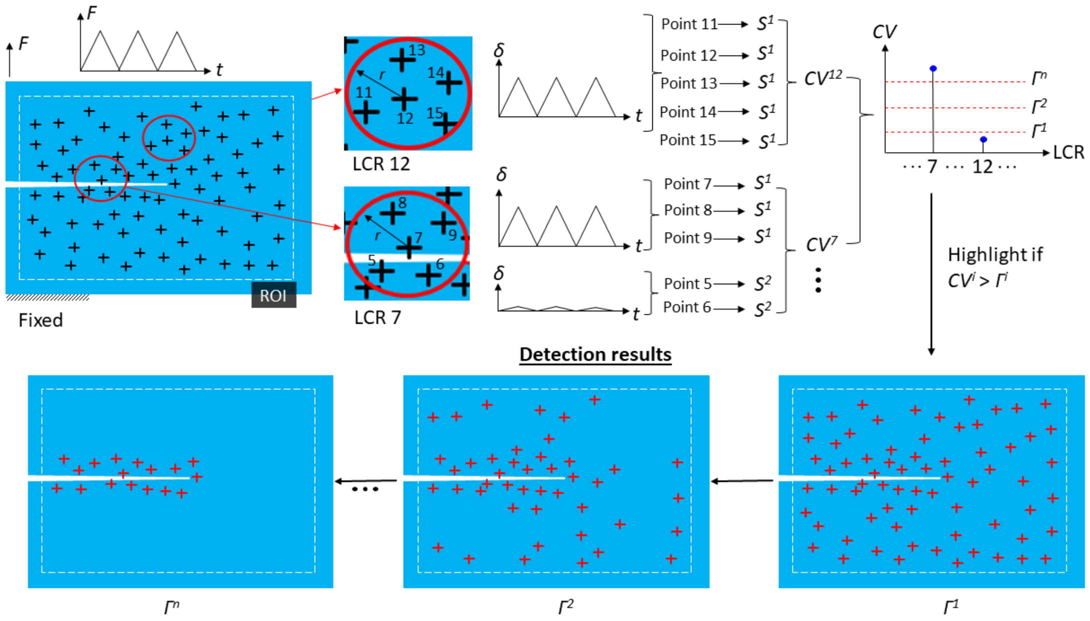


Figure 7 Illustration of the proposed crack detection algorithm using a cracked steel plate under fatigue loading

Figure 7 illustrates the crack detection algorithm using a steel plate with a fatigue crack as an example. A repetitive fatigue load *F* is applied to the top of the plate, and the bottom of the plate is fixed. A ROI is selected that leaves small margins on all four sides of the plate, and feature points are detected using the Shi-Tomasi algorithm. Two LCRs, one over the crack and another one away from the crack, both containing 5 feature points, are considered for demonstration purpose. LCR 12 is located away from the crack, hence feature points 11 to 15 are subjected to rigid body movement, hence the displacements of all feature points $\delta^{11}(t)$ to $\delta^{15}(t)$ are similar under the fatigue load *F*. The standard deviations of feature point movements $\delta^{11}(t)$ to $\delta^{15}(t)$ should have similar magnitudes denoted as S^{1}, resulting in a very small coefficient of variation CV^{12} for LCR 12, where the superscript 12 denotes the center feature point of the LCR. On the contrary, LCR 7 is over the crack, leading to two distinct displacement patterns: the three feature points above the crack share a similar displacement and the two feature points below the crack have a small displacement because of the fixed boundary condition. Therefore, the coefficient of variation is higher for LCR 7, denoted as CV^{7}. With the *CV* for all LCRs computed, a range of threshold values (Γ^{1}-Γ^{n}) is used to filter out the feature points as described previously, generating a series of feature point maps. As the threshold increases, the feature points start to concentrate around the actual fatigue crack, serving as the crack detection result, as shown in Figure 7.

Experimental Setups

The proposed methodology was experimentally validated using two laboratory setups for fatigue crack detection through both 2D and 3D videos. The two test setups are described in this section.

In-plane Fatigue Crack Setup for 2D Video

An experimental test based on a compact, C(T), specimen was performed to validate the proposed approach for GMC of 2D videos and the subsequent in-plane fatigue crack detection. The specimen was loaded in tension to develop an in-plane fatigue crack. Figure 8 shows the specimen and the test setup. The C(T) specimen is made of grade A36 steel with a thickness of 6.35 mm. Other dimensions of the specimen are shown in Figure 8a. A servo-hydraulic uniaxial load frame was used to apply cyclic fatigue loading to the specimen. The specimen was installed in the load frame with two clevises. The top clevis was fixed, and the fatigue load was applied through the bottom clevis. Kong et al. (2017) developed a realistic loading protocol following ASTM E1820-15 (ASTM, 2015) to ensure realistic fatigue crack openings. A constant range of stress intensity factor $\Delta K = 22$ MPa√m and a stress ratio $R = 0.6$ were used. A fatigue crack was initiated and propagated under a loading frequency of 10 Hz. During the loading, ΔK was kept constant by continuously lowering the magnitude of the load cycles as the crack propagated. An adhesive tape measure was affixed to the C(T) specimen to visually observe crack growth over time. For every 1.6 mm of crack growth, ΔF was decreased to maintain the constant ΔK of 22 MPa√m throughout the test. Fatigue crack propagation was stopped at 50.8 mm of crack length measured from the notch of the specimen.

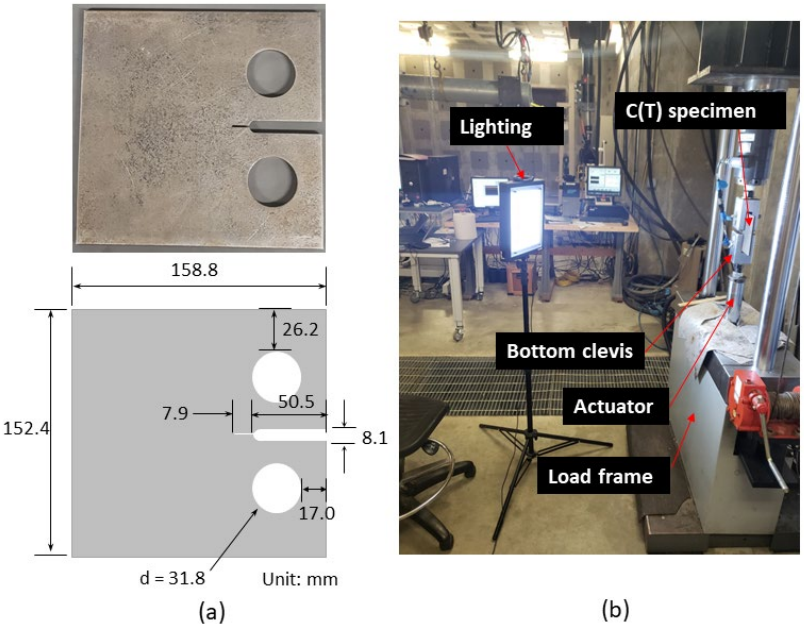


Figure 8 Test setup for 2D videos: (a) C(T) specimen; and (b) loading scheme

After the fatigue crack was developed, a cyclic load ranging from F_{min} = 2.67 kN to F_{max} = 6.23 kN at 1 Hz frequency was applied. An 8-second video was recorded using a hand-held Samsung Galaxy s9 Plus mobile phone. The side view of the experimental setup for in-plane fatigue crack detection using the C(T) specimen is illustrated in Figure 9. Note that the video captures not only the specimen, but also the top and bottom clevises, whose surfaces are not on the same plane as the specimen, leading to two different focal distances Z_1 and Z_2, and hence depth variation within the video scene. To ensure that the recorded video can be considered as a 2D scene, the camera was located roughly 135 cm away from the clevises, which is sufficiently far to make the depth variation negligible. However, in actual field inspections, the clevis conditions do not exist and video can be recorded from a closer distance to the structural surface. The video was recorded at a resolution of 3,840 pixels × 2,160 pixels at 30 frames per second. Supplemental lighting was used while recording due to poor indoor lighting condition, to increase robustness of feature detection.

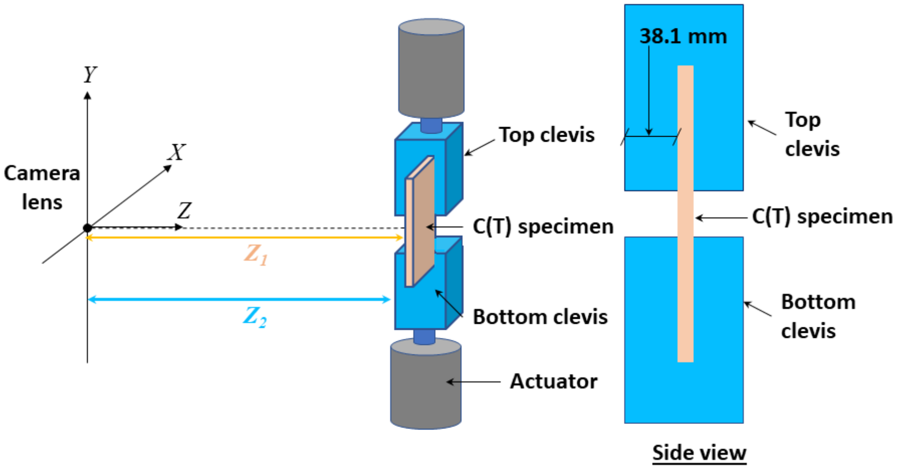


Figure 9 Schematic of the test setup including the camera position

After the fatigue crack was developed, a cyclic load ranging from F_{min} = 2.67 kN to F_{max} = 6.23 kN at 1 Hz frequency was applied. An 8-second video was recorded using a hand-held Samsung Galaxy s9 Plus mobile phone. The side view of the experimental setup for in-plane fatigue crack detection using the C(T) specimen is illustrated in Figure 9. Note that the video captures not only the specimen, but also the top and bottom clevises, whose surfaces are not on the same plane as the specimen, leading to two different focal distances Z_1 and Z_2, and hence depth variation within the video scene. To ensure that the recorded video can be considered as a 2D scene, the camera was located roughly 135 cm away from the clevises, which is sufficiently far to make the depth variation negligible. However, in actual field inspections, the clevis conditions do not exist and video can be recorded from a closer distance to the structural surface. The video was recorded at a resolution of 3,840 pixels × 2,160 pixels at 30 frames per second. Supplemental lighting was used while recording due to poor indoor lighting condition, to increase robustness of feature detection.

Out-of-plane Fatigue Crack Setup for 3D Video

For steel girder bridges, out-of-plane distortion-induced fatigue cracks commonly occur in web-gap regions where the girder web, flange, and the crossframe connection plate meet. As shown in Figure 10, a half-scale girder-to-crossframe subassemblage was used for validating the proposed methodology for detecting out-of-plane fatigue cracks using 3D videos. The girder has a length of 2,845 mm and a depth of 917 mm, and the web thickness is 9.5 mm. The bottom flange of the steel girder was restrained by steel channels connected to the laboratory strong floor to approximate the condition of a real bridge restrained by the concrete deck. A crossframe was attached to a stiffener at the center of the girder web through a connection plate. The connection plate was only welded to the web of the girder to produce the web-gap region as shown in Figure 9. When load is applied at the far end of the crossframe, it simulates traffic load that causes differential vertical displacement between adjacent girders, resulting in out-of-plane fatigue cracking in the web-gap region. As illustrated in Figure 11, a complex fatigue crack with three distinct branches labeled as A, B, and C, respectively, was generated with a fatigue load with a range of 2.2 kN to 25.5 kN. In particular, branch A was first generated within the weld between the connection plate and web after 21,000 load cycles (Dellenbaugh et al., 2020). After that, at around 1,700,000 cycles, branch A crack continued to grow and bifurcated into two separate branches, B and C, into the web (Al-Salih et al., 2021). In this study, for video collection, a cyclic fatigue load of 0.89 kN to 17.8 kN was applied at 0.5 Hz to the far end of the crossframe by a hydraulic actuator. It was noticed that the crack movement under fatigue loading was dominated by branch A, as the propagation of the branched cracks reduced the driving force at the two crack fronts. A short, 5 second video was recorded using a head-mount Microsoft HoloLens 2 (*HL2*) as shown in Figure 11. The video resolution was 1,920 × 1,080 pixels with a framerate of 30 frames per second. Similar to the 2D case, supplemental LED lighting was used due to the poor indoor lighting in the laboratory.

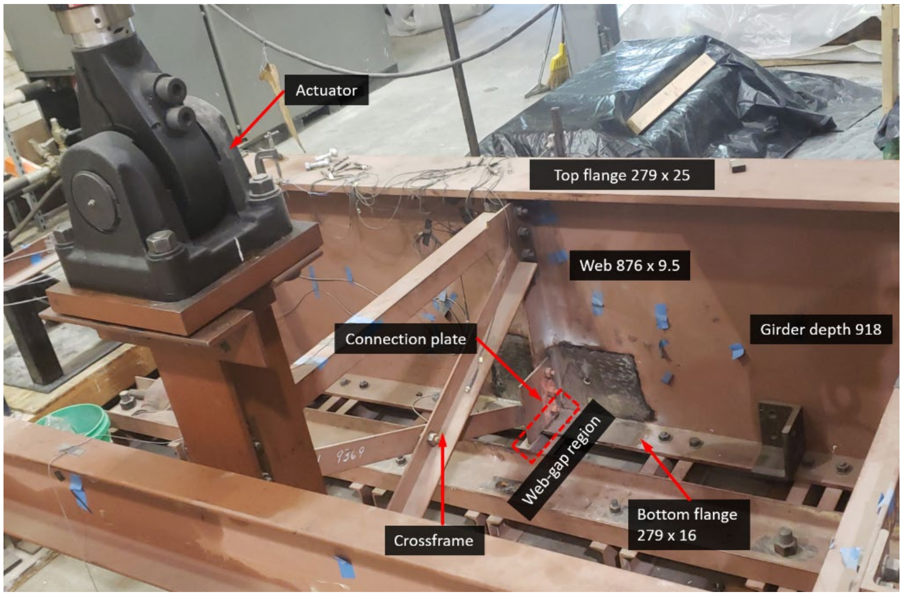


Figure 10 Bridge girder test setup for 3D videos (unit: mm)

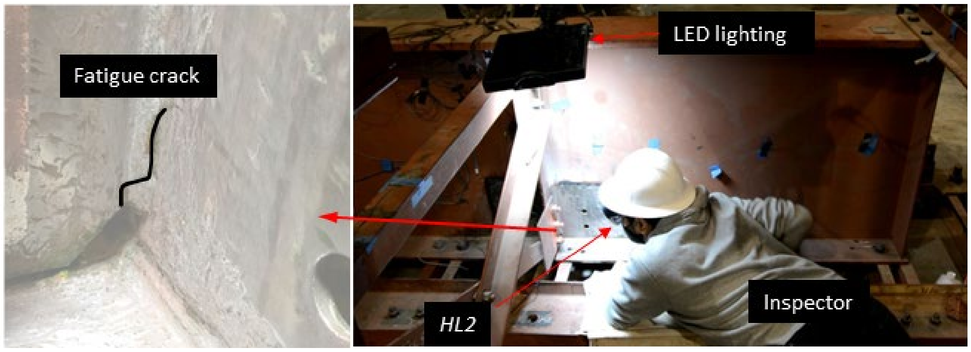


Figure 11 Out-of-plane fatigue crack and video recording using the *HL2* headset

Results and Discussions

In-plane Fatigue Crack Detection using 2D Video

Result Based on Unstable Video

To demonstrate the necessity of GMC for 2D videos, the raw unstable video recorded by the hand-held smartphone camera was first applied for fatigue crack detection. To reduce computation, the recorded 2D video for the C(T) specimen was first reduced to a resolution of 1,980 × 1,080 pixels. Figure 12a shows the initial frame of the video and the selected ROI. Generally, ROI can be selected by leaving a 10% to 15% margin from all four sides of the frame as boundary regions can fall outside of the camera field of view due to camera motion. Here, ROI is selected to remain inside of the C(T) specimen to avoid capturing the differential motion between the edge of plate and the background. A total of 3,000 feature points were detected using the Shi-Tomasi feature detection algorithm, shown in Figure 12b. The diameter of LCR mainly depends on the camera resolution and the distance of camera to the structural surface. It was found an LCR with a radius of 35 pixels is adequate for videos recorded from a 10 to 15 cm distance from the structural surface with a standard HD resolution of 1,920 × 1,080 pixels. More details on the impact of LCR radius can be found in Kong and Li (2018). In this study, the radius of LCR for the C(T) specimen video was selected as 30 pixels. Figure 12 shows a series of crack detection results over the selected range of threshold values. The actual fatigue crack is highlighted by a solid black line for reference. The first result with a low threshold value of 0.001 is shown in Figure 13a. Because of very low threshold, almost all features detected in the initial frame passed the threshold. Next, results based on two larger threshold values, 0.005 and 0.009, are shown in Figure 13b and Figure 13c, respectively. These threshold values are still below the noise level of the tracked feature point displacement, hence providing no obvious improvement. Starting from the threshold value of 0.013, as shown in Figure 13d, more features points away from the crack are removed. However, the detected point cloud spreads sparsely over a wide area. Further increasing the threshold to 0.017 and 0.021, as shown in Figure 13e and Figure 13f, the point cloud density is further reduced, and some of them surround the fatigue crack. However, due to the global camera motion present in the unstable video, the detection results also contain many feature points sparsely distributed over the entire ROI, which are considered as false positive results, making the interpretation challenging. Upon further increasing the threshold value, all feature points are filtered out. Hence, without removing the global motion due to the moving camera, the algorithm was not successful in detecting the in-plane fatigue crack.

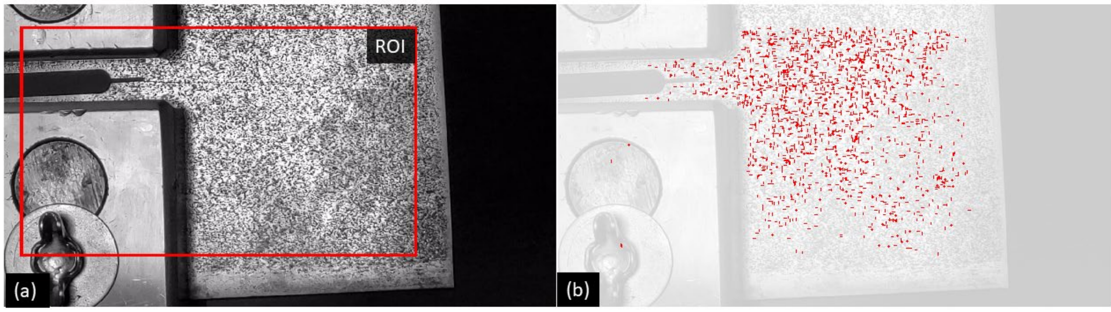


Figure 12 The initial frame of the 2D video with the selected ROI (a) and all feature points detected by the Shi-Tomasi algorithm (b). Note that the brightness of image in (b) is enhanced to highlight the feature points

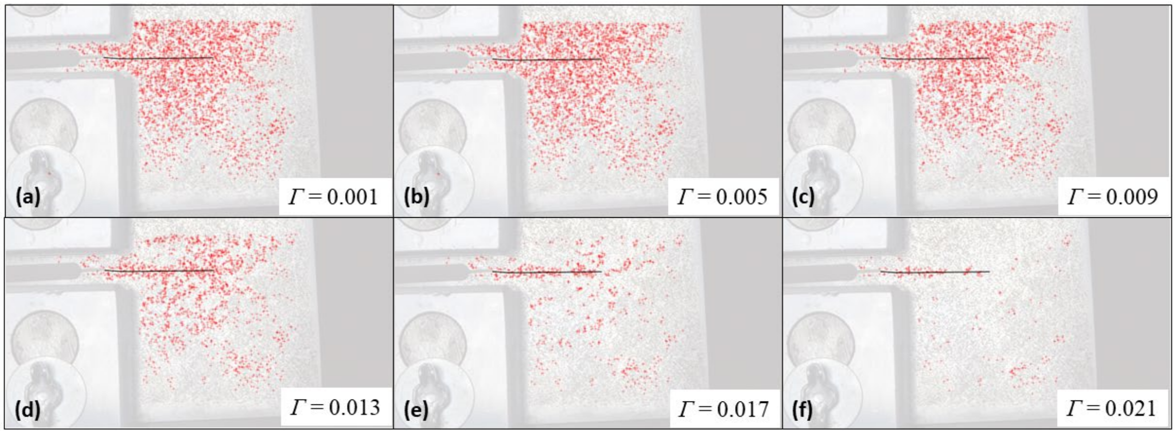


Figure 13 In-plane fatigue crack detection result of a C(T) specimen using the raw unstable 2D video

Result Based on Global Motion Compensated Video

In this study, to apply the 2D GMC algorithm, eight pyramid levels are adopted for the standard HD resolution of 1,980 × 1,080 pixels. At each pyramid level, 2 iterations are performed before propagating motion parameters to the next level. In particular, we first estimate affine parameters between the target image and the reference image at the lowest resolution based on Equation 2. Subsequently, the target image is warped using the estimated affine model. Then, the affine parameters are estimated again between the warped image and the reference image. The newly estimated motion parameters are then used to warp the target image at the next level of pyramid. The process was performed for all 247 pairs of video frames in the 8 second video with 248 image frames. In each pair, the first frame was selected as the reference frame. To demonstrate the effectiveness of the 2D GMC algorithm, Figure 14 compares the mean of all frames in the unstable video with that of the global motion compensated video. Due to the presence of camera motion, the mean of all frames in the unstable video (Figure 14a) is very blurry. However, since the GMC algorithm aligns the coordinates of all frames, the mean of all frames in the global motion compensated video (Figure 14b) is much clearer. In other words, the tracked feature point displacement will be less contaminated by the camera motion, reflecting the true differential motion pattern due to the fatigue cracking.

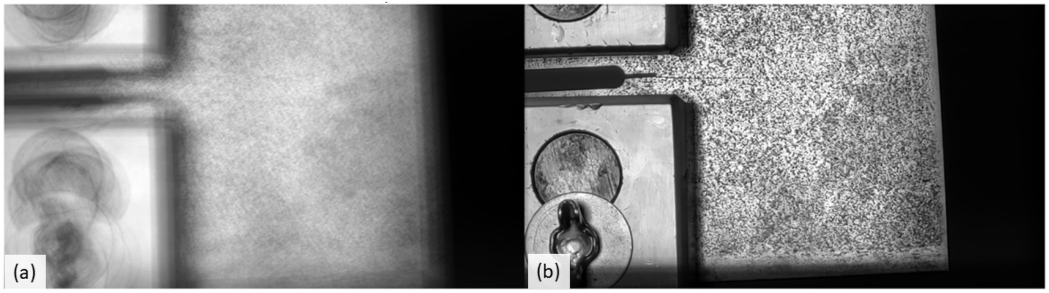


Figure 14 Comparison between the mean of all frames in the 2D video before (a) and after (b) the GMC

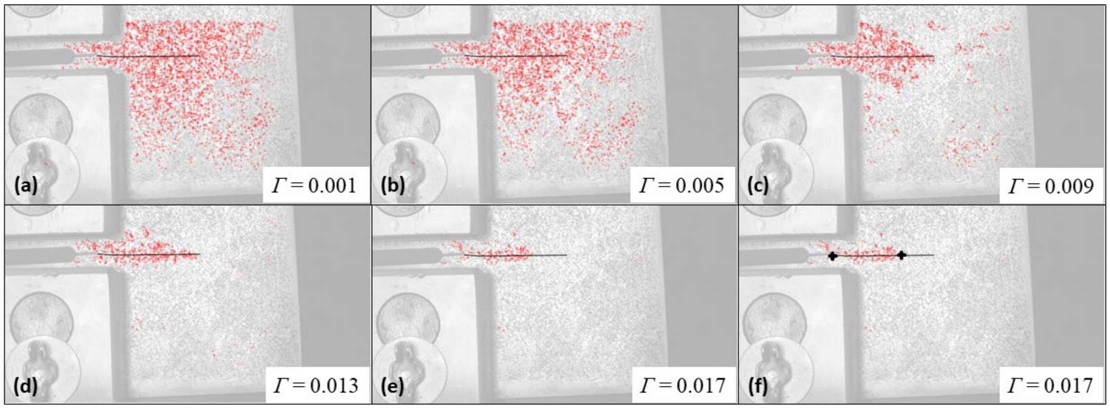


Figure 15 In-plane fatigue crack detection result of a C(T) specimen using the global motion compensated 2D video

With the global motion compensated video, the crack detection algorithm was applied to detect the fatigue crack using the same parameters (ROI, LCR, and thresholds) as the raw unstable video case, and the results are listed in Figure 15. Similar to the unstable video case, the first and second threshold values are too small to filter out feature points in uncracked regions. However, unlike the unstable video, the third threshold value ($\Gamma = 0.009$) is now able to remove majority of the feature points far away from fatigue crack, as shown in Figure 15c. For $\Gamma = 0.013$, the improvement over the unstable video is much more pronounced, as the majority of detected feature points now form a single cluster surrounding the fatigue crack. However, some sparse features isolated in small groups are still present, hence it is not selected as the final result. By further increasing the threshold to 0.017, as shown in Figure 15e, the detected feature points form a single group around the crack and no significant isolated feature points are detected. Therefore, it is selected as the final crack detection result.

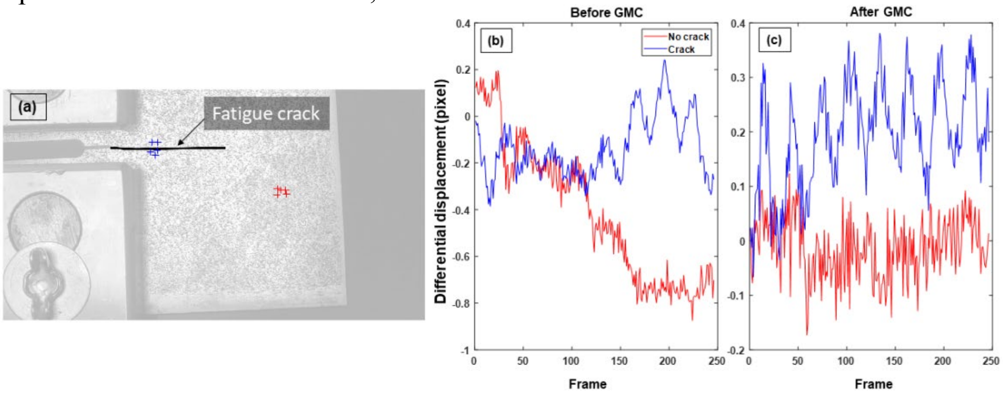


Figure 16 Surface motion tracking of the 2D video: (a) feature points for cracked and uncracked regions; (b) differential displacement before GMC; (c) differential displacement after GMC

To further understand the impact of camera movement and GMC on the effectiveness of the crack detection algorithm, surface motions are extracted from both cracked and uncracked regions of the C(T) specimen for illustration. The two selected feature groups on the C(T) specimen are shown in Figure 16a, and differential displacements of two feature point pairs, one from the cracked and another from uncracked region, are shown in Figure 16b and Figure 16c for the raw unstable video and the motion compensated video, respectively. As shown in Figure 16b, for the unstable video, the displacements of feature points in both cracked and uncracked regions are dominated by the global camera movement, resulting in higher differential movement in the uncracked region compared to the cracked region. This would further lead to false positive crack detection using the unstable video. Hence, the tracked surface movement from the unstable video is not able to correctly distinguish movement patterns between regions with and without a fatigue crack. On the contrary, after the GMC, the feature point displacements are dominated by the surface motion induced by fatigue crack opening and closing. As a result, as depicted in Figure 16c, the differential motion of the feature points over the cracked region exhibits a much larger amplitude than the that from the uncracked region. In addition, the GMC algorithm demonstrated significant improvement in the quality of tracking the fatigue crack induced surface motion, as Figure 16c clearly shows the eight fatigue load cycles.

Quantification of the Detected Crack

Once the fatigue crack is detected and localized, the length of the detected crack can be estimated and compared with the actual crack length. The reason to quantify the length of the detected crack is twofold. First, crack length is a critical part of standard inspection reports for steel bridges. Second, compared to crack width, which is dependent on the load level, crack length at any given stage is independent from the load level. As show in Figure 15f, the two feature points located at the left and right extreme positions are located from the detected feature point cluster presented in Figure 15e. The distance between these two feature points is 364 pixels. This pixel distance is then converted to the physical unit based on the plate dimension from the notch to the end of the specimen, which is 108.3 mm and 1,194 pixels, resulting in a scaling factor of 0.091mm per pixel. Using the scaling factor, the detected crack length is determined as 33.5 mm. Since the actual crack length is measured as 50 mm, 67% of the fatigue crack is detected using the proposed methodology. As noted by Kong et al. (2018), the level of crack opening/closing decreases significantly near the crack tip, making it challenging to locate the crack tip using the proposed methodology.

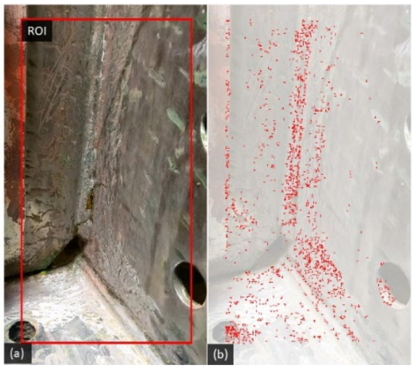


Figure 17 The initial frame of the 3D video with ROI (a): and all feature points detected by the Shi-Tomasi algorithm (b). Note that the brightness of image in (b) is enhanced to highlight the feature points

Crack Detection Results for Out-of-plane Fatigue Crack

Result Based on Unstable Video

Again, to demonstrate the necessity and effectiveness of the proposed 3D GMC algorithm presented in Section 2.2.2, the raw unstable 3D video recorded using the head-mounted HL2 for the out-of-plane fatigue crack was first processed for crack detection. Figure 16a shows the initial frame of the video and the selected ROI. A total of 3,000 feature points were selected using the Shi-Tomasi feature detection algorithm as shown in Figure 17b. The radius of LCR was selected as 30 pixels. The crack detection results for a range of threshold values are presented in Figure 18. In Figure 18a, most features are retained due to the very low threshold value of 0.01. Then, the threshold value was gradually increased by an increment of 0.005. Results in Figure 18b and Figure 18c are similar to Figure 18a, with minimum feature points filtered out. As threshold value kept increasing, feature points were eliminated from the results. However, the feature points did not cluster around the actual fatigue crack. Thus, crack detection using the raw unstable 3D video is considered unsuccessful.

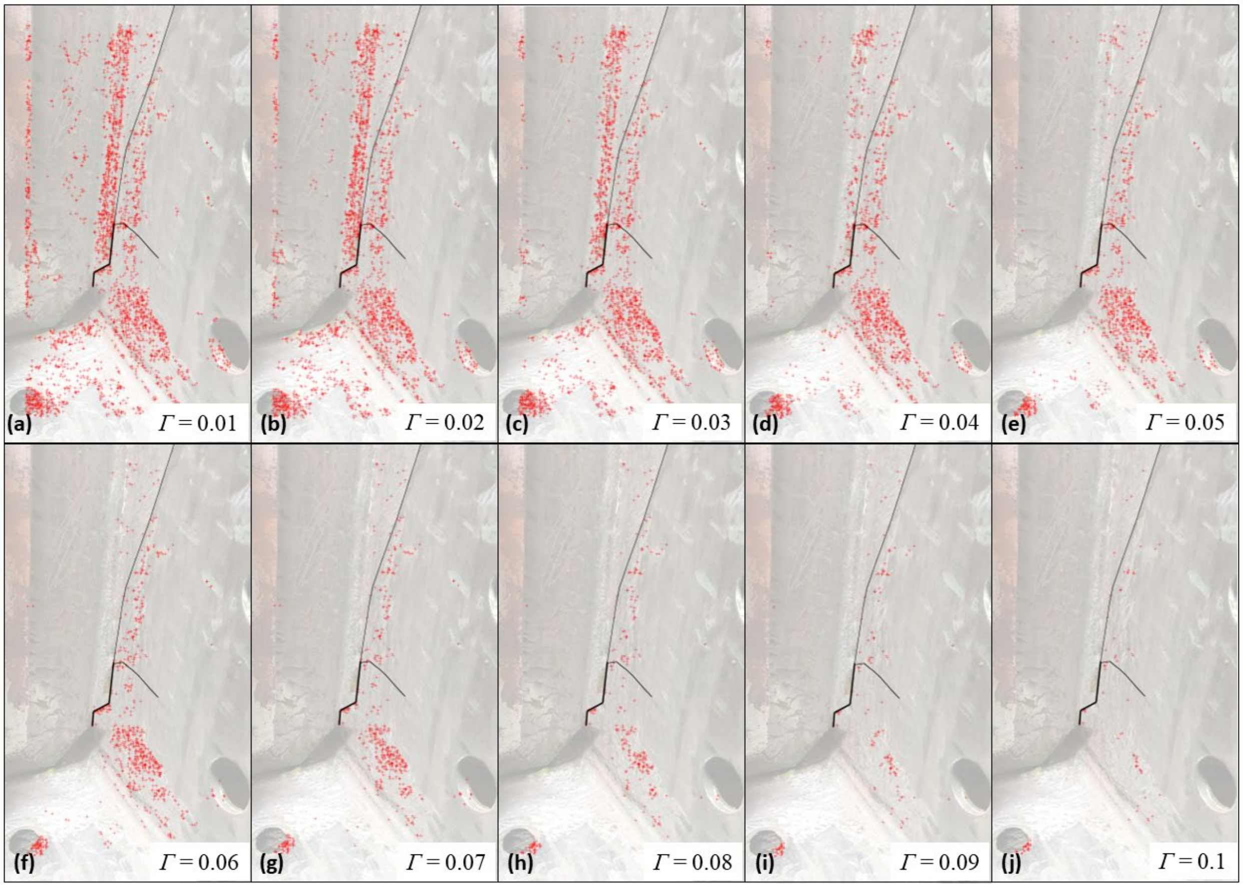


Figure 18 Out-of-plane fatigue crack detection result of the bridge girder specimen using unstable 3D video

Result Based on Global Motion Compensated Video

The proposed 3D GMC approach based on the extended bundled camera paths was applied to the recorded unstable video. We divided the video frame into a 10×10 grid. 20 iterations were applied to minimize the objective function in Equation (3) to obtain the optimized and smoothed camera paths. In the second stage, a high-pass filter with a cutoff frequency of 0.11 Hz was used to remove the low-frequency global motion. Because the 3D GMC is computationally expensive, the video resolution was reduced from 1,980 × 1,080 pixels to 1,280 × 720 pixels. To illustrate the result of the proposed GMC algorithm for 3D videos, Figure 19 compares all 100 translation parameters associated with the grid cells of the 3D video before and after the GMC. In Figure 18, three sets of camera paths in both x and y directions are illustrated, including the camera paths from the original raw video (Figures 18a and 18d), the optimized and smoothed camera paths (Figure 19b and Figure 19e), and the final global motion compensated camera paths after high-pass filtering (Figure 19c and Figure 19f). As shown in the figures, the translation movements in the x direction of the original camera paths reach up to 92 pixels, which then reduce to 77 pixels in the smoothed paths. Similarly, in the y direction, translation parameters b_2 reach 51 pixels in the original camera paths, which then reduce to 45 pixels in the smoothed paths. Even though the bundled camera paths approach removed the high frequency components from the camera paths, low frequency translation is still present. After filtering the mean camera path and subtracting the filtered component from each individual camera path, the new translation parameters reduced to less than 31 pixels (Figure 18c) in the x direction and 12 pixels (Figure 18f) in the y direction. The homographies obtained using the new translation parameters represent not only the smoothed camera paths but also global motion compensated transformation matrices. These transformation matrices are then used to warp the video frames to obtain the motion compensated 3D video. Figure 19 compares the mean of all video frames before and after the GMC. Figure 19 (a) shows the blurry mean image due to the camera motion that leads to misaligned video frames. After applying the proposed 3D GMC approach, although still some residual camera motion remains, the mean of motion compensated video frames became much clearer.

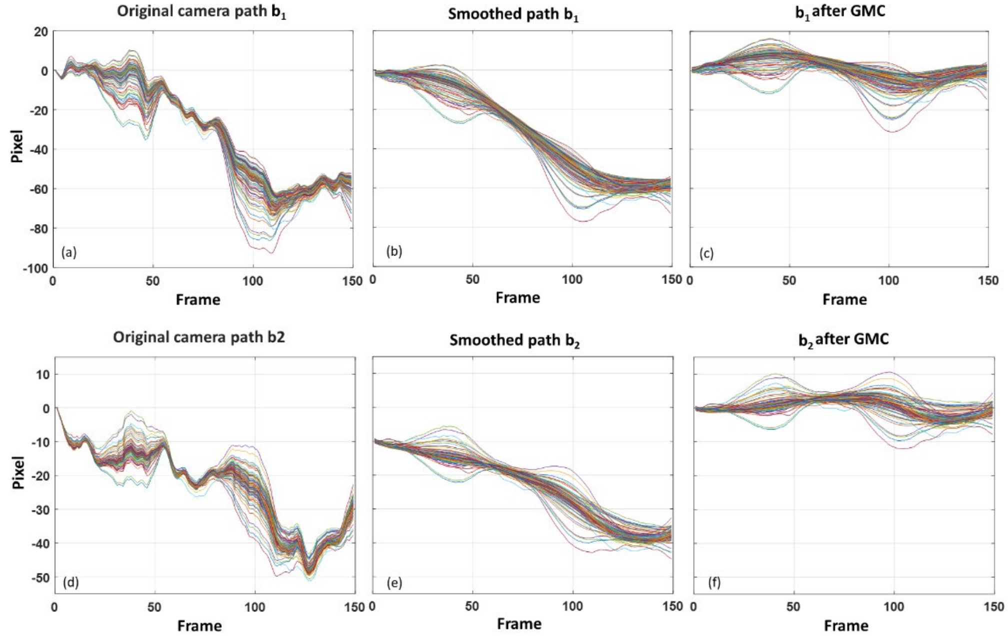


Figure 19 Bundled camera paths of the two translation parameters

Subsequently, the motion compensated video was processed by the proposed crack detection approach. The ROI was set the same as in Figure 16a. The radius of LCR was selected to be 15 pixels because of the reduced resolution, and a total of 3,000 feature points were detected using the Shi-Tomasi feature detection algorithm. The crack detection results are shown in Figure 20. As the threshold values increase, feature points start to be filtered out from the crack detection result. Compared with Figures 20a and 20b, a significant reduction of feature points can be seen in Figure 20c. By continuing the increase of threshold, as shown in Figure 20h, feature points start to cluster in three different groups. The final result shown in Figure 20j was selected when the feature points are grouped in a single cluster, showing the actual fatigue crack location marked by the solid line. The result demonstrates the effectiveness of the proposed fatigue crack detection methodology using videos that contain 3D scenes. As mentioned previously, branch A dominated the crack movement as the propagation of the branched cracks reduced the driving force at the two crack fronts. As a result, the fatigue crack branches into the girder web were not detected due to reduced stress demand in branched cracks.

To further explore the impact of GMC on surface motion tracking of the 3D video, differential displacement patterns of feature points associated with two LCRs extracted from the 3D video of the bridge girder are compared in Figure 21 before and after GMC. The fatigue crack is highlighted by a solid black line, while the feature points over the crack are shown in blue crosses, and feature points away from the crack are displayed in red crosses. Figures 21b and 21c present the differential displacement patterns of feature points from the 3D video before and after GMC, respectively. As can be seen, before motion compensation, the differential motion of feature point pair in the uncracked region is much higher than the cracked region. This explains the failure in crack detection using the unstable 3D video. On the other hand, the differential displacement of feature points over the cracked region in the global motion compensated 3D video clearly shows a larger amplitude corresponding to surface motion due to crack opening and closing than that of the uncracked region. The ability to uncover the differential motion pattern despite the global camera movement is the main contribution of the proposed methodology in achieving robust detection of fatigue cracks using a moving camera.

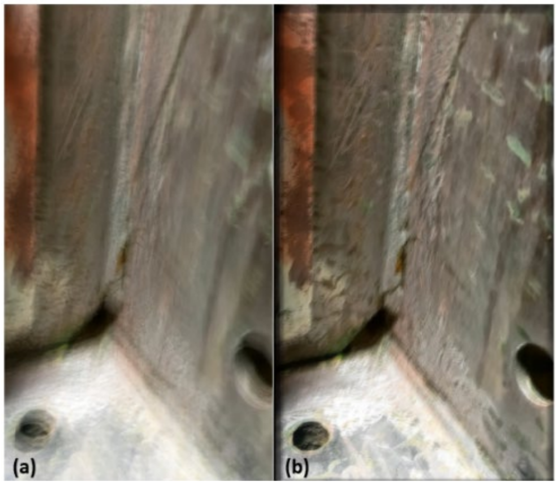


Figure 20 Mean of all frames in the 3D video before (a) and after (b) GMC

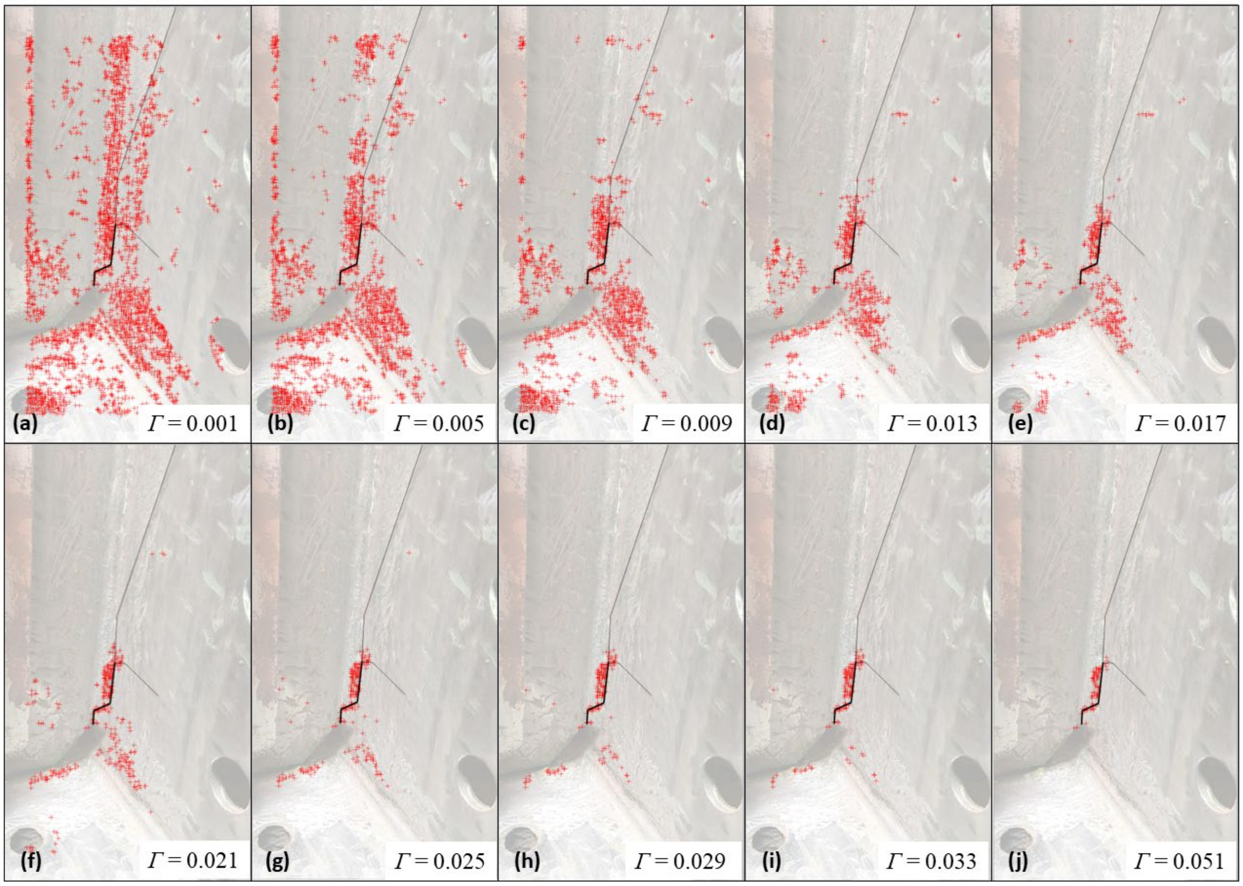


Figure 21 Out-of-plane fatigue crack detection result of the bridge girder specimen using the global motion compensated 3D video

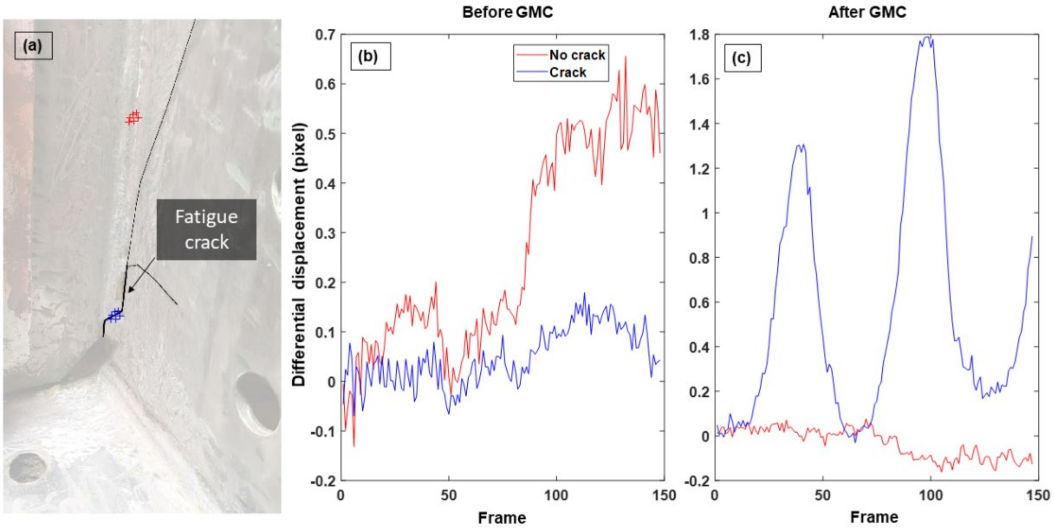


Figure 22 Surface motion tracking of the 3D video: (a) feature points for cracked and uncracked regions; (b) differential displacement before GMC; (c) differential displacement after GMC

Computation Time of the Proposed GMC Algorithms

The computation time for GMC of the 2D video is around 274 seconds (4.5 minutes) on a machine with an intel i9 processor and 64 GB of RAM. The 2D video contains 248 frames with standard HD resolution of 1980 × 1080 pixels making computation cost about 1.1 seconds per video frame. For GMC of the 3D video, the computation time is around 261 seconds (4.4 minutes) on the same machine. The 3D video has 149 frames of standard HD resolution, which was reduced to 1280 × 720 pixels for GMC. Hence, the computation cost is about 1.75 seconds per video frame. The proposed vision-based approach can be utilized with mixed reality devices to enable human-machine collaboration to help inspectors have more accurate visual inspection and expedite the decision-making process. The achieve this goal, future developments will focus on further reducing the computation time of the algorithm to enable near real-time result delivery and decision making.

Distance-based Fatigue Crack Detection Algorithm

This section describes the proposed distance-based methodology for vision-based fatigue crack detection on a structural surface. Figure 23 demonstrates the concept and steps involved in the proposed method. In the first step, a short video of a fatigue crack prone region in the structure is recorded under several fatigue load cycles similar to the displacement-based approach. Videos can be recorded from moving platforms such as UAVs, mixed reality headset, etc. The second step is to detect salient feature points within the ROI in the first video frame and track their locations throughout the video (Figure 23b). Unlike the displacement-based approach, GMC is not required before tracking the location of the surface feature points.

Once locations of feature points are tracked, features are grouped in LCR. The distances between unique feature point pairs within LCR are computed. Distance of feature point pair remain unchanged if there is no crack between features. However, the distance between feature pair changes when fatigue crack present between them open and close under fatigue loading. By scanning all unique feature point pairs within ROI using the LCR and detecting differential movement patterns that exceeds a threshold value, are highlighted as highlighted and shown as detected crack. In the end, collection of all highlighted feature point pairs shows location of the fatigue crack. The details of each component are provided in the following subsections.

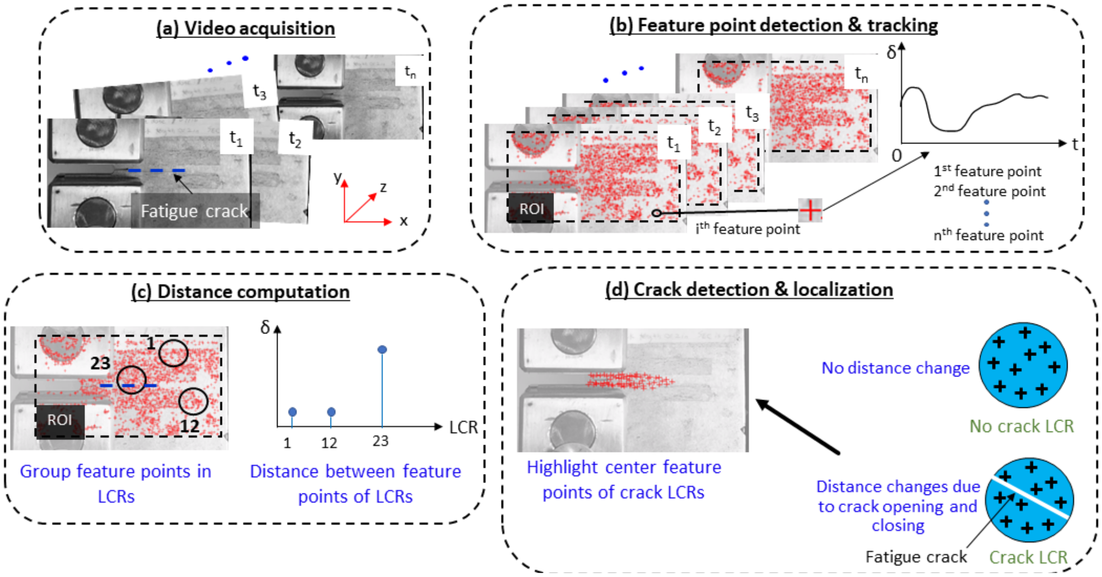

(a) Video acquisition
t1
t2
t3
tn
Fatigue crack
y
z
x
(b) Feature point detection & tracking
t1
t2
t3
tn
ROI
ith feature point
δ
t
0
1st feature point
2nd feature point
nth feature point
(c) Distance computation
1
23
12
ROI
δ
LCR
1
12
23
Group feature points in LCRs
Distance between feature points of LCRs
(d) Crack detection & localization
Highlight center feature points of crack LCRs
No distance change
No crack LCR
Distance changes due to crack opening and closing
Fatigue crack
Crack LCR

Figure 23 Overview of the distance-based fatigue crack detection method

Crack Detection Algorithm

The crack detection algorithm analyzes the structural surface motion for discontinuities caused by fatigue crack opening and closing under live load. Fatigue crack movement is usually very small; hence, the surrounding parts of a structural surface share the same global rigid movement. First, we establish the origin of the coordinate system at the top left corner of the image. Hence, the bottom right corner of the video frame would be the last coordinates of the system. The location of i^{th} feature point of t^{th} frame is denoted by $P_i^t = (x_i^t, y_i^t)$ where x_i^t and y_i^t are x and y coordinate, respectively. The location of each feature point throughout the video stream is stored during the feature tracking process. The distance between the ith feature point and jth feature point of the tth frame is denoted as $D_{i,j}^t$ and can be computed by $\sqrt{\left(x_i^t - x_j^t\right)^2 + \left(y_i^t - y_j^t\right)^2}$.

To illustrate the displacement-base crack detection algorithm, consider four video frames from a video stream as shown in Figure 24. The top left corner has coordinate (0,0) as the origin and the bottom right corner has coordinate (1920, 1080) corresponding to the video resolution. The field of view of the camera is shown by the grey rectangle, which covers a steel plate with a fatigue crack shown by the blue rectangle. In the first frame, the top left corner of the steel plate is at the origin of the coordinate system. The feature points within the ROI are shown by plus marks. Consider two LCRs, including LCR 1 located on the fatigue crack and LCR 2 located away from the fatigue crack. LCR 1 has a total of 7 feature points, with feature point 1 located at the center of the LCR. Feature points 1 to 4 are below the fatigue crack and 5 to 7 are above the fatigue crack. The distance between feature point 1 and feature points 2 to 7, denoted by $D_{1,2}^1, D_{1,3}^1, \dots, D_{1,7}^1,$ respectively, are depicted next to the enlarged view of the LCRs. LCR 2 has a total of 6 feature points with feature point 10 located at the center of the LCR. The distance between feature 10 and feature points 11 to 15 are denoted as $D_{10,11}^1, D_{10,12}^1, \dots, D_{10,15}^1,$ respectively.

In the second video frame, the camera has moved during the videotaping; as a result the steel plate has moved to the right within the camera field of view. In this frame, fatigue crack has not opened due to the lack of live load. The distance of feature point pairs in LCR 1 and LCR 2 are computed and shown next to the enlarged LCR views. The distance between feature point pairs within LCR in the first frame and the second frame remains virtually identical, i.e. $D_{1,2}^1 \approx D_{1,2}^2$,…, $D_{10,15}^1 \approx D_{10,15}^2$, because feature points within a local vicinity move in an almost identical manner under the global camera motion. Even though the coordinates of feature points have changed due to the camera motion, their distances have remained the same. This is true for all the feature points located within the LCR.

In the third video frame, the camera has moved further, and the fatigue crack has opened under the live load. The steel plate has moved left and down in the camera field of view resulting in a large change of coordinates of feature points compared to frame 2. LCR 1 is located on the crack, and as presented in the enlarged view, the distances between feature point pairs 1-5, 1-6, and 1-7, $D_{1,5}^{3}$, $D_{1,6}^{3}$, $D_{1,7}^{3}$ has increased due to the crack opening. However, the distance between the remaining feature pairs $D_{1,2}^{3}$, $D_{1,3}^{3}$, $D_{1,4}^{3}$, and $D_{10,11}^{3}$ to $D_{10,15}^{3}$ continued to remain unchanged under the camera motion as well as fatigue crack opening. As illustrated in the figure, the increased distances between feature point pairs 1-5, 1-6, and 1-7 indicate that LCR 1 has differential motion compared to LCR 2.

In the fourth frame, the steel plate has moved up in the camera field of view and the fatigue crack has closed. The distances between feature point pairs are presented in the figure. $D_{10,11}^{4}$ to $D_{10,15}^{4}$ remain the same compared with $D_{10,11}^{3}$ to $D_{10,15}^{3}$ as the camera movement imposes the same motion to the feature points. However, $D_{1,5}^{4}$ to $D_{1,7}^{4}$ are different compared with $D_{1,5}^{3}$ to $D_{1,7}^{3}$ due to the crack closure. Moreover, $D_{1,2}^{4}$ to $D_{1,4}^{4}$ remain unchanged compared to $D_{1,2}^{3}$ to $D_{1,4}^{3}$. While LCR 1 demonstrates two distinct distance patterns due the differential motions caused by the crack opening and closing under live load, LCR 2 observes only one uniform distance pattern due to the lack of crack movement. For both LCRs, the rigid body motion imposed by the camera movement does not affect the distance pattern. As a result, camera motion compensation is not necessary when distance is employed as the criteria.

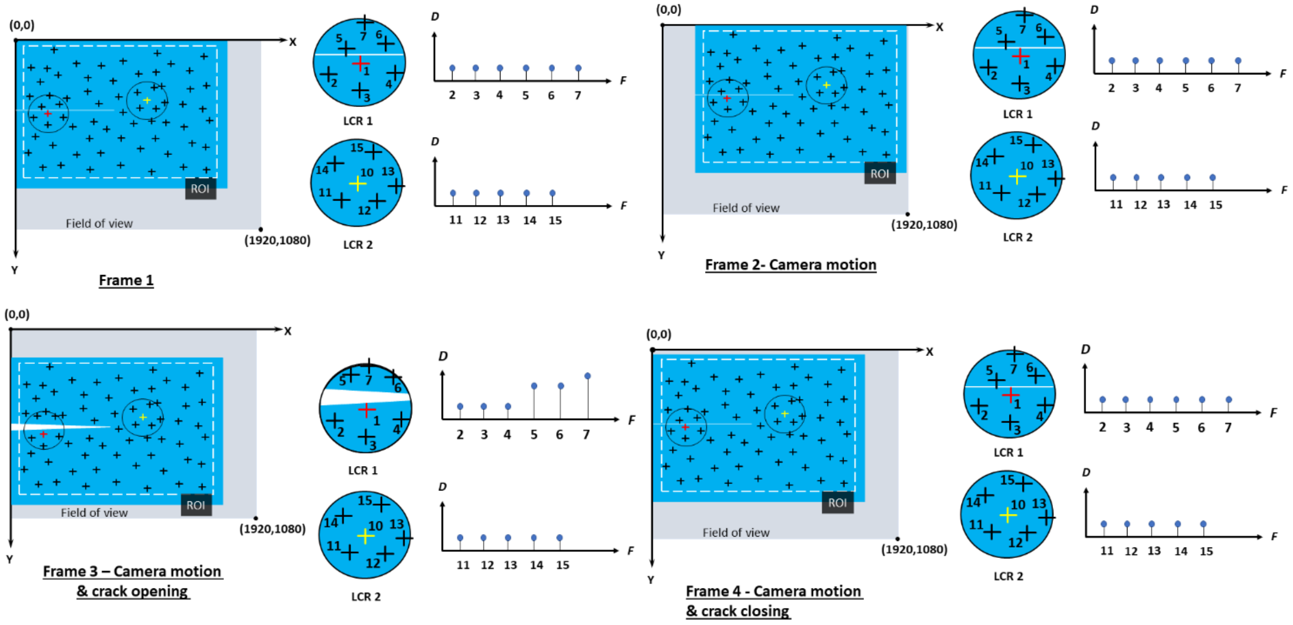


Figure 24: Crack detection algorithm

For efficient analysis of the surface motion through distance, a random feature point within the ROI is selected as the center of the first LCR, and a search is performed to gather feature points within the LCR. After that, unique feature point pairs are determined, and their distance histories are computed through all the video frames. A 3rd-degree polynomial trend is computed and removed from the distance history to eliminate any inaccuracy introduced during the tracking process. The distance histories of feature point pairs are then analyzed to uncover differential patterns caused by fatigue crack movements. To provide a single-value metric for analyzing the variation of distance, the standard deviation $S_{i,j}$ is adopted to quantify the magnitude of the feature point movements under fatigue loading, where subscripts *i* and *j* indicate the ith and jth feature point, respectively. If the standard deviation $S_{i,j}$ is higher than a threshold value, then that feature point pair is highlighted as part of the fatigue crack detection result. This process is repeated by choosing another feature point as the center of the LCR until all feature points have been analyzed. The final crack detection result consists of all the highlighted feature point pairs for which differential movement pattern was discovered within the associated LCR. Since these highlighted feature points trace along the fatigue crack, they signal the location and extent of the detected crack in an intuitive way.

Quantification of Detected Fatigue Crack

There are several ways to quantify the accuracy of prediction results. Image classification models assign each image with a categorical label, and it is then compared with the true categorical label. In computer vision, most segmentation tasks use intersection over union (IOU) as a metric to measure goodness of fit of the model. Vision based deep learning model for semantic segmentation assigns label to each pixel of the image. These labels of each pixel are then compared to labels of the corresponding ground truth pixels. Our vision-based approach uses salient feature points to analyze differential surface motion, and features surrounding the crack are highlighted as the detected crack, which can be considered as segmenting the area associated with the detected features as the crack region. The crack detection result has sparse data points rather than a continuous area labelled as the crack region. Moreover, the width of the cluster of detected feature points depends on the radius of the LCR. To provide a quantitative measure of the crack detection result, a new approach is developed to quantify the predicted sparse data points using IOU. The proposed method consists of three major steps: 1) clustering the detected feature points and obtaining the boundaries, 2) determining the boundary of the ground truth, and 3) evaluating the result through IOU. Each step is explained in detail as follows.

Clustering

Clustering is a process of grouping data points based on similarities. Data points in a group are comparatively more similar to data points in another group. To quantify similarity or distinction, a dissimilarity measure (or distance metric) specific to the domain of application and data set is used. There are several clustering algorithms such as K-means, gaussian mixture models (GMM), hierarchical clustering, density-based spatial clustering algorithm (DBSCAN), and spectral clustering. Hierarchical clustering and K-means are distance-based clustering algorithms. DBSCAN is based on density of regions, and GMM is based on mixture of gaussian distribution. K-means and GMM require manual input such as the number of clusters while hierarchical clustering provides multiple clusters. On the other hand, DBSCAN clusters the detected feature points based on density, and it does not require any manual input hence is adopted in this study.

DBSCAN requires two key parameters: 1) epsilon (ε), and 2) minimum number of points (*MinPts*). ε is the neighborhood distance that determines if a point is in the cluster or not. For example, a point is 11 units away from the cluster *K*. If ε is 11 or less, then the point is considered in the cluster *K*. Otherwise the point is considered outside of the cluster *K*. The second parameter *MinPts* decides how many points are required to form a cluster.

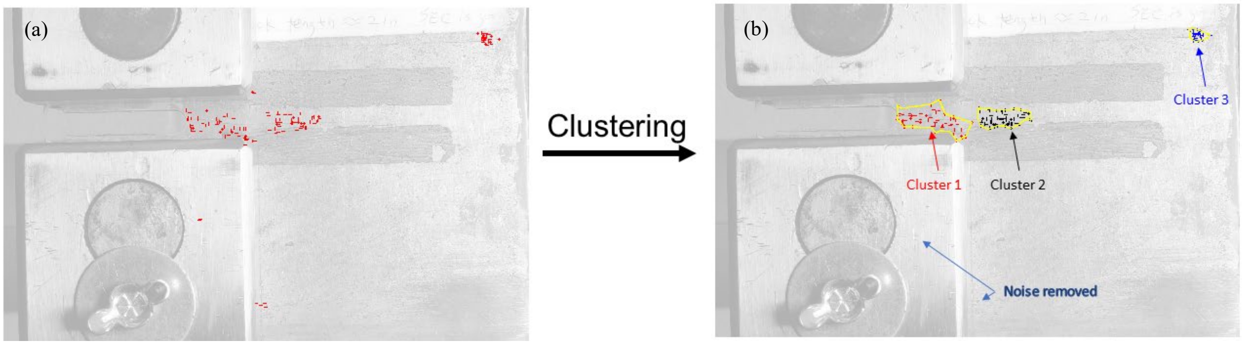
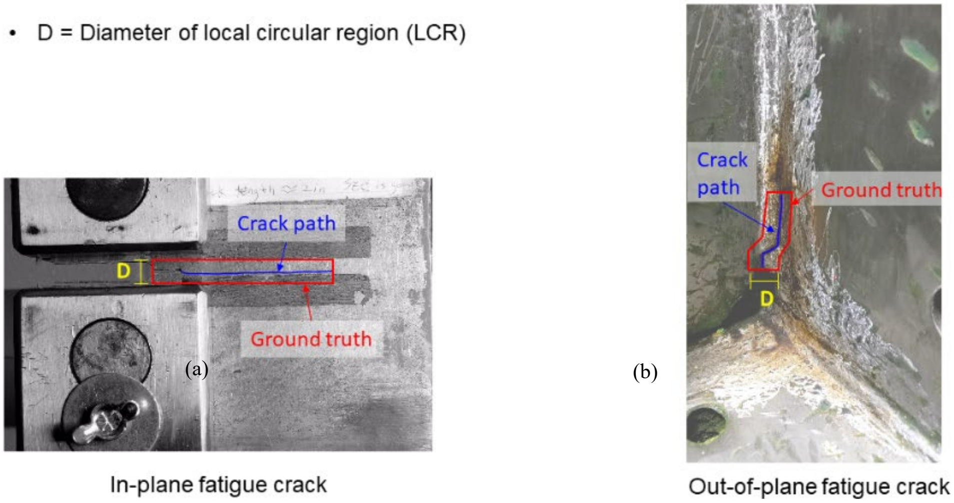


Figure 25: Clustering of detected feature points

We define ε as the radius of LCR and 10 as *MinPts*. DBSCAN starts with an arbitrary feature point. Then the point's neighborhood is defined based on ε. If it contains at least *MinPts* features, it is considered as a cluster. Otherwise, that point is labeled as noise. After that, the second point is considered, which can be clustered in the same group or a new group or determined as noise. Figure 25(a) shows the crack detection result with a low threshold value, and Figure 25(b) shows the clustering result, in which three clusters of feature points are shown as red, black, and blue. Feature points located on the bottom clevis and the clevis to plate boundary are more than ε pixels away from all the three clusters. Moreover, those isolated feature point groups contain less than *MinPts* points (10) hence are removed as noise. For the three clusters, the outermost points of each cluster are determined to create a boundary. Then, each pixel inside the boundary is labeled as a crack pixel. Thus, the detected sparse feature points are now converted into the continuous areas for quantification.



In-plane fatigue crack

Out-of-plane fatigue crack

Figure 26: Ground truth labeling

Ground truth

The second step is to determine the ground truth based on the actual crack length and location. Figure 26(a) shows the in-plane fatigue crack path of the C(T) specimen in a solid blue line. The Ground truth is determined as a rectangular region with its length equal to the crack length plus the notch of the specimen. Since the width of the detected feature point cluster depends on the diameter of the LCR, the width of the ground truth is defined as the diameter D of the LCR. The crack is located in the middle of the rectangle region. For the bridge girder, the fatigue crack has three distinct branches A, B, and C. Since the crack movement is dominated by branch A, only branch A is considered as the ground truth for quantification. As shown in Figure 26b, branch A constains three linear segments, hence a polygon with height as D is selected as the ground truth.

Intersection over Union (IOU)

The third step is to computer the IOU of the detected crack region with respect to the ground truth. The IOU describes the extend of overlap between two regions, and is formulated as:

$$IOU = \frac{Area\ of\ Intersection}{Area\ of\ Union}$$

The IOU ranges between 0 to 1, where 0 indicates no overlap and 1 indicates perfect overlap. Figure 27 demonstrates the three examples of IOU values. The red box denotes the predicted label, and the green box represents the ground truth. As illustrated in the figure, an IOU of 0.40 shows good localization but poor coverage, and an IOU of 0.73 indicates good localization and coverage, while an IOU of 0.92 shows very high overall accuracy in both detection and coverage.

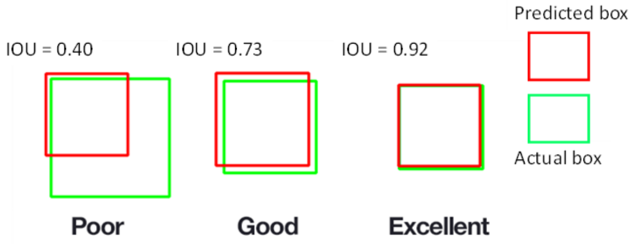


Figure 27: Illustration of different values of IOU

Experimental setups

The proposed distance-based method was experimentally tested using the same laboratory setups as the displacement-based method through both 2D and 3D videos.

Results and Discussions

To compare the accuracy and performance of both methods, same videos were used to detect fatigue crack using the proposed distance-based approach. Results are discussed in the following section.

In-plane Fatigue Crack Detection using 2D Video

The selected ROI and initial detected features identical to the displacement-based method are shown in Figure 28. A total of 3,000 feature points were detected using the Shi-Tomasi feature point detection algorithm. The radius of LCR was selected as 35 pixels. The crack detection results for a range of threshold value are presented in Figure 29. The actual crack is highlighted by a black line as the reference. Figure 29a shows the crack detection result with a very low threshold value of 0.01. Therefore, almost all feature points are kept in the result. Figures 29b and 29b show the crack detection results for threshold value 0.06 and 0.07, respectively. In both results, feature points still cover the entire ROI, so they are not considered as the final crack detection result. Figure 29d presents the crack detection for the threshold value of 0.09. In this result, features are clustered in a single group and do not spread over the entire ROI. Thus it was determined to be the final crack detection result.

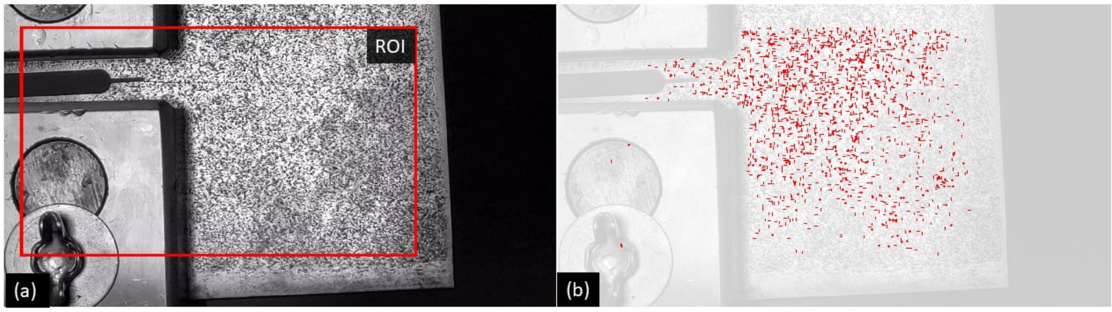


Figure 28: The initial frame of the 2D video with the selected ROI (a) and all feature points detected by the Shi-Tomasi algorithm (b). Note that the brightness of image in (b) is enhanced to highlight the feature points

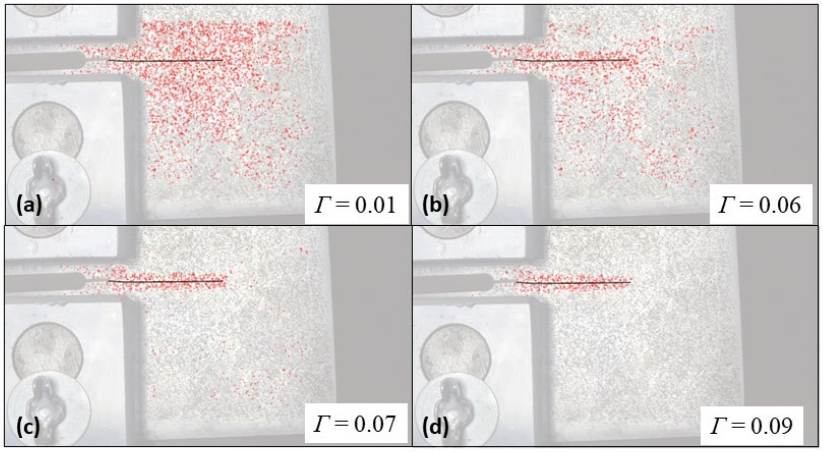


Figure 29: In-plane fatigue crack detection result of a C(T) specimen

To understand the effectiveness of the distance-based method to discover differential surface motion patterns, the distance histories of two feature pairs from the cracked and non-cracked regions are plotted in Figure 30. A feature pair from the cracked region are represented by blue marks, while a pair from non-cracked region are highlighted by orange marks as shown in Figure 30a. The distance history of the cracked region

(Figure 30b) shows eight cycles corresponding to the 0.5Hz fatigue load applied to the C(T) specimen leading to a higher standard deviation. On the other hand, the feature point pair from non-cracked region has very low distance change resulting in a much lower standard deviation. In the threshold scanning process, as the threshold value increases, feature pairs with low relative differential motion will be filtered out while feature pairs with higher relative differential motion will remain in the crack detection result.

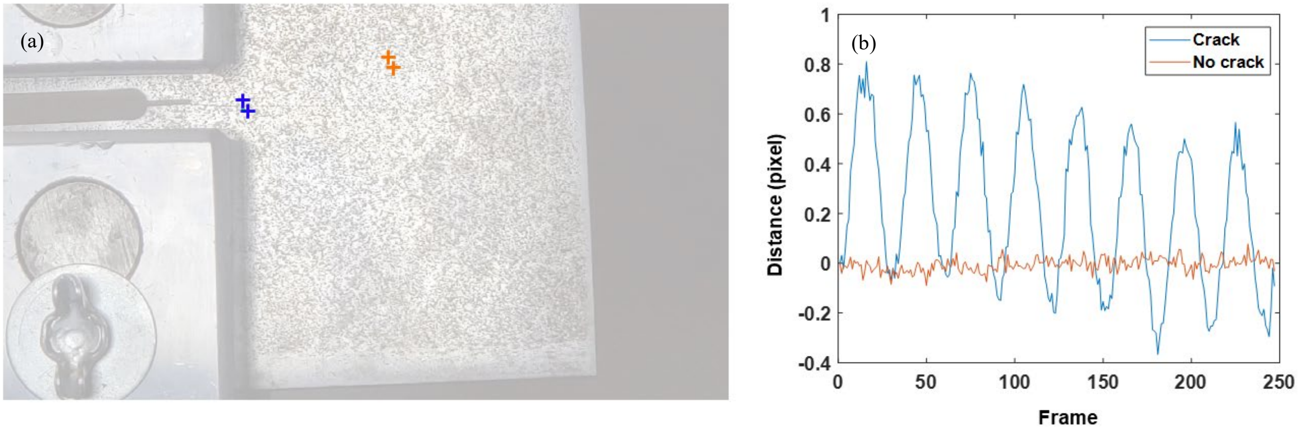


Figure 30: Feature point distance history for crack and uncracked region

Quantification of the Detected Crack

Once the fatigue crack is detected and localized, the accuracy of the detected crack is computed for both the displacement-based approach and the distance-based approach. First, clustering is performed for the crack detection results. Figure 31a shows the final crack detection result using the displacement-based method, and Figure 31b illustrates the clustered crack detection results. In the analysis, ε was set as 35 corresponding to the radius of LCR, and *MinPts* was set as 10. The crack detection result was then clustered into three groups based on the parameters and a few isolated feature points were removed as noise. Figure 31c shows the clustering result overlaid on top of the ground truth. Similarly, the quantification of crack detection using the distance-based approach is shown in Figure 32. The IOU scores for the displacement-based and distance-based crack detection results are calculated as 0.38 and 0.73, respectively. Evidently, the distance-based approach achieved a significantly higher accuracy in fatigue crack detection using the 2D video.

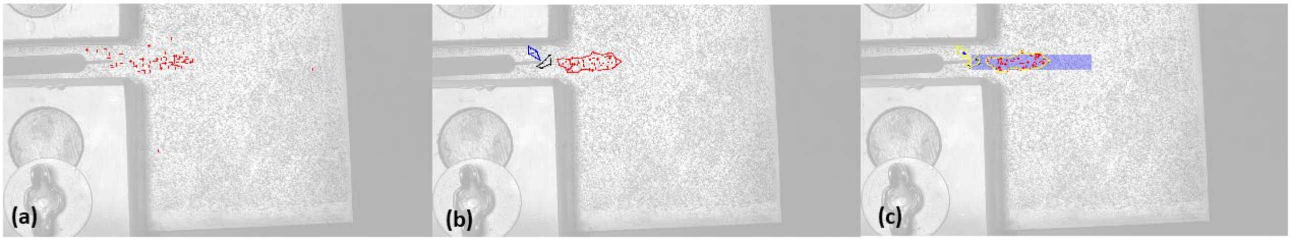


Figure 31: Quantification of detected crack in the C(T) speicmen using the displacement-based method (a) detetcted crack; (b) clustered crack; and (c) ground truth and clustered crack

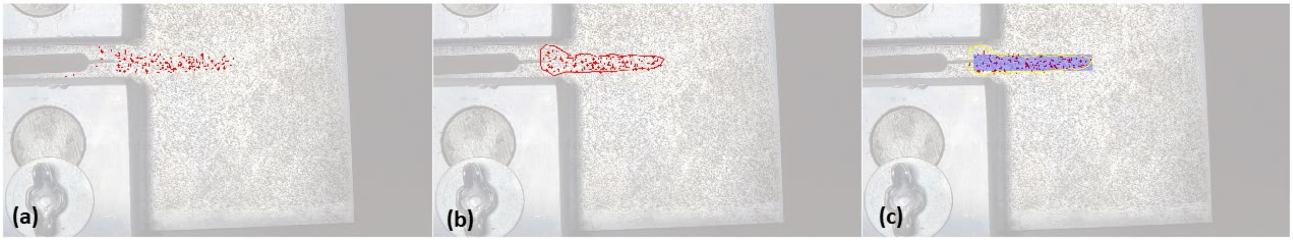


Figure 32: Quantification of detected crack in the C(T) speicmen using the distance-based method (a) detetcted crack; (b) clustered crack; and (c) ground truth and clustered crack

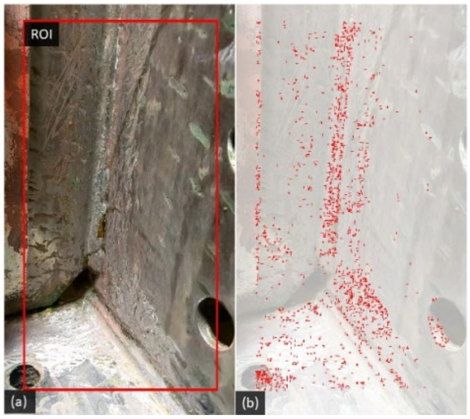


Figure 33: The initial frame of the 3D video with ROI (a) and all feature points detected by the Shi-Tomasi algorithm (b). Note that the brightness of image in (b) is enhanced to highlight the feature points

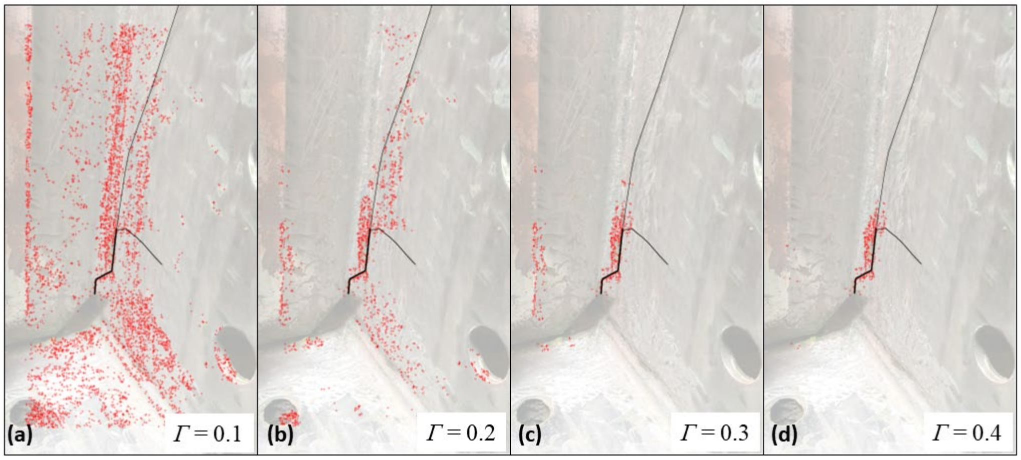


Figure 34: Out-of-plane fatigue crack detection result of the bridge girder specimen

Out-of-plane Fatigue Crack Detection using 3D Video

The selected ROI and the initial detected features identical to the displacement-based method are shown in Figure 33. A total of 5,000 feature points were detected using the Shi-Tomasi feature point detection algorithm. The radius of LCR was selected as 35 pixels. The crack detection results for a range of threshold values are presented in Figure 34. The actual crack is highlighted by black lines as the reference. Figure 34a shows crack detection with a low threshold value of 0.1; therefore, almost all feature points are kept in the result. Figures 34b and 34c illustrate the crack detection results for threshold values of 0.2 and 0.3, respectively. In both results, feature points were still distributed over the entire ROI, so they are not considered as the final crack detection result. Figure 34d presents the crack detection for the threshold value of 0.4. In this result, features are clustered in a single group and do not spread over the entire ROI. Thus it was determined to be the final crack detection result.

The effectiveness of the proposed distance-based crack detection approach is also evident in the distance history of feature pairs from the cracked and non-cracked regions, as shown in Figure 35. One feature pair is selected from cracked region denoted by blue marks, while a feature pair from no-cracked region is shown by orange marks. Their distance histories are plotted in the Figure 35b. The distance history of the non-cracked region feature pair shows no significant change over time because the feature points are moving similarly due to the rigid body movement. However, the distance history of the feature pair from the cracked region cleary shows two cycles corresponding to the applied fatigue load due to the crack movement. Capturing the cyclic motion leads to a higher standard deviation, enabling successfully crack detection through the differential motion pattern.

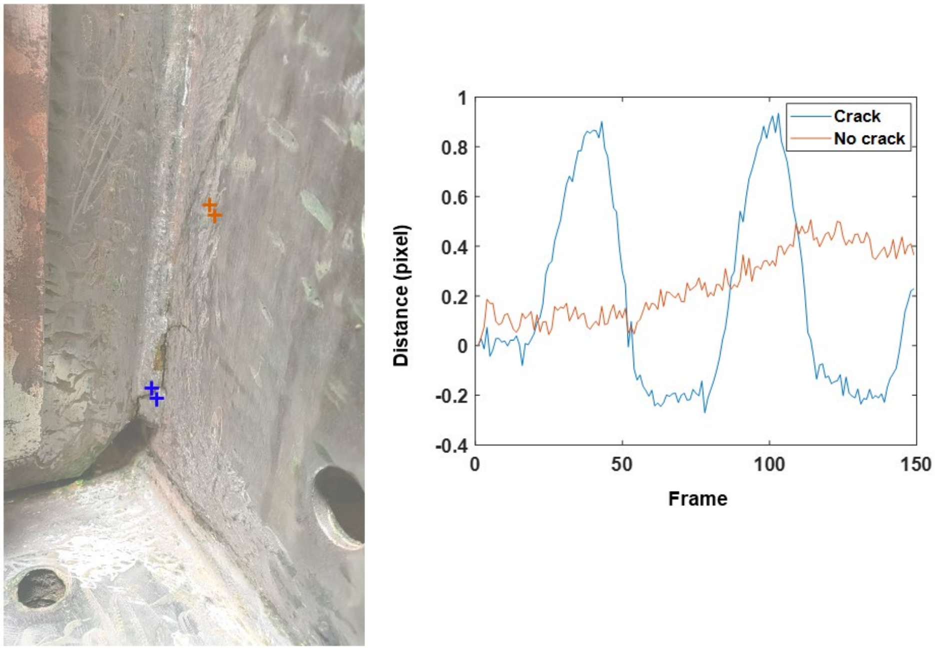


Figure 35: Feature point distance history for crack and uncracked region

Quantification of the Detected Crack

Once the fatigue crack was detected and localized, the accuracy of the detected crack was computed for both the displacement-based approach and the distance-based approach. Figure 36a shows final crack detection result using displacement-based method, and Figure 36b represents the clustered crack detection results. Again, in the analysis, ε was set as 35 corresponding to the radius of LCR, and *MinPts* was set as 10. The crack detection result was clustered into one group based on the parameters and a few isolated feature points were removed as noise. The clustered result is overlaid on top of the ground truth as shown in Figure 36c. The corresponding results for the distance-based approach are shown in Figure 37. The IOU scores for the displacement-based and distance-based approaches are 0.35 and 0.66, respectively. Similar to the 2D video , the distance-based approach achieved significantly higher accuracy in fatigue crack detection in 3D video.

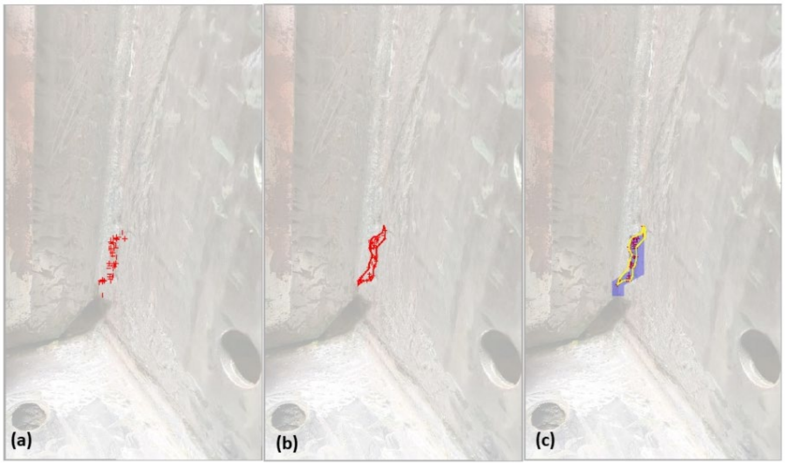


Figure 36: Quantification of the detected crack in the bridge girder speicmen using the displacement-based method (a) detetcted crack; (b) clustered crack; and (c) ground truth and clustered crack

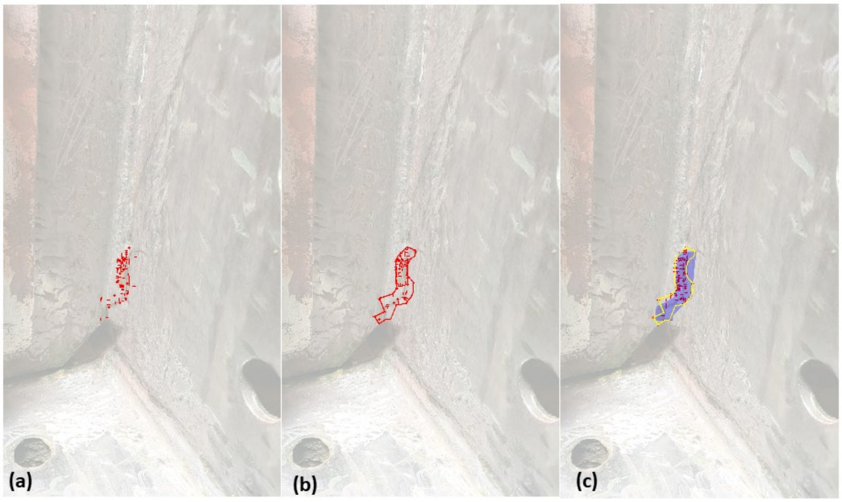


Figure 37: Quantification of the detected crack in the bridge girder speicmen using the distance-based method (a) detetcted crack; (b) clustered crack; and (c) ground truth and clustered crack

Parametric Study

A parametric study was conducted to understand the impact of fatigue load level on the crack detection accuracy. Videos were taken from two different views as shown in Figure 38. View 1 looks at the cracked region at roughly an equal distance from the connection plate and the girder web. In view 2, the video is captured by staying almost parallel to web of the girder, leading to a much higher parallax effect compared with view 1. A total of 10 load cases were considered in this parametric study. The minimum load level (F_{min}) in each load case was 0.2 kip and the maximum load level (F_{max}) started at 0.5 kips for LC1 and was increased by 0.5 kip in each subsequent load case. As a result, the first load case has a load range of 0.2 kip to 0.5 kip and the last load case ranges from 0.2 kip to 5 kip. Table 1 summarizes the parameters for all the load cases considered in the parametric study and their respective IOU results.

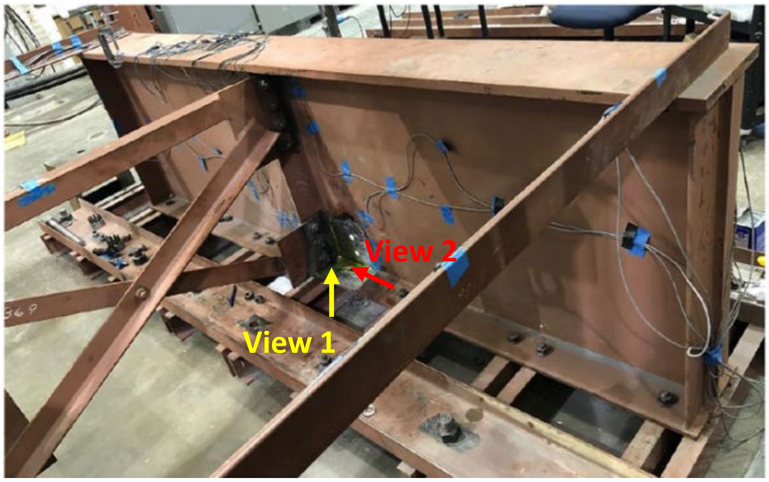


Figure 38: Two camera angles, view1 and view 2, for the distortion-induced fatigue test setup

Using the proposed IOU measure, we further quantified the results for the parametric study based on the large-scale bridge specimen with out-of-plane fatigue crack. Figure 39 and Figure 40 show the detected crack overlaid on top of the ground truth. In particular, the ground truth is shown in a blue polygonal area and the crack detection results are shown with regions with yellow boundaries and the contained feature points. The IOU results are summarized in Table 1.

Table 1: IOU results for parametric study variations of out-of-plane fatigue crack in bridge girder specimen

Load Case	Load Level (kips)		Crack Opening (mm)		IOU	
	F_{min}	F_{max}	View 1	View 2	View 1	View 2
LC1	0.2	0.5	0	0	0	0
LC2	0.2	1	<0.5	<0.5	0	0.42
LC3	0.2	1.5	0.6	0.6	0.34	0.44
LC4	0.2	2	0.7	0.72	0.40	0.43
LC5	0.2	2.5	0.9	1	0.72	0.44
LC6	0.2	3	1	1.2	0.68	0.68
LC7	0.2	3.5	1.1	1.5	0.77	0.64
LC8	0.2	4	1.3	1.64	0.66	0.68
LC9	0.2	4.5	1.5	2	0.71	0.70
LC10	0.2	5	1.8	2.2	0.62	0.69

As shown in Table 1, videos recorded with View 1 angle have IOU values ranging from 0 to 0.77. In general, as expected, increasing the load level increases the IOU value, making it easier to detect the crack. IOU is 0 for lower load cases such as LC1 and LC2, and is 0.34 and 0.40 for LC3 and LC4. For load cases LC5 to LC10, the IOU values are higher than 0.66, indicating the detected crack area covers more than 2/3 of the actual crack area. For View 2, the IOU ranges between 0.42 and 0.44 for LC 2 to LC5. The detected crack area is more than 64% for load cases LC6 to LC10. Table 1 also lists the crack opening levels associated with each load case under View 1 and View 2, which indicate that the algorithm consistently detected the fatigue crack when the crack opening is higher than 0.5 mm.

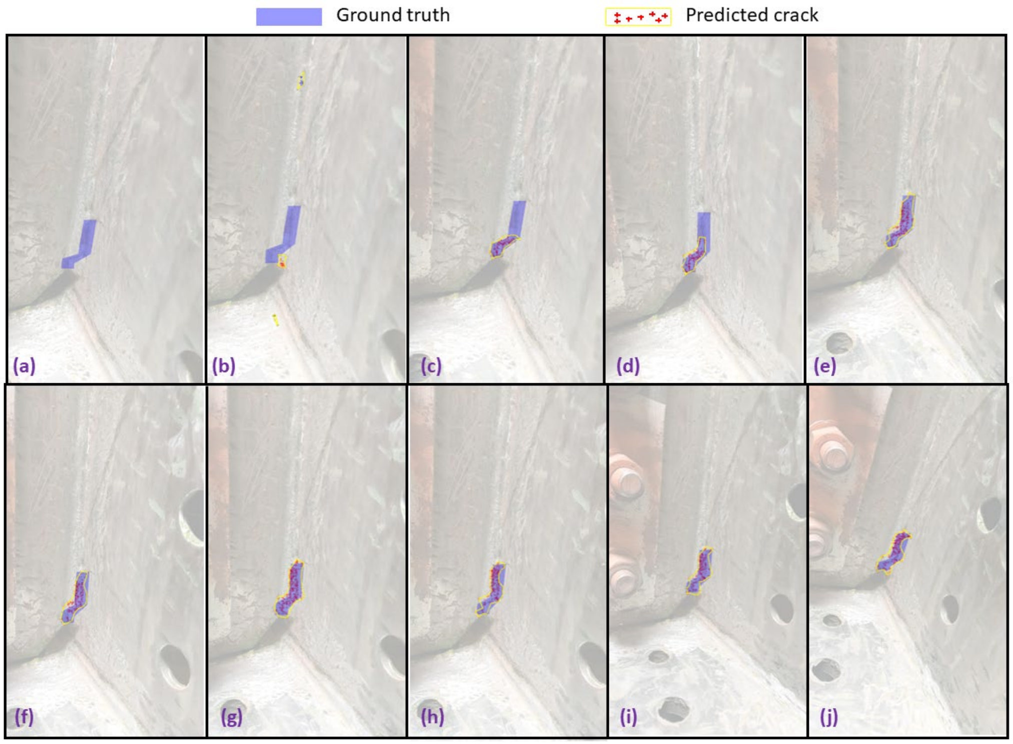


Figure 39: Crack detection results with the ground truth from videos with View 1: (a) LC1, (b) LC2, (c) LC3, (d) LC4, (e) LC5, f) LC6, (g) LC7, (h) LC8, (i) LC9, and (j) LC10

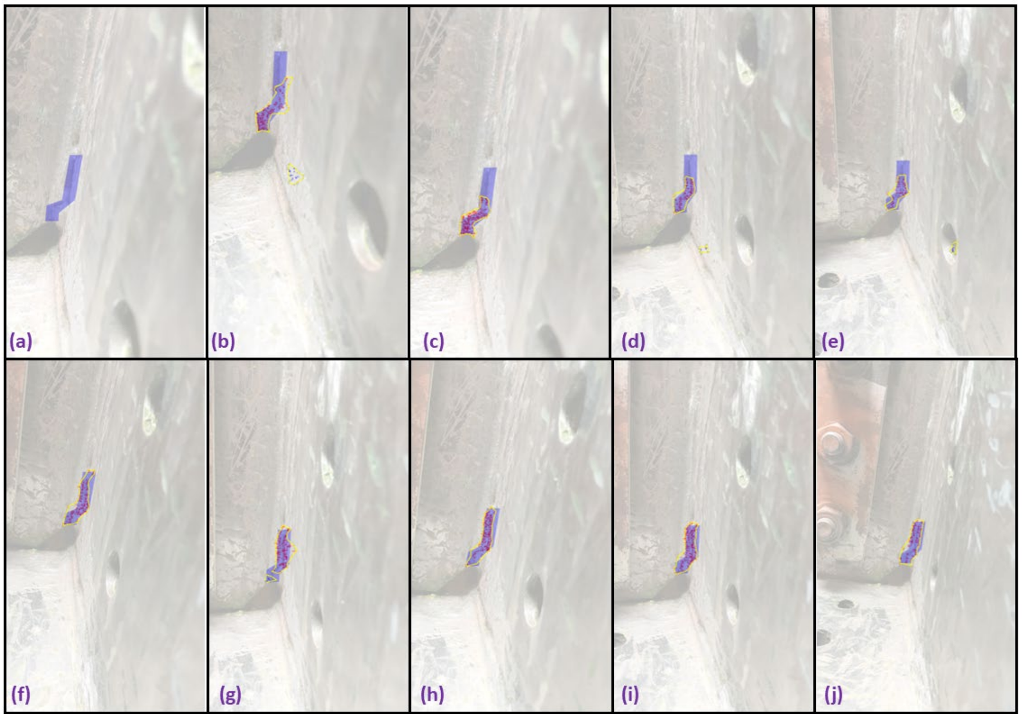


Figure 40: Crack detection results with the ground truth from videos with View 2: (a) LC1, (b) LC2, (c) LC3, (d) LC4, (e) LC5, f) LC6, (g) LC7, (h) LC8, (i) LC9, and (j) LC10

Computation Time of the Proposed Distance-based Algorithm

Since no GMC is required, the distance-based method has much higher computation efficiency compared with the displacement-based method. The computation time for the distance-based crack detection for the 2D video is around 29 seconds on a machine with an intel i9 processor and 64 GB of RAM. The 2D video contains 248 frames with standard HD resolution of 1980 × 1080 pixels, resulting in computation cost of about 0.12 seconds per video frame. For the 3D video, the computation time is around 31 seconds using the same machine. The 3D video has 149 frames of standard HD resolution, which was then reduced to 1980 × 1080 pixels. Hence, the computation cost is about 0.2 seconds per video frame. The proposed vision-based approach can be utilized with mixed reality devices to enable human-machine collaboration to help inspectors achieve more accurate visual inspection and expedite the decision-making process.

DEVELOPMENT OF AUGMENTED REALITY SOFTWARE FOR HUMAN-CENTERED FATIGUE CRACK INSPECTIONS

Augmented Reality (AR) software has been developed for fatigue crack localization during visual inspection. The software localizes the fatigue crack using a holographic interface for this purpose. The hologram is converted from the feature points generated by the computer vision algorithm developed in this project. Therefore, the software creates holograms based on the feature points generated by the MATLAB software. Figure 41 schematically shows the AR software's information circulation and workstream.

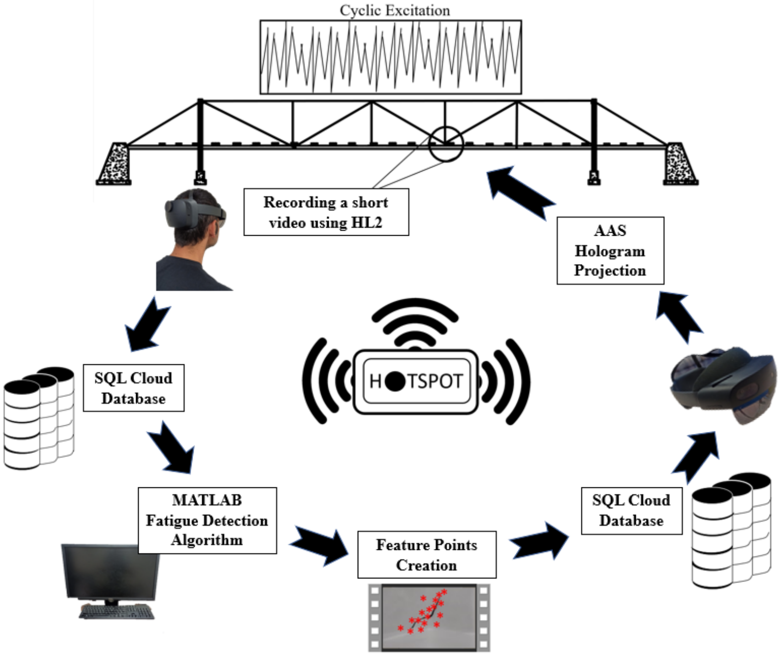


Figure 41: Architecture of the developed AR software for fatigue crack inspection

As shown in Figure 41, the AR software integrates various components to enable human-centered fatigue crack inspection. The research team developed an automatic workstream between all the components to achieve a user-friendly platform. The following section describes each part involved in the AR software. Note that to establish the connection between the different components, a hotspot connection is employed through a wireless router.

AR Headset

Augmented Reality (AR) has recently attracted great attention in SHM because of its unique abilities (Sadhu et al., 2023, Moreu et al., 2022). As a wearable technology with the ability of on-board processing and augmenting the real-world view with virtual objects, AR has the potential to improve inspector's perception during inspections. For example, Malek et al. (2022) deployed the simple Canny edge detection algorithm to an AR device utilizing its limited computational capacity, enabling real-time calculation of crack width through AR. Meanwhile, other researchers have explored AR technology through the connection with a central processing unit for more complex analyses (Mascarenas et al., 2021; Wang et al., 2021; Outay et al., 2022).

In this study, the Microsoft HoloLens 2nd generation (HL2) is adopted as the AR device. HL2 can conduct 3D scanning (spatial mapping) of the surrounding environment, programming, and projection (Napolitano et al., 2019). Spatial mapping, further described in this report, is utilized to find the correct place of fatigue cracks for hologram anchoring. In this study, HL2 is utilized to generate and anchor the holographic image to the real world in near-real-time. As shown in Figure 42, the HL2 is equipped with four depth cameras (two on each side), enabling high-quality 3D mapping of the surrounding environment. In addition, HL2 has two eye-tracking sensors for following the inspector's vision and a high-quality camera to capture photos or record videos of the environment. Moreover, HL2 is able to blend the real and virtual medias and provide photos and videos of surroundings, including the anchored virtual holographic objects.

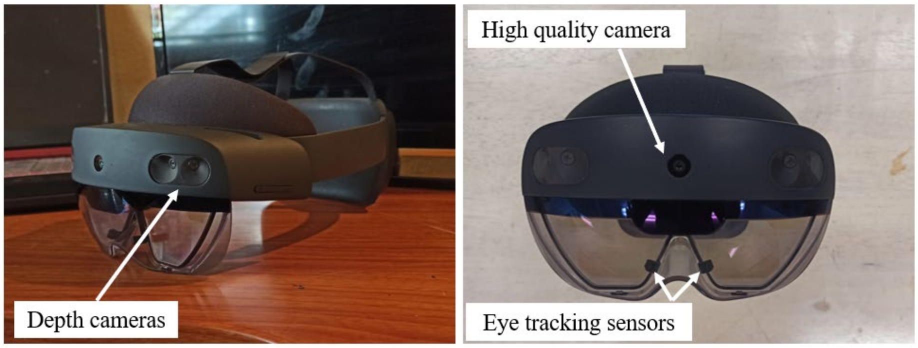


Figure 42 Microsoft HoloLens 2nd generation (HL2)

SQL Cloud Database

To establish the connection between the HL2 and the server that runs the crack detection algorithm, an SQL (Structured Query Language) database is created, and a wireless router is used to provide a hotspot to connect the various components (MATLAB, SQL database, and HL2), as shown in Figure 1. All connections provided two-way communication, enabling the SQL database to send and receive data from other components. For this purpose, three different tables are created in the SQL database, as shown in Figure 43, with each table performing a specific task. Specifically, the “preprocessed_videos” table detects the short video recorded by the HL2 using the "Video Upload" virtual button. Then the software sends the video to the database through the wireless hotspot. The server downloads the uploaded video and analyzes it with the MATLAB algorithm to generate feature points and detect possible fatigue cracks from the recorded video. Then, the server sends the coordinates of all the feature points back to the database through the hotspot connection. The “postprocessed_data” table receives the coordinates of the feature point group and sends it to the HL2 to generate a hologram and project it over the actual fatigue crack. The “lastframe_transform” table is a part of the developed Auto Anchoring System (AAS), which helps HL2 to accurately anchor the hologram. It is worth noting that all the steps are performed automatically without any need for user intervention.

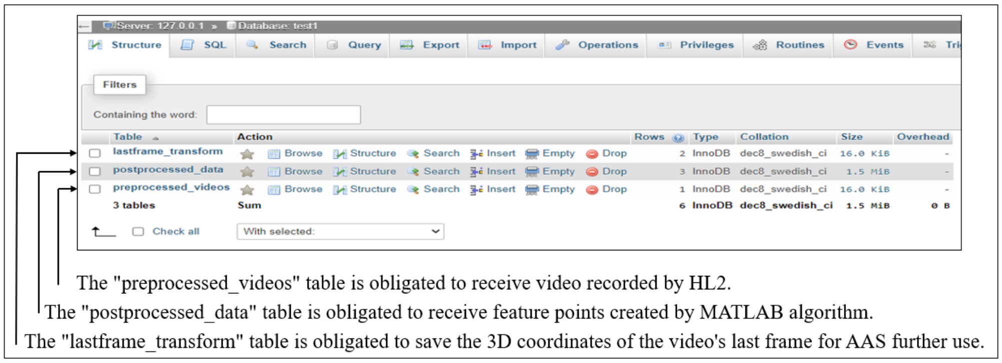


	Table	Action							Rows	Type	Collation	Size	Overhead
□	lastframe_transform	☆	Browse	Structure	Search	Insert	Empty	Drop	2	InnoDB	dec8_swedish_ci	16.0 KiB	-
□	postprocessed_data	☆	Browse	Structure	Search	Insert	Empty	Drop	3	InnoDB	dec8_swedish_ci	1.5 MiB	-
□	preprocessed_videos	☆	Browse	Structure	Search	Insert	Empty	Drop	1	InnoDB	dec8_swedish_ci	16.0 KiB	-
	3 tables	Sum							6	InnoDB	dec8_swedish_ci	1.5 MiB	0 B

The "preprocessed_videos" table is obligated to receive video recorded by HL2.
The "postprocessed_data" table is obligated to receive feature points created by MATLAB algorithm.
The "lastframe_transform" table is obligated to save the 3D coordinates of the video's last frame for AAS further use.

Figure 43 The SQL database established to connect different software’s components

Auto Anchoring System (AAS)

A novel AAS is developed to achieve automated anchoring and hologram surface projection. For this purpose, the innate ability of HL2 to conduct a 3D analysis of the surrounding area called spatial mapping is utilized. The software produces a hologram using the feature points generated from the video analysis and anchors it based on the HL2's spatial understanding of the surrounding environment. Although the HL2 provides reasonable 3D mapping of the surrounding area, two main challenges remain, including: 1) identification of the correct surface for projecting the hologram, and 2) anchoring the hologram over the crack in correct position and orientation. To address these challenges, the research team first developed an invisible rectangle parallel to the last frame of the video recorded by the HL2 using the last user's eye direction saved by HL2 during recording. In addition, to prevent the projection of the invisible rectangle behind the cracked surface, a short distance between the last video frame and the invisible rectangle is considered. The invisible rectangle hence floats in a constant position parallel to the last video frame between the last video frame's 3D position and the actual cracked surface.

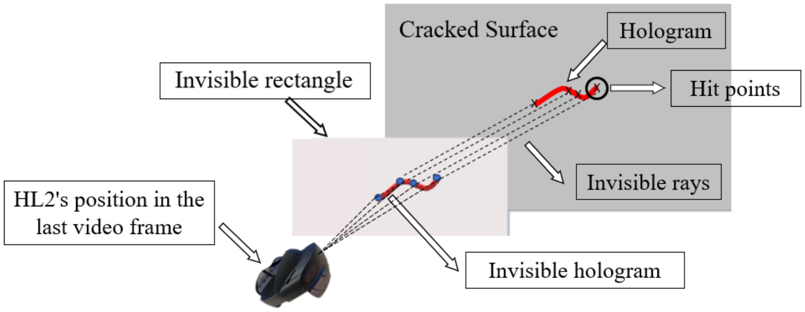


FIGURE 44 Illustration of the Auto Anchoring System (AAS)

Figure 44 schematically illustrates the developed AAS. A constant independent position for the invisible rectangle is provided by detaching its dependence on the camera position. Although this rectangle is the preliminary placement for the hologram, it is invisible to the user, hence the hologram appears to be floating in the air in the virtual media. In the next step, invisible rays are utilized to convey each holographic feature point from the invisible rectangle to the cracked surface. The endpoints of these rays are their hit points with the first HL2-detected spatial map surface. This surface is the desired surface for anchoring the holograms because the user looks directly at the area to record the short video, hence there are no other surfaces between the last video frame and the cracked surface. As a result, the software anchors the hologram on the crack in the correct direction. Moreover, this approach anchors the hologram in the right location as the projection is independent from the user's angle. Therefore, the user does not need to look at the surface perpendicularly during video recording, making this approach practical for inclined surfaces as well as surfaces perpendicular to the user's line of sight.

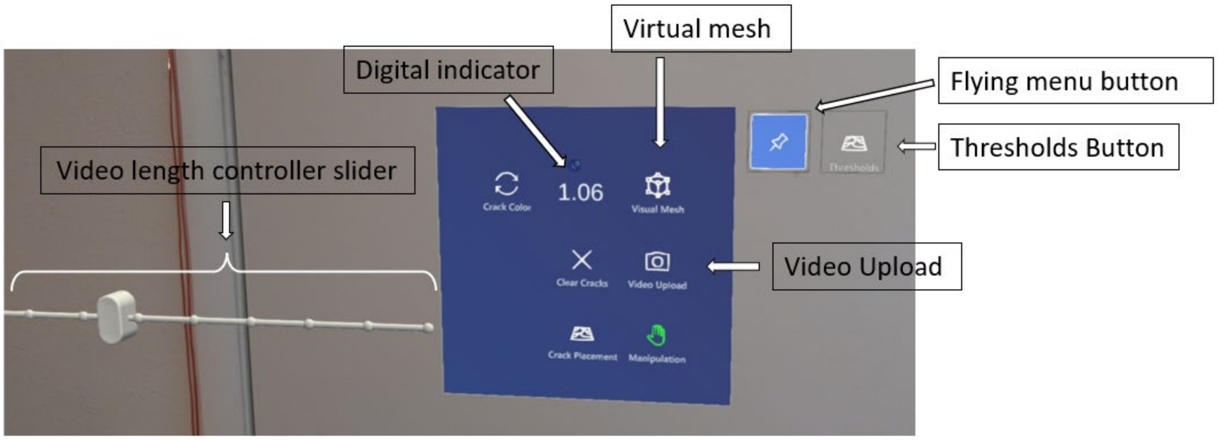


Figure 45 Virtual menu of the AR software for fatigue crack inspection

AR Virtual Menu

To enable a user-friendly AR interface, a virtual menu is created for the AR software to allow the user to easily operate the software. Figure 45 shows the virtual menu with its functionalities labeled. The functionalities are explained as follows.

Virtual Mesh: A 'Visual Mesh' button (Figure 45) is accessible on the top right part of the software's virtual menu. The visual mesh button utilizes HL2's understanding of the surrounding environment, referred to as "spatial mapping." This function enables the user to see every object the HL2 analyzes as its 3D surrounding environment, enabling the inspector to verify that the HL2 is analyzing the appropriate surface during the inspection.

Figure 46 shows the spatial mapping of the half-scale laboratory steel bridge girder specimen utilized for indoor testing, illustrating the level of detail captured by HL2. In addition, the mesh generated Figure 46 shows the HL2's ability to provide a detailed representation of the steel structural element surfaces.

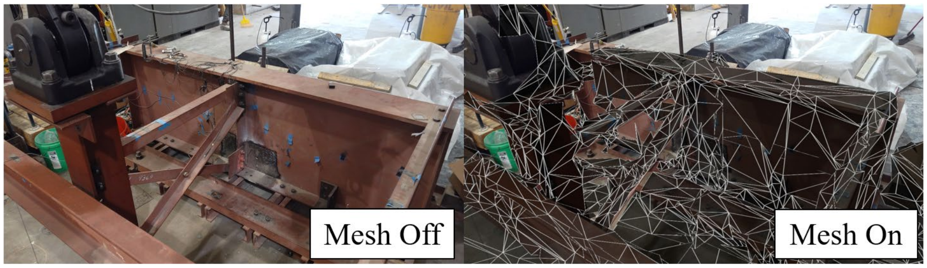


Figure 46 Visual mesh button reveals the HL2's representation of real-world surfaces steel girder test

Digital Indicator: A digital indicator, shown in Figure 45, is developed to measure and indicate in real-time the distance from the user to the closest perpendicular surface. This function helps the inspector maintain an appropriate distance from the surface, ensuring high-quality video recording for the subsequent analysis. Figure 47 (a) shows the testing of the digital indicator using a tape measurer as the reference. Figure 47 (b) shows the accuracy testing at various distances. Those photos were captured by the HL2's camera from the HL2's view.

Video Upload: As mentioned earlier, this method utilizes a short video to analyze and localize fatigue cracks. The HL2 records a short video including the direction of the user's vision in the last video's frame. Then, the video, along with the direction of user' vision and the 3D position of the HL2 camera, is automatically uploaded to the database described in the next section. An automatic connection between this function and the cloud database is established through a hotspot. Additionally, the same connection is provided between the MATLAB algorithm and the cloud database, so that MATLAB can automatically detect the new video uploaded to the database from the HL2. MATLAB then analyzes the recorded video and provides feature points of the detected fatigue crack, of which the coordinates are then uploaded back to the cloud database.

In the next step the software automatically detects a new group of feature points uploaded to the cloud database by MATLAB. Afterwards, it extracts the feature points as a group and relates them to the last video frame coordinates uploaded from HL2. HL2 then creates a hologram using the grouped feature points and projects it based on the feature points coordinates over the fatigue crack in the correct location and orientation using the coordinates of the last video frame. As explained previous, the software uses a novel projection method called AAS, described in a previous section.

From the use perspective, the inspector just needs to push the video upload button in the HL2's virtual media and look at the fatigue-prone area under active traffic loading for video recording. The software would take care of the rest and projects a hologram over the fatigue crack to show the detection result for the inspector.

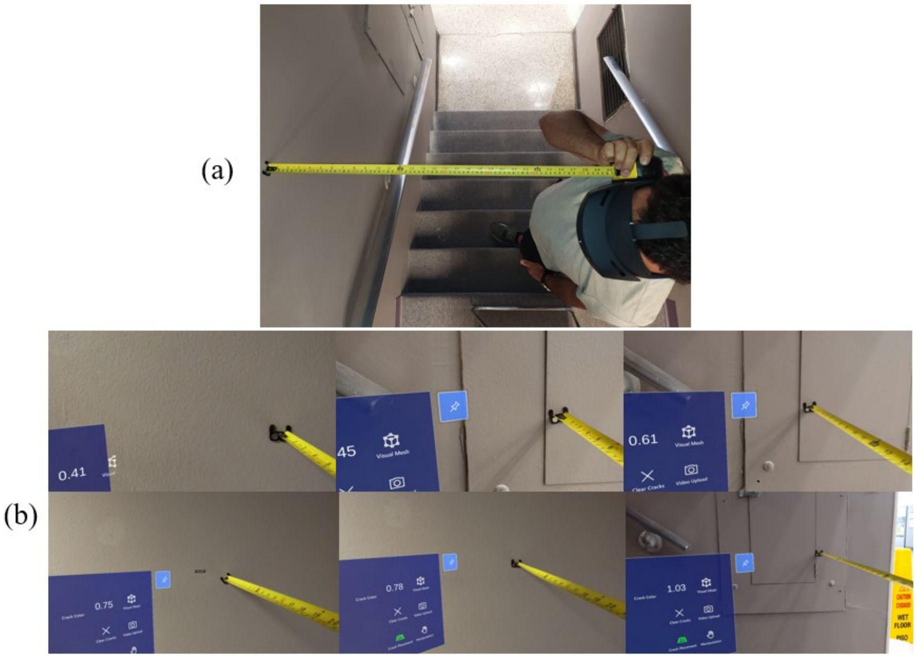


Figure 47 Digital indicator's accuracy test (a) test setup (b) HL2's view

Table 2 summarizes the results of the digital indicator's accuracy tests, demonstrating high measuring accuracy with less than 1 cm error for a 1-meter distance.

Table 2 Test results for the digital indicator's accuracy

Test No.	Tape Measure (m)	HL2 (m)	‖Tolerance‖ (m)
1	0.4	0.4	0
2	0.45	0.45	0
3	0.6	0.61	0.01
4	0.76	0.75	0.01
5	0.77	0.78	0.01
6	1.04	1.03	0.01

Video Length Controller: This slider lets the user control the video length (1~10 seconds) recorded by HL2. The length of the video can be adjusted to ensure enough fatigue load cycles are captured in the recorded video.

Flying Menu: The flying Menu function shown in Figure 45 switches the virtual menu between flying and hovering modes. The virtual flying menu refers to how the menu follows the inspector's movement during the inspection. The flying mode eliminates the inspector's need to manually move the menu during the inspection and leaves the inspector's hands free for other tasks. On the other hand, the hovering mode is appropriate when the user is trying to inspect a specific part of the structure. By switching to the hovering mode, the inspector can manually move the virtual menu to the appropriate position.

Thresholds: The 'Thresholds' function brings up several virtual buttons based on the number of thresholds for crack detection defined by the inspector. It automatically updates the menu according to the specific range of thresholds defined for each analysis. Figure 48 shows the virtual threshold buttons revealed by pushing the 'Thresholds' button. Each button corresponds to a single threshold result generated by the crack detection algorithm. This button enables the inspector to interact with the crack detection result by easily checking results associated with different thresholds, hence selecting the best result for documentation, tracking, and decision making.

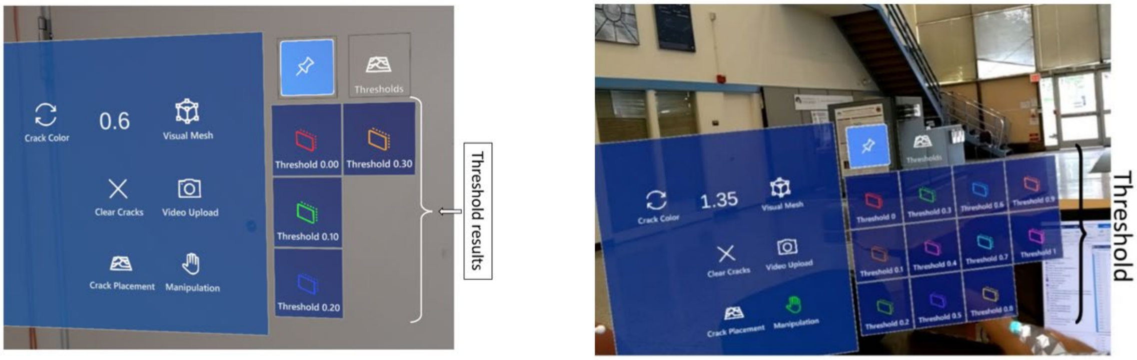


Figure 48 Virtual threshold functions automatically generated based on the crack detection results

Preliminary Field Experiment of the AR software

Preliminary indoor tests were designed and carried out to support software debugging and updating. A 2D benchmark test using a "paper crack" was created. Figure 49 shows the benchmark test, which includes a flaw within a paper that could be opened and closed through cyclic loading, mimicking fatigue crack behavior.

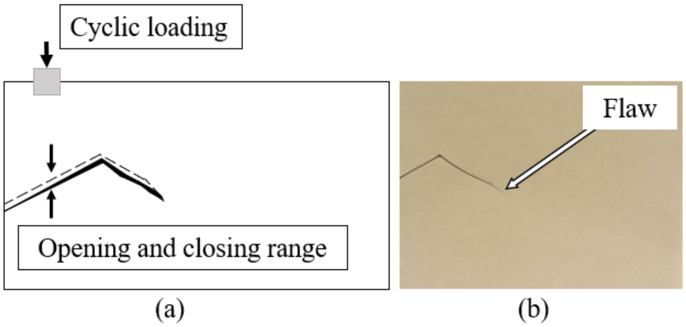


Figure 49 (a) In-plane flaw benchmark (schematic) (b) actual benchmark (photo)

The developed AR software was utilized to detect the flaw in the benchmark under cyclic loading. Figure 50 represents the crack detection result using AR projected on the benchmark. In this figure, the hologram generated by the HoloLens was laid over the crack to indicate the location of the fatigue crack to the inspector.

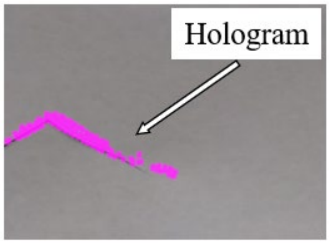


Figure 50 Hologram of the crack detection result using in-plane crack benchmark

The applicability of the developed AR fatigue inspection software was also evaluated for out-of-plane cracks. To this end, a 3D benchmark based on paper flaw was also utilized to simulate out-of-plane fatigue crack behavior. Figure 51 represents the benchmark and the flaw.

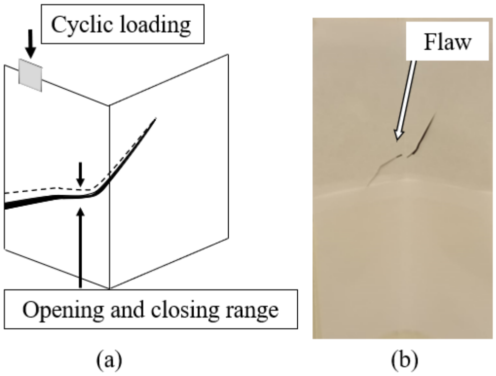


Figure 51 (a) Out-of-plane flaw benchmark (schematic) (b) actual benchmark (photo)

Figure 52 shows the generated hologram result using the AR-developed software on the 3D out-of-plane crack benchmark. The hologram was generated based on the feature points resulting from the MATLAB analysis. The hologram was laid over the crack to localize the flaw to the inspector.

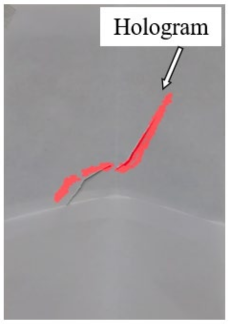


Figure 52 Hologram of the crack detection result using out-of-plane crack benchmark

After completing the software development phase through extensive debugging and testing, the research team conducted a preliminary field experiment. The goal is to test the software in outdoor enviroment. A benchmark surface including a simulated crack on a piece of paper was attached to a steel railway bridge structure. The simulated crack on the benchmark could open and close under external load. Figure 53 shows the outdoor benchmark, the experiment, and its result, which indicate the developed software could perform well under outdoor environment.

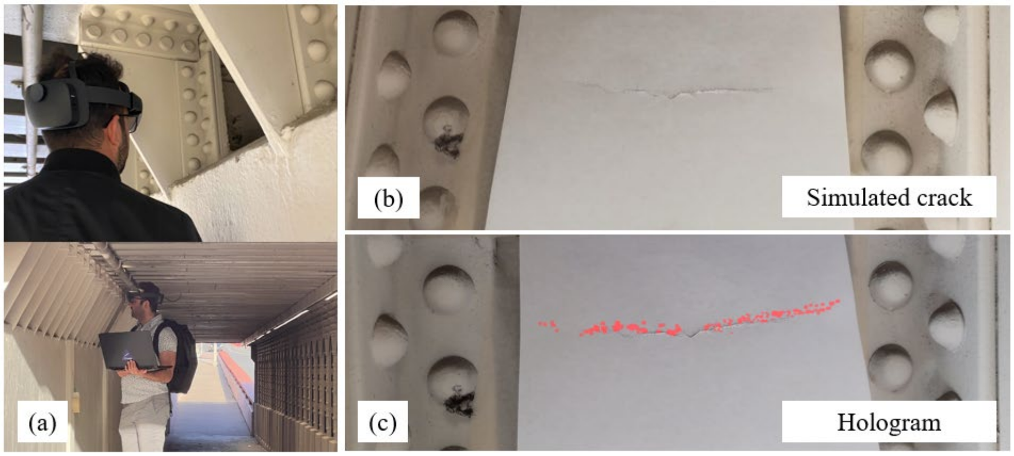


Figure 53. The preliminary field experiment (a) inspector visually inspects the simulated crack using the AR software (b) simulated fatigue crack (c) hologram anchored on the simulated crack

User Manual

This section provides hardware and software information for inspectors to use the developed AR software. Table 3 shows all hardware and software required to utilize and run the AR sofwtare. Also, Figure 54 illustrates the hardware and software schematically.

Hardware Connection

A local wireless connection between the HL2 and the laptop needs to be established through a hotspot. This local connection provides fast communication and doesn't depend on internet access. For this purpose, a typical Wi-Fi router could be used as shown in Figure 55.

Table 3 Required software and hardware

	Name	Version	Link
Software	Unity	2019.4.23f1 or later	https://unity3d.com/get-unity/download
	Visual Studio	2019 or later	https://visualstudio.microsoft.com/downloads/
	Mixed Reality Toolkits (MRTK)	5	https://github.com/Microsoft/MixedRealityToolkit-Unity
	MATLAB	All versions	https://www.mathworks.com/?s_tid=mlh_gn_logo
	XAMPP	V3.3.0 or later	https://www.apachefriends.org/download.html
	AR Software	-	The software developed for this project
Hardware	Laptop	-	-
	HL2	-	-
	Hotspot	-	-

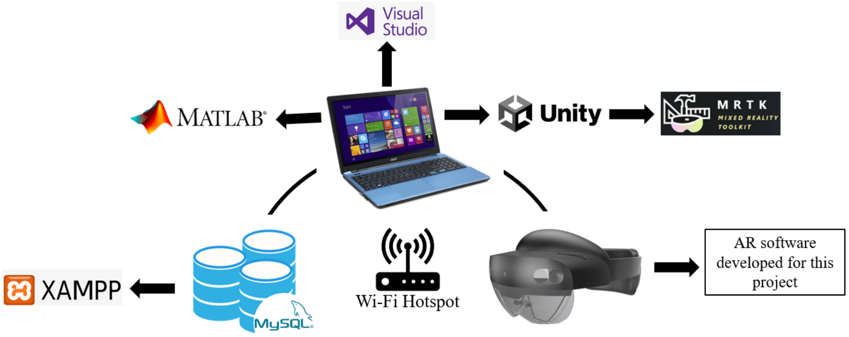

Visual Studio
MATLAB
Unity
MRTK MIXED REALITY TOOLKIT
XAMPP
MySQL
Wi-Fi Hotspot
AR software developed for this project

Figure 54 Hardware and software needed to use the developed AR software

Figure 55 (a) Wi-Fi router (b) Hotspot device

The developed AR-based inspection tool contains three critical parts:

1. MATLAB-based computer vision algorithm for fatigue crack detection

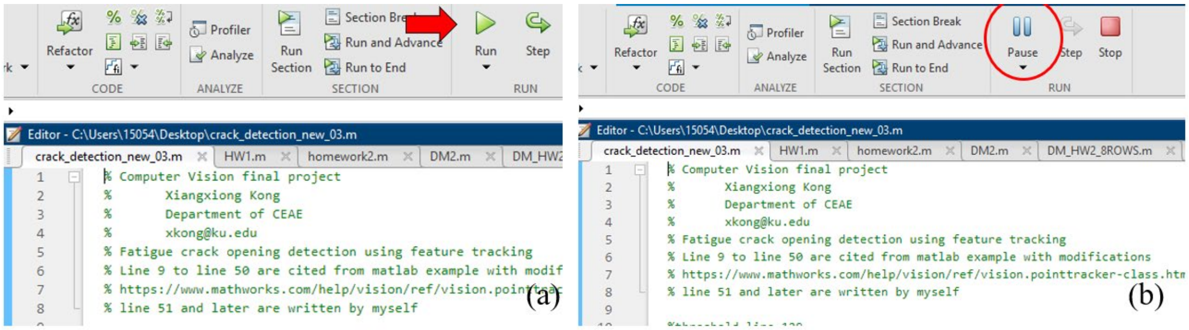
2. AR-based software for HL2
3. SQL database to connect 1 and 2.

Software Instruction

Before starting the crack detection process, it is important to make sure all components are ready by checking the following:

1. HL2 and laptop both are connected to the same Wi-Fi Hotspot.
2. The Database is connected and standing by.
3. The MATLAB algorithm is running and waiting for the video (Figure 56 (b)).
4. The AR software in HL2 is open and ready.

Figure 56 shows the MATLAB software running at the laptop. Once the MATLAB code is executed at the laptop, it enters the standby mode waiting for the new video uploaded to the server by the HL2, as shown in Figure 56b.



Editor - C:\Users\15054\Desktop\crack_detection_new_03.m

crack_detection_new_03.m | HW1.m | homework2.m | DM2.m | DM_HW2

```
% Computer Vision final project
%       Xiangxiong Kong
%       Department of CEAE
%       xkong@ku.edu
% Fatigue crack opening detection using feature tracking
% Line 9 to line 50 are cited from matlab example with modif
% https://www.mathworks.com/help/vision/ref/vision.pointtrac
% line 51 and later are written by myself
```

Editor - C:\Users\15054\Desktop\crack_detection_new_03.m

crack_detection_new_03.m | HW1.m | homework2.m | DM2.m | DM_HW2_8ROWS.m

```
% Computer Vision final project
%       Xiangxiong Kong
%       Department of CEAE
%       xkong@ku.edu
% Fatigue crack opening detection using feature tracking
% Line 9 to line 50 are cited from matlab example with modifications
% https://www.mathworks.com/help/vision/ref/vision.pointtracker-class.htm
% line 51 and later are written by myself
```

Figure 56 MATLAB software (a) before running (b) waiting to receive new uploaded video from HL2

The user can also choose the adjust the video recording length using the slide bar. Once everything is ready, the user simply needs to hit the virtual "Video Upload" button, as shown in Figure 57, and then look at the target surface for the selected amount of seconds to record the video under live load. The software takes care of the rest of the process and finally anchors the feature points from the MATLAB algorithm. Finally, the user can check different results corresponding to different threshold values defined in the MATLAB algorithm in an interactive manner.

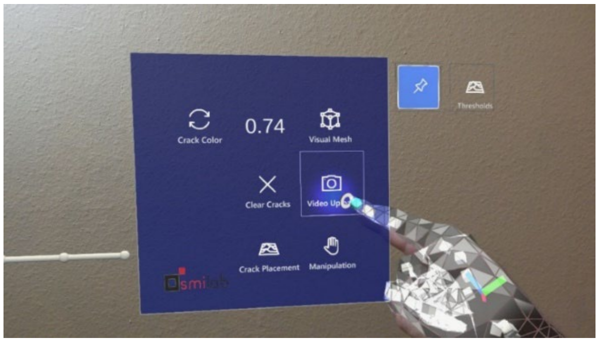


Figure 57 **Virtually pushing the "Video Upload" button**

LABORATORY AND FIELD IMPLEMENTATIONS

This section presents the laboratory and field implementations of the developed AR-based tool for human-centered fatigue crack inspection. The following sections provide detailed descriptions of both implementations.

Laboratory Implementation

Once the AR software is developed, the two teams met at KU to integrate the new distance-based crack detection algorithm implemented in MATLAB into the AR software to ensure seamless workflow. A SQL database is required to connect the crack detection program and HL2 and is downloaded and setup at KU. The entire process of bridge inspection was carried out at KU based on a large-scale bridge girder to cross-frame connection. Each step is explained and illustrated below.

Step 1: Turn on the Wi-Fi hotspot, then connect the server hosting the MATLAB program and HoloLens 2 to the local Wi-Fi network. Figure 58 shows a portable Wi-Fi hotspot, HL2 and a server hosting the MATLAB program connected to the local network of through the Wi-Fi hotspot. Then, run the MATLAB program in the server to bring it to the standby mode.

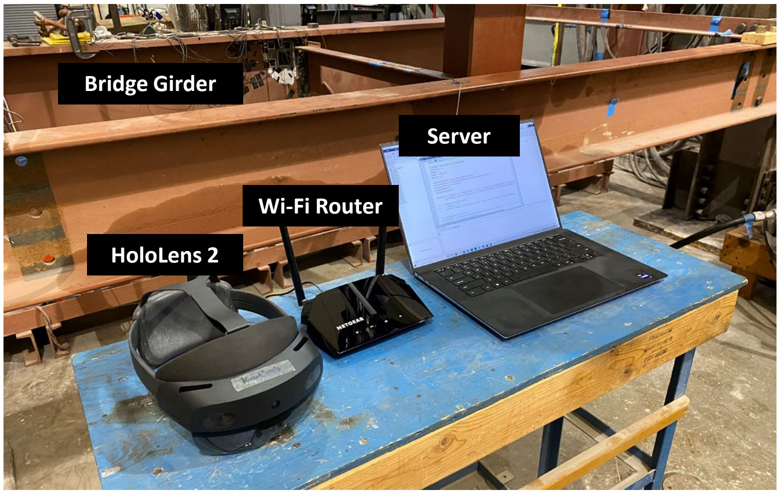


Figure 58 Hardware used in AR-based fatigue crack inspection

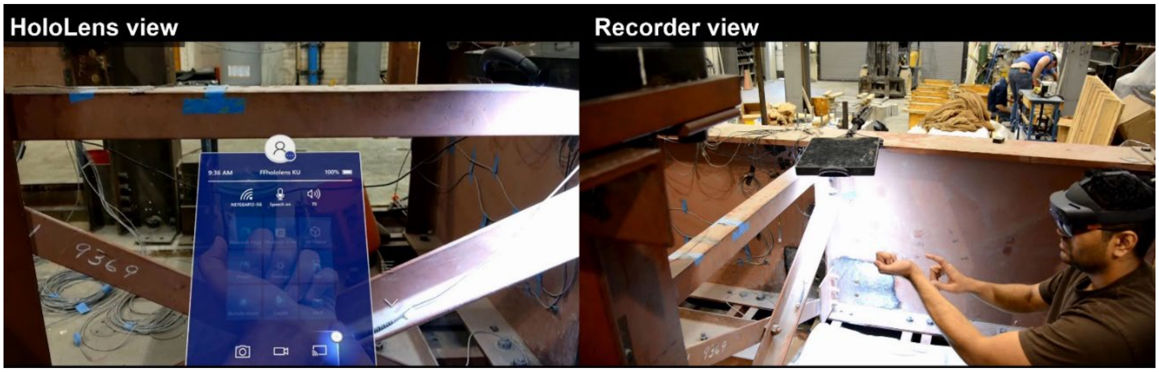


Figure 59 Inspector opening the windows menu of the HL2 at the inspection site

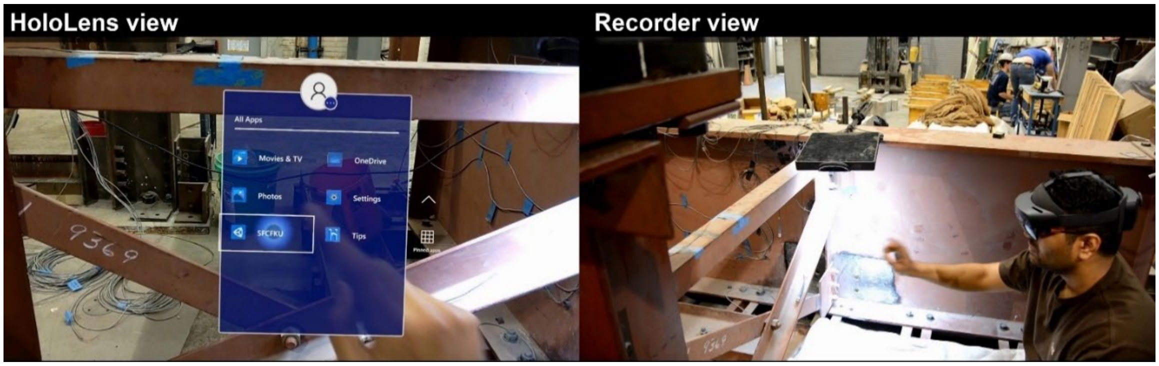


Figure 60 Inspector opening the AR software

Step 2: Go to the inspection site and open the HL2's main Windows menu. Figure 59 shows the inspector's view when he was opening the HL2's Windows menu near the bridge girder specimen at KU Structural Testing laboratory. Recorder view shown on the right half of the figure shows the recording camera view videotaping the entire bridge inspection process. Note that the two views are synchronized in the figure such that the inspector view on the left side of the figure is what inspector is looking through the HL2.

Step 3: Open the AR software in the HL2. Figure 60 shows both the HoloLens view and the inspector when the AR software interface was opened.

Step 4: Select the video recording length using the slide bar. Figure 60 shows both the HoloLens view and the inspector selecting the video recording length.

Step 5: Move the software menu at a convenient location. Once the AR software is open, application menu will appear in-front of the inspector's eyesight. Depending on the inspector's preference, move the software menu near the target inspection region by gripping the pin button as shown in Figure 62.

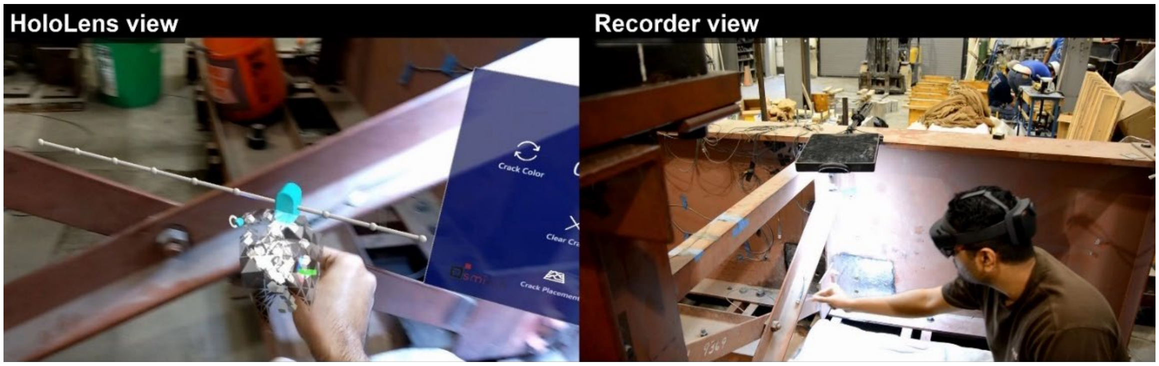


Figure 61 Inspector selecting video recording length using the slide bar

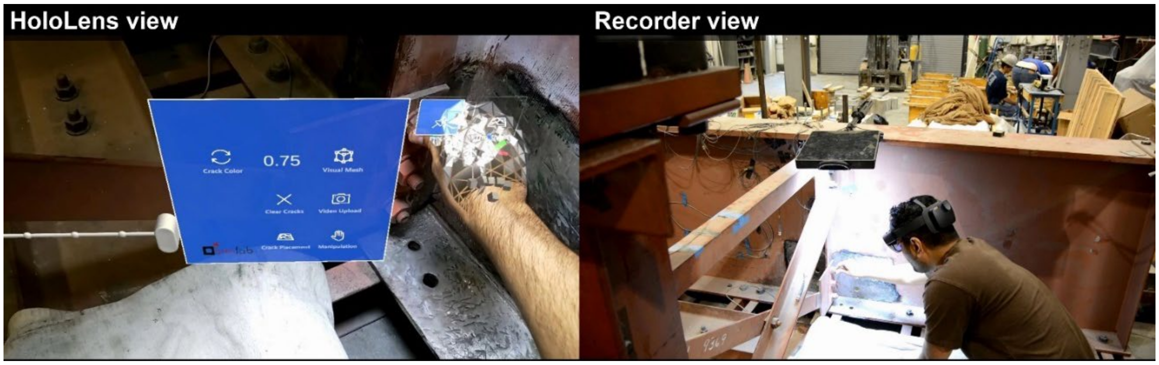


Figure 62 Inspector moving the AR software menu at a convenient location for inspection

Step 6: Collect the video by hitting the ‘Video Upload’ button. The inspector can start collecting the video after getting in the right position for the crack inspection. A video of selected length is automatically collected by hitting the ‘Video Upload’ button. This step is demonstrated by Figure 63.

Step 7: Once the video is recorded, it will be sent to the laptop/server for crack detection using MATLAB. This step is demonstrated by Figure 64.

Step 8: Once the video is processed in near-real time (around 30 seconds), the results of detected crack corresponding to a range of threshold values are sent to the HL2 as holograms. The inspector can then select a specific threshold value to see the crack detection result. For example, as shown Figure 65, the inspector first selects the zero-threshold value to inspect the initial feature points from the region analyzed. Then as demonstrated in Figure 66 to Figure 70, the inspector selects higher threshold values and inspects the detected fatigue crack results. In this process the inspector changed his position multiple time and could still see the fatigue crack feature points anchored on top of the crack, demonstrating the effectiveness of the developed AAS for hologram anchoring. As shown in Figure 70, the crack was successfully identified based using the developed AR-based tool.

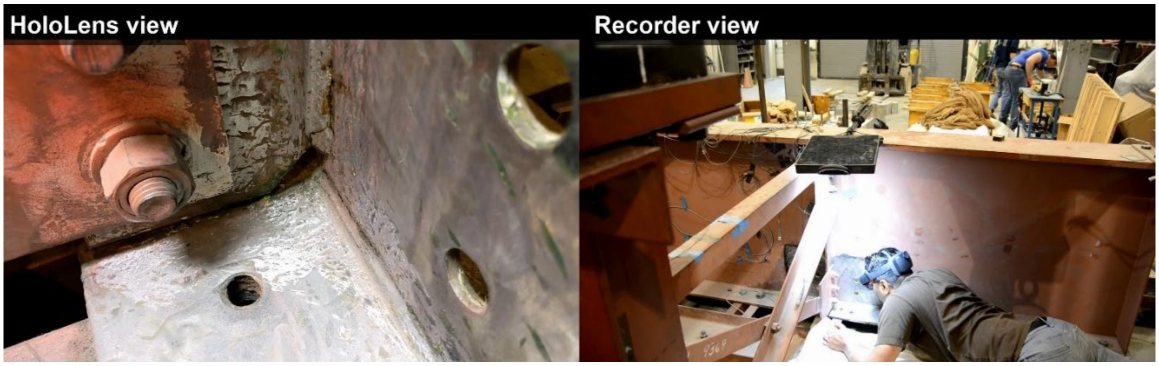


Figure 63 Inspector collecting a video for crack detection

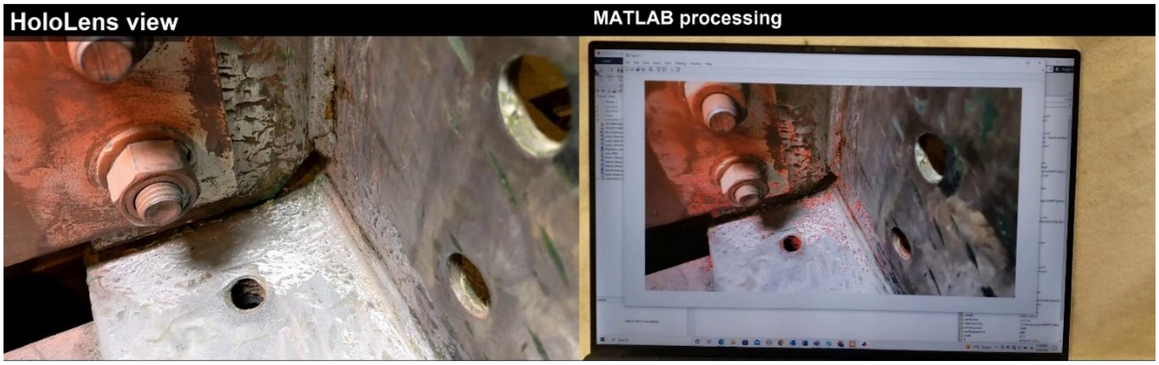


Figure 64 MATLAB crack detection algorithm running on the laptop/server

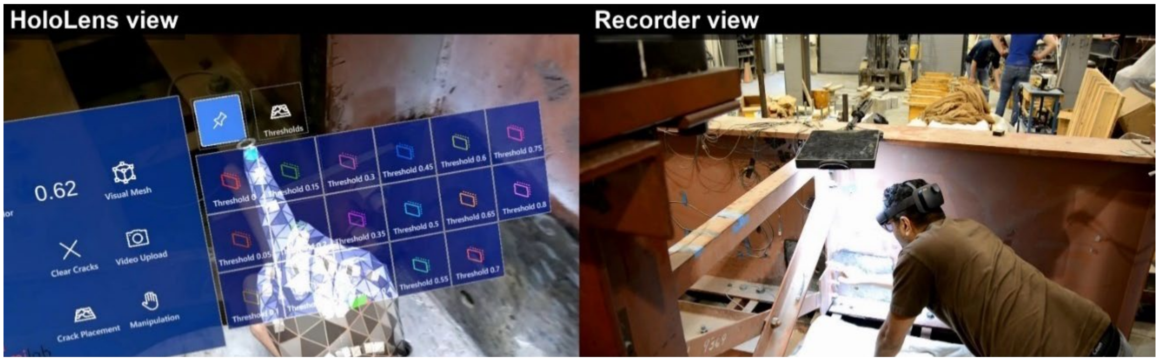


Figure 65 Inspector selecting the zero-threshold value to see the whole region analyzed for crack detection

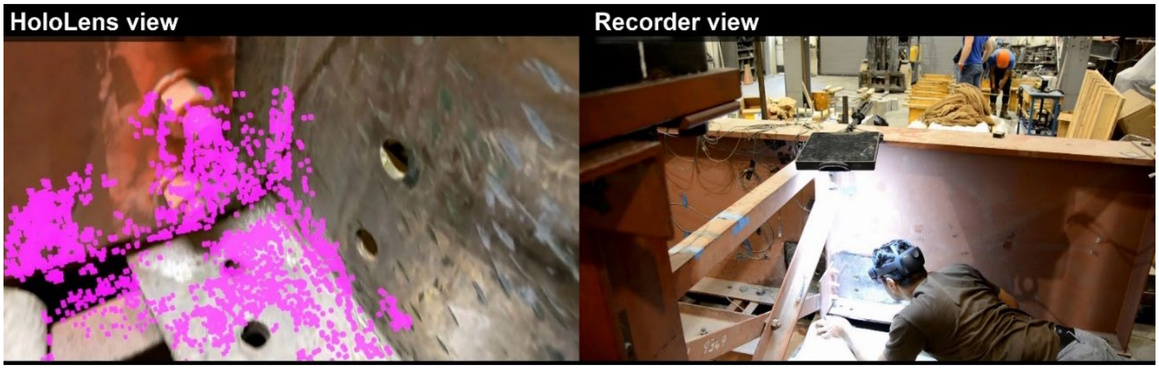


Figure 66 Inspecting the initial feature point cloud detected from the inspected region

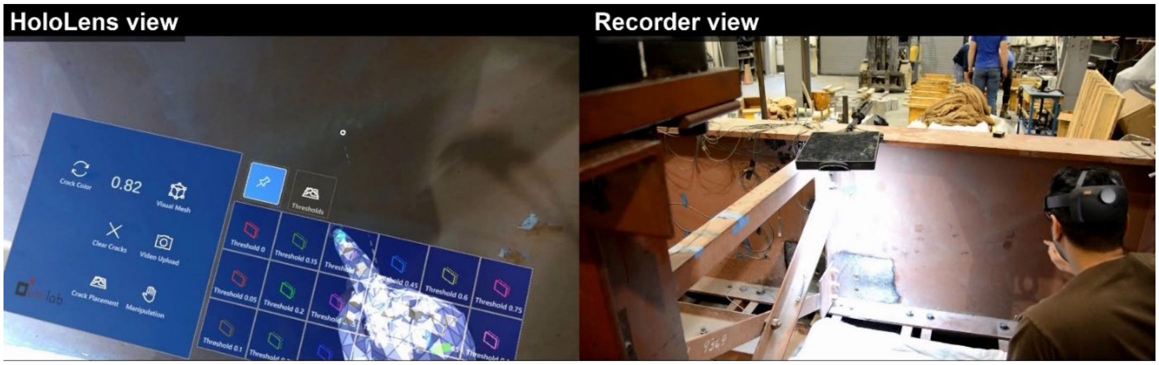


Figure 67 Inspector selecting a higher threshold value

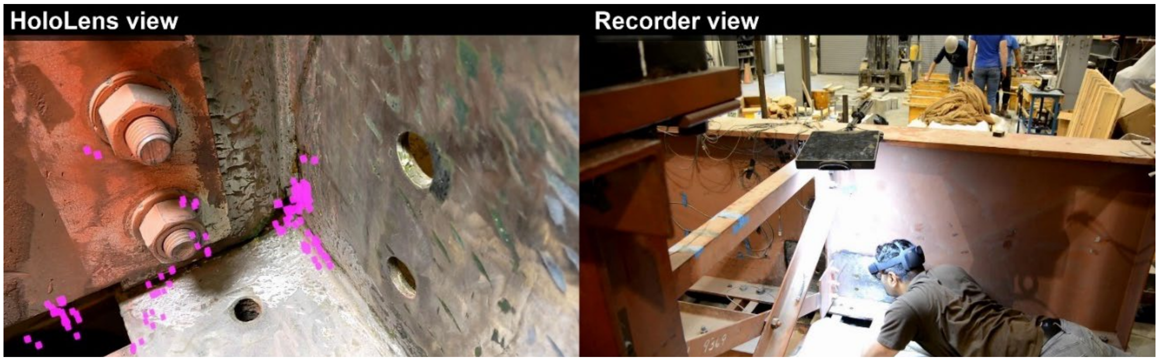


Figure 68 Inspecting the detected crack with a higher threshold value

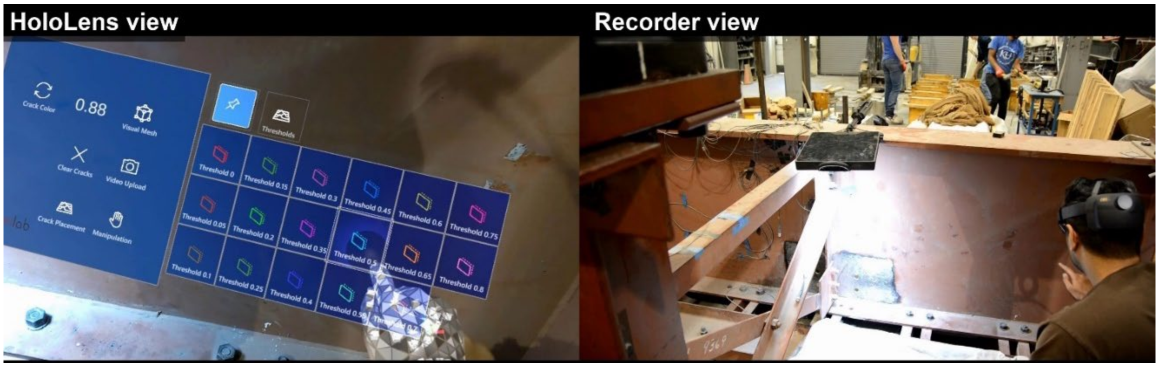


Figure 69 Inspector selecting the final threhosld value

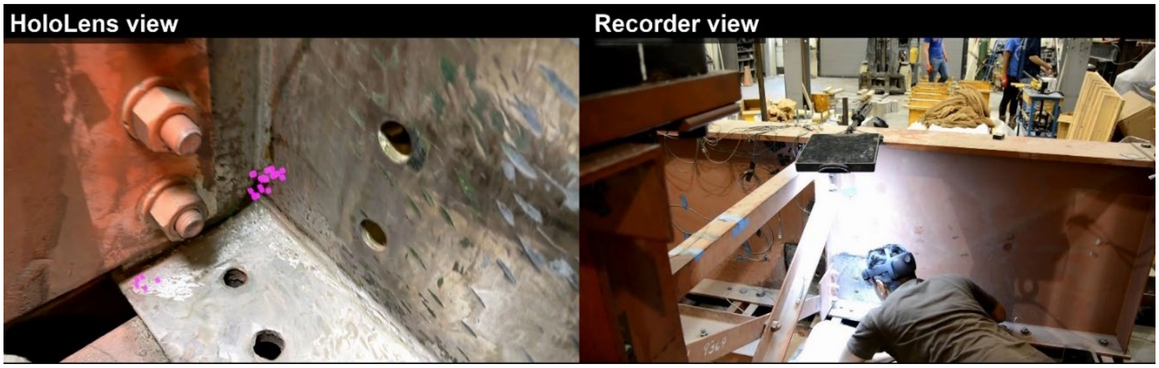


Figure 70 Inspecting the detected crack with the final threhosld value

Field Implementation

A field test was conducted to validate the developed fatigue crack detection algorithm integrated with AR application in near real-time. Figure 71 shows a group photo of the joint KU-UNM team in the field. Kansas DOT (KDOT) provided the bucket truck pictured in the background to bring the team members to reach target regions of the bridge girders for inspection.

Figure 71 The joint KU-UNM team under the test bridge in the field

The team visited a steel highway bridge designated the 70-105-41732-128 (eastbound) bridge on the I-70 highway (Figure 72) near Kansas City, Kansas. The bridge has various types of out-of-plane fatigue cracks, with one example illustrated in Figure 73.

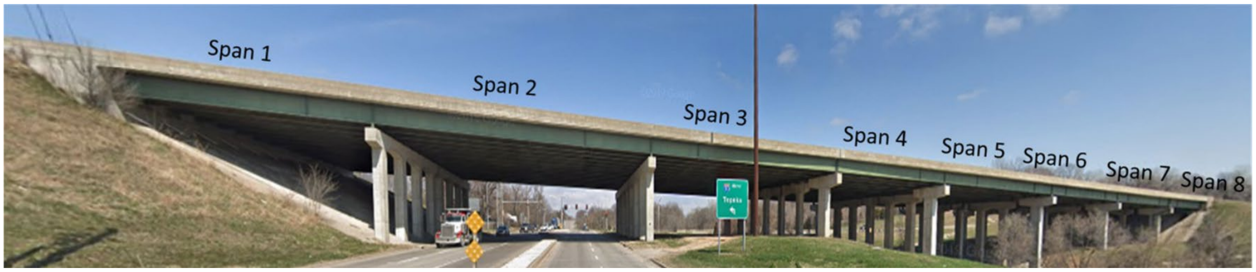


Figure 72 Span layout of the I-70 highway bridge with Span 3 selected for testing

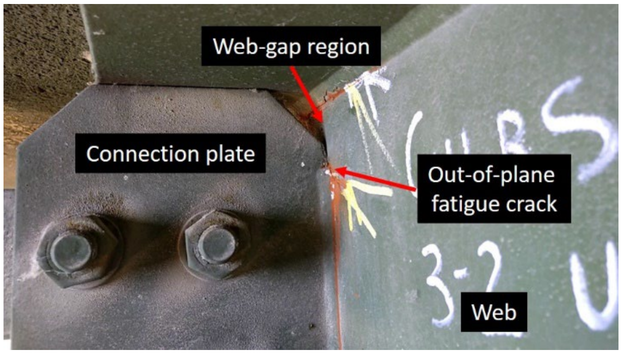


Figure 73 (a) Distortion-induced out-of-plane fatigue crack at the web-gap region, and (b) in-plane fatigue crack on the web

The inspection process requires the HL2, a laptop as server, and a Wi-Fi router that connect the HL2 with the server. Figure 74 shows the bridge inspector (PhD student) wearing the HL2, a laptop and a Wi-Fi router placed in a basket attached to the bucket indicating all required items can be easily carried with the inspector for actual bridge inspection. The first step in the inspection process is to connect the HL2 and laptop to a common Wi-Fi router to establish communication between them. Once the HL2 and laptop were connected to Wi-Fi, the inspector reaches to the fatigue prone web-gap region in span 3 of the bridge using the bucket truck. All steps of the inspection process are shown in Figure 75 to Figure 88, in which two views are presented: 1) recorder view on the left-hand side and 2) HoloLens view on the right-hand side. The recorder view is from the video recorded using a mobile phone videotaping the inspector, and the HoloLens view records the mix-reality view the inspector saw from the HL2. Both videos were synchronized and stitched together side-by-side for better visualization and illustration of the entire bridge inspection process using the developed AR software.

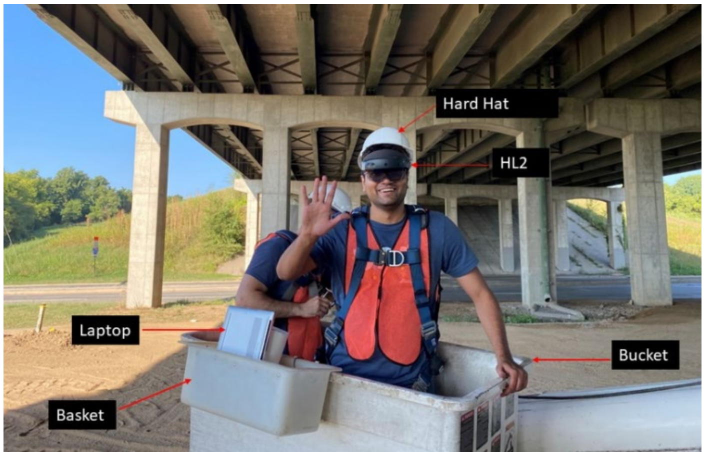


Figure 74 Bridge inspector preparing for the inspection process

First, as shown in Figure 5, inspector opens the HL2 menu. Then, the inspector opens the AR software named ‘SCFKU’ (Figure 76). After opening the AR software, the inspector moves the software menu at a convenient location (Figure 77). Figure 78 demonstrates the inspector selecting a proper length for video recording using the slide bar in the menu. For this particular test, the video recording length was selected to be 9 seconds. After that, the inspector hit the video upload button (Figure 79) to start video recording, and Figure 80 shows that inspector was looking at the web-gap region to record a video using the developed AR software. After the video was recorded, it was uploaded to the server where it was processed by MATLAB with the computer vision algorithm as shown in Figure 81. Upon completion of video processing, the results were transferred back to the HL2 and converted into holograms.

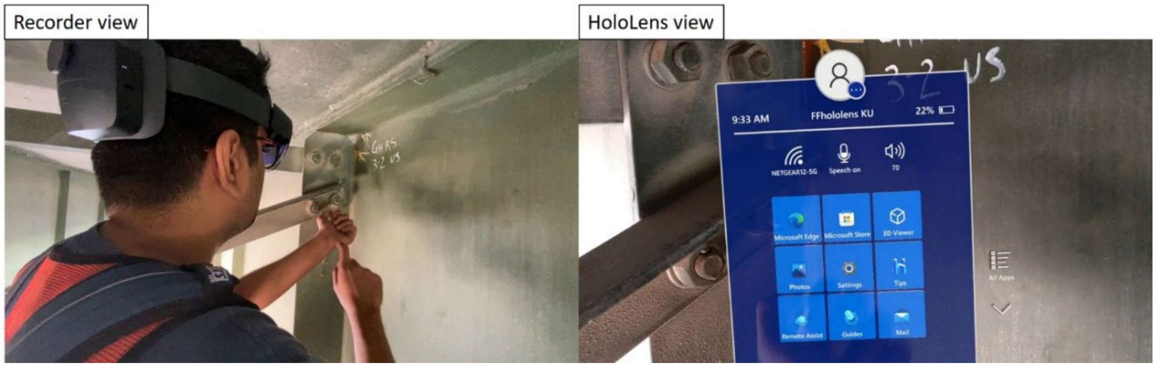


Figure 75 Opening the HL2 menu

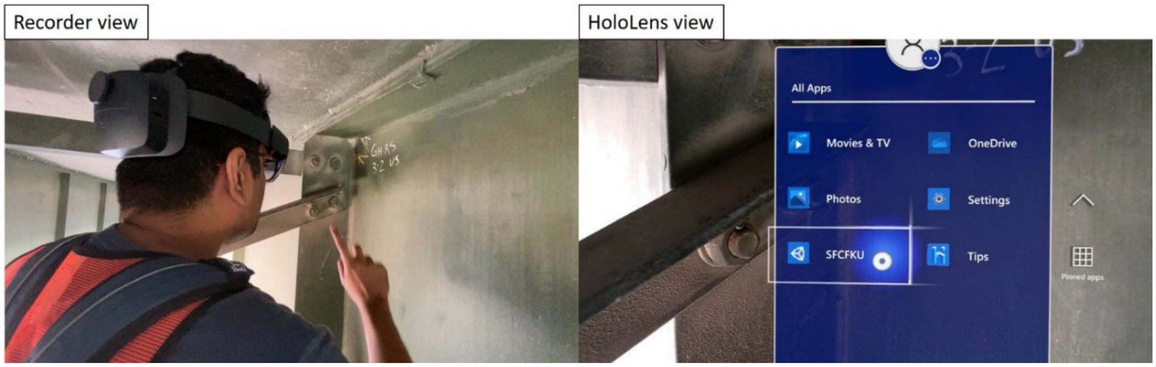


Figure 76 Opening the AR software

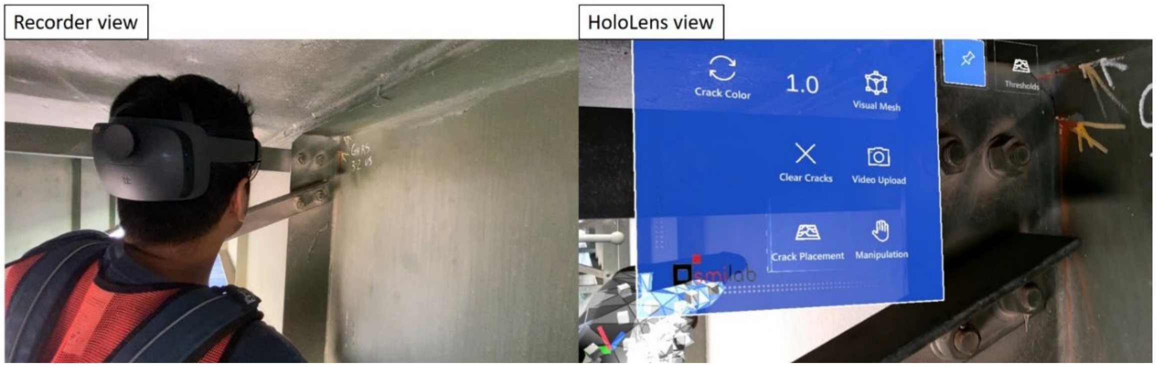


Figure 77 Moving the AR software menu to a convenient location

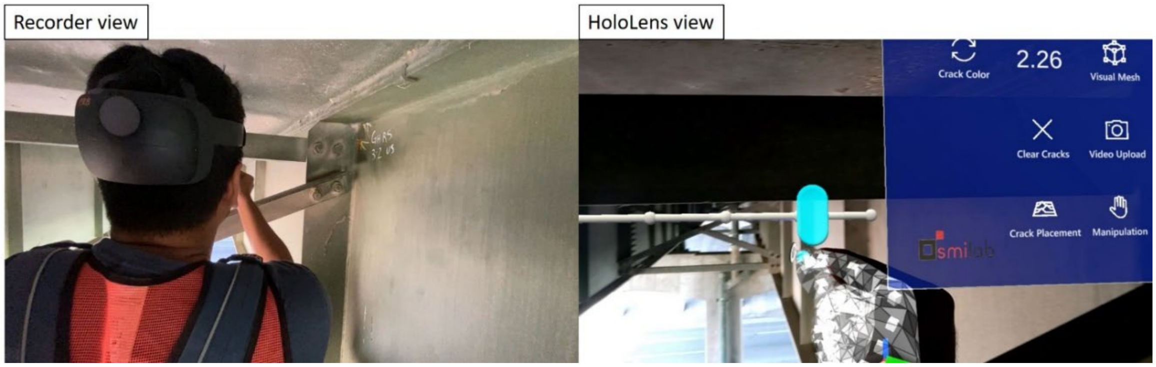


Figure 78 Selecting video recording length using the slide bar

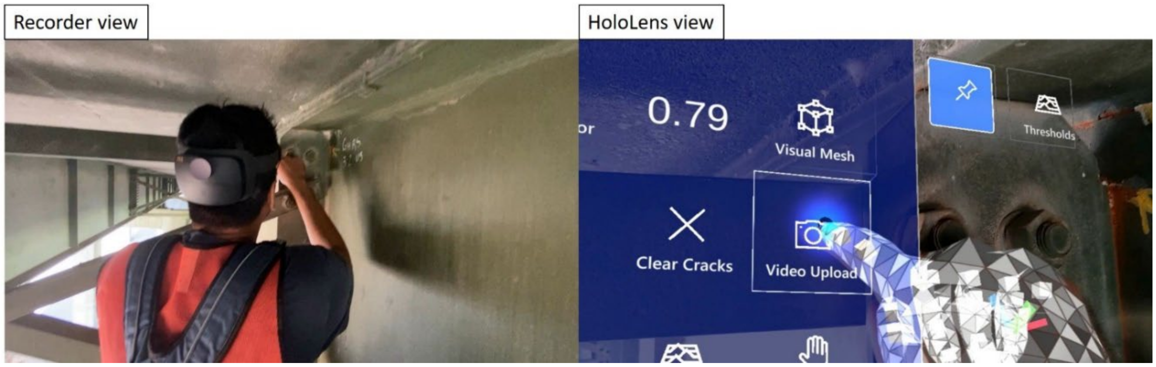


Figure 79 Hitting the video upload button to start recording

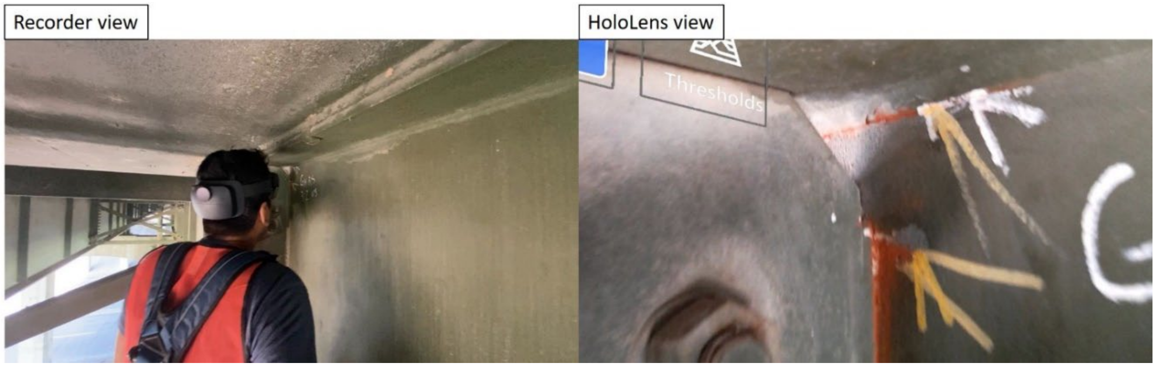


Figure 80 Recording a video for fatigue crack detection using the HL2

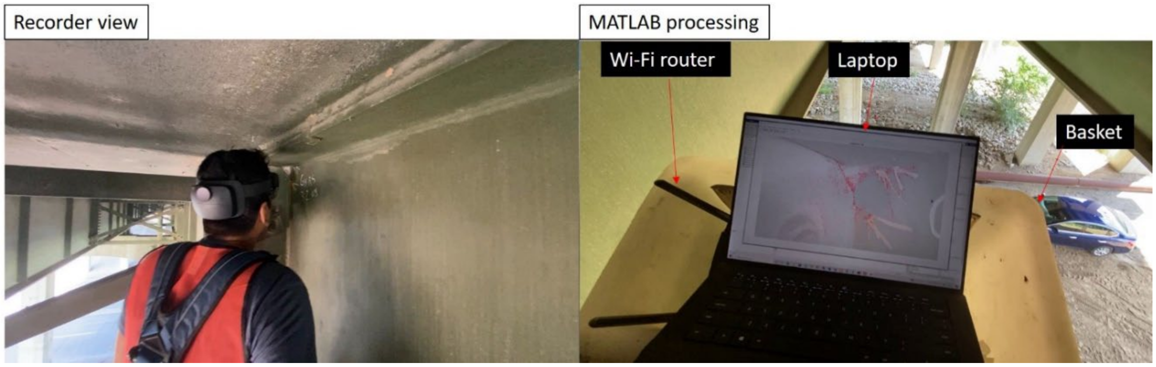


Figure 81 MATLAB crack detection algorithm running on the laptop/server simultaneously

The inspector first selected 0 threshold value (Figures 82 and 83) to inspect the initial point cloud from the region (Figure 84). To separate feature points associated with fatigue cracks from those associated with noncrack regions, the inspector selected a higher threshold value of 0.2 (Figure 85) to see the reduced feature points (Figure 86). The projected feature points with a threshold of 0.2 are grouped in multiple clusters, but they still cover the entire region. Hence, the inspector selected a higher threshold value of 0.38 (Figure 87) and inspected the crack detection result projected on the bridge girder (Figure 88). With a threshold value of 0.38, the feature points were clustered along the edge of the connection plate, including the top horizontal edge and the angled edge, indicating there are differential movements between the connection plate and the bridge girder under traffic loading.

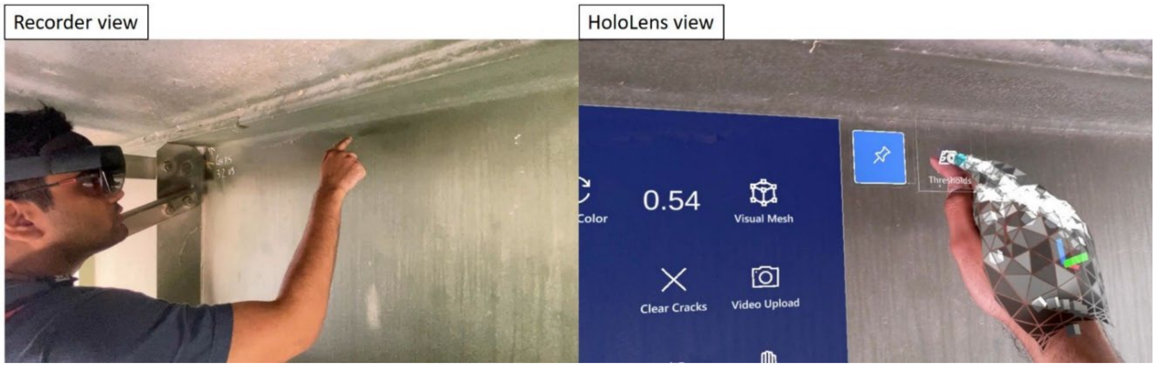


Figure 82 Enabling the threshold button after the MATLAB processing is done

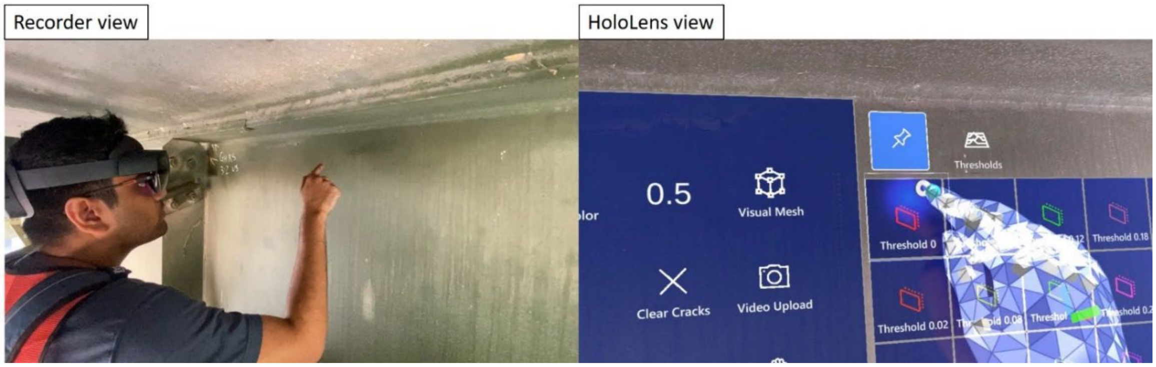


Figure 83 Hitting the button with zero threshold value to see the whole region analyzed for crack detection

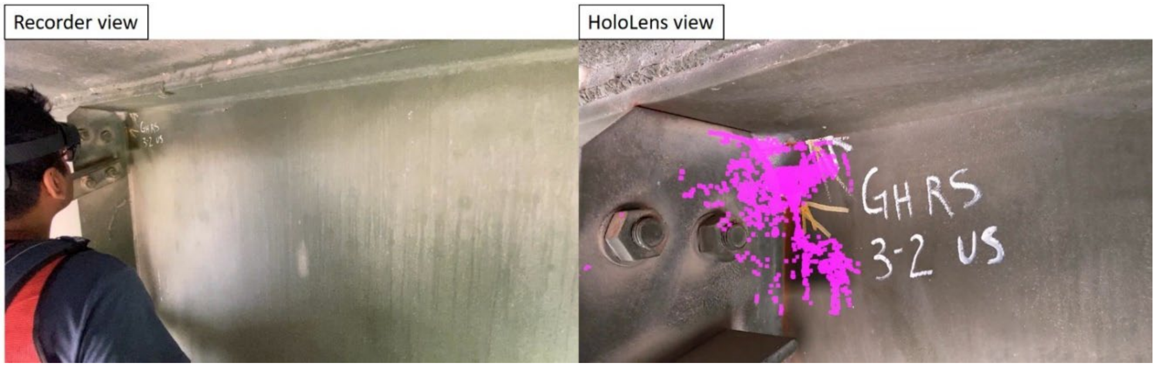


Figure 84 Inspecting the initial feature points cloud detected from the inspected region

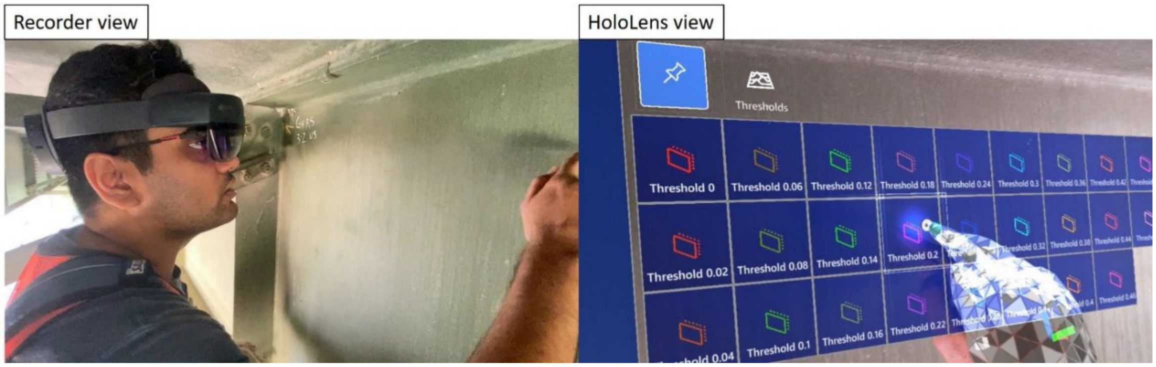


Figure 85 Selecting the threshold value of 0.2

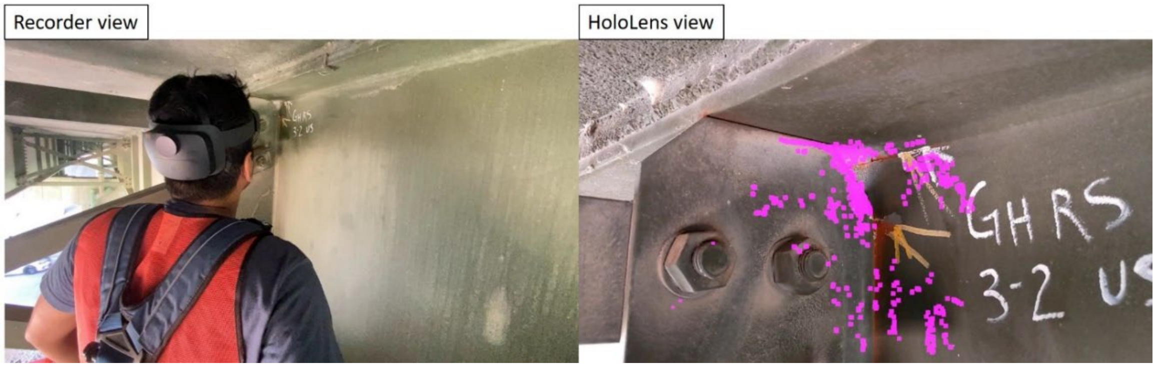


Figure 86 Inspecting the crack detection result for threshold value of 0.2

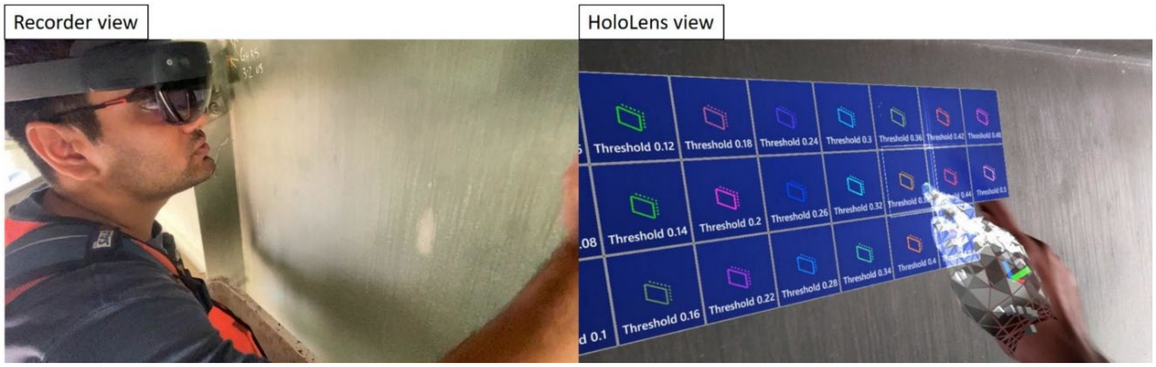


Figure 87 Selecting the threshold value of 0.38

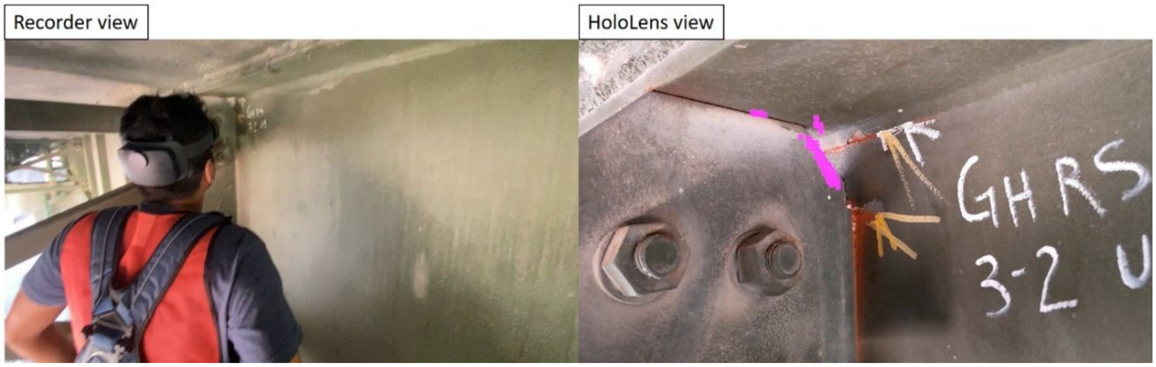


Figure 88 Inspecting the detected crack area with threshold value of 0.38

As shown in Figure 88, there are two distortion-induced fatigue cracks marked by previous KDOT inspections, including a vertical crack along the weld toe between the web and connection plate and a horizontal crack between the top flange and the web. However, the fatigue cracks were not picked up by the feature points due to their very small openings under the traffic loading, while the movement of the connection plate against the bridge girder, although very small as well, has been detected, as evidenced by the feature points clustered along the edges of the connection plate. The small opening of the fatigue crack could be attributed to the lack of adequate loading from the traffic during the field test and the high level of redundancy for the steel bridge. Overall, this field inspection offers practical insights for the proposed idea in the following three aspects:

1) The result proves the effectiveness of the proposed crack detection concept by detecting differential surface motions.
2) The result also highlights the challenge for field implementation especially for fatigue cracks that experience extremely small openings under operational loadings.
3) This field test also demonstrates the seamless flow of the developed AR-based software integrated with computer vision algorithms and many features to support human-centered bridge inspections.

PLANS FOR IMPLEMENTATION

The laboratory and field validations demonstrated the potential and revealed areas for improvement of the developed AR-based tool in facilitating human-machine collaboration for efficient fatigue crack inspections of steel bridges. First, both the laboratory and field implementations show that the AR software seamlessly integrated the computer vision algorithm into an interactive AR environment utilizing cloud database and holograms and achieved near real-time delivery of results for field inspections. Second, the laboratory tests using both 2D and 3D fatigue crack setups validated the effectiveness of the developed video-based computer vision algorithms to detect active fatigue cracks through surface motion tracking and analysis. Third, the field tests also revealed some practical challenges in detecting fatigue cracks with low opening and closing movements, possibly due to the lack of adequate loading from the traffic and high level of redundancy in the bridge structure. In addition, the AR software needs further development to fully support human-centered bridge crack detection, documentation, and decision making. The team has both short-term and long-term plans to continue to improve the developed AR-based fatigue crack inspection tool.

In the short term (12~18 months), the team will explore several ways to continue to improve the accuracy of the video-based fatigue crack detection algorithm. The field tests showed that the video-based algorithm needs more accurate surface motion tracking to detect very small fatigue crack movements. Feature tracking can be improved by 1) introducing additional surface texture possibly through sprayable patterns, 2) introducing more robust quality indicators for feature point detection and tracking, and 3) improving the video pixel resolution using a sperate high-resolution camera which can also be placed closer to the structure surface. In addition, the team plans to further develop the AR software, in particular the auto anchoring system (AAS) for its spatial accuracy in 3D positioning and dimension.

In the long term, the team would like to include other types of crack detection algorithms in the software to expand its damage detection capability. For example, in addition to the current video-based algorithms for fatigue crack detection, image-based algorithms for detecting deteriorations and damages for both steel and concrete structures can be included. Moreover, the team also plans to continue to reduce the computational cost of these algorithms such that they can be directly implemented within the mixed reality device using the limited processing power. Having the on-board processing capability will avoid the need for a separate server and the hotspot, greatly increasing the flexibility and efficiency of field inspection using the AR-based tool.

Finally, Kansas DOT has been very supportive for our field testing of the developed inspection tool. The team will continue to engage Kansas DOT and our other industry and government partners to seek further development and testing opportunities. In addition, the research team will continue to disseminate the results to other state DOTs through various venues such as the AASHTO (American Association of State Highway and Transportation Officials) committees and the TRB (Transportation Research Board) annual meetings.

CONCLUSIONS

Advances in computer vision have provided cost-effective, rapid, and non-contact solutions for structural health monitoring. The objective of this research project is to integrate computer-vision-based motion tracking and augmented reality techniques to empower bridge inspectors to perform fatigue crack detection, tracking, and characterization. To this end, fatigue crack detection algorithms have been developed based on feature point-based motion tracking of structural surface and AR-based human-centered platform has been created to project the detection result as holograms on top of the physical view. In this project, we proposed two strategies to enable fatigue crack detection using video-based computer vision. The first strategy is based on surface displacement tracking and utilizes global motion compensation approaches to remove camera motion from 2D and 3D videos. The second strategy is based on surface distance tracking which avoids the need for global motion compensation, hence demonstrating higher computational efficiency for near real-time crack detection. AR software integrating vision-based crack detection algorithm has been developed to enable bridge inspector perform fatigue crack inspection using mixed-reality device such as Microsoft HoloLens2. The conclusions of the research are summarized as follows:

1. Videos recorded from non-stationary platforms such as mixed-reality headsets have camera induced global motion. Laboratory testing based on C(T) and bridge girder specimens demonstrated that GMC is necessary to detect the fatigue crack successfully using feature point displacement-based method with unstable videos. The proposed GMC methods were effective at removing camera motion from 2D and 3D videos. The developed displacement-based crack detection algorithm successfully detects in-plane and out-of-plane fatigue cracks. However, the computation time for crack detection using GMC and displacement-based method is as high as 1.75 seconds per frame using a machine with intel i9 processor and 64 GB of RAM, making it challenging for near real-time fatigue crack inspection.
2. To facilitate near real-time crack detection, a novel feature point distance-based approach was developed. The differential motion due to crack movement is detected by analyzing distance between feature pairs on the surface, thus avoiding the need for GMC. The proposed method was validated in laboratory testing for near real-time fatigue crack detection in 2D and 3D videos. The computation time for the distance-based approach is 0.2 seconds per video frame, making it suitable for near real-time fatigue crack inspection. This approach was integrated with the AR software for fatigue crack detection.
3. A method was developed to quantify the detected crack using discrete feature points over continuous fatigue crack. The proposed crack detection approach had an IOU score as high as 0.77 implying the robustness of the crack detection algorithm in laboratory setting.
4. AR software integrating computer vision-based crack detection algorithm was developed for human-machine collaboration. A novel AAS system was developed through an invisible rectangle to overcome challenges in HL2's 3D mapping and to anchor the hologram in the right location and orientation. The AAS system is practical for inclined surfaces as well as surfaces perpendicular to user's line of sight. The AAS system was validated in both laboratory and field tests.
5. The developed AR software was successfully tested in the laboratory on a half-scale bridge girder specimen. This proof-of-concept testing shows that the developed AR-based fatigue crack inspection tool integrated with computer vision techniques is functioning seamlessly and smoothly, achieving the originally proposed goals.
6. The field test demonstrated the seamless flow of the developed AR-based software integrated with computer vision algorithms and the many features to support human-centered bridge inspections. The result proved the effectiveness of the proposed crack detection concept by detecting differential surface motions. The result also highlighted the challenge for field implementation especially for fatigue cracks that experience extremely small openings under operational loadings. Further research is needed to continue to improve the AR-based software and the crack detection algorithm.

REFERENCES

1. Abdel-Qader I, Abudayyeh O and Kelly ME, "Analysis of edge-detection techniques for crack identification in bridges," *Journal of Computing in Civil Engineering*, **17**(4), (2003): 255-263.
2. Al-Salih, H., Juno, M., Collins, W., Bennett, C., & Li, J.. Application of a digital image correlation bridge inspection methodology on geometrically complex bifurcated distortion-induced fatigue cracking. Fatigue & Fracture of Engineering Materials & Structures, 44(11), (2021), 3186-3201.
3. Amezquita-Sanchez JP and Adeli H, "Signal processing techniques for vibration-based health monitoring of smart structures," *Archives of Computational Methods in Engineering*, **23**(1), (2016): 1-15.
4. ARTBA, Bridge Report American Road and Transportation Builder's Association, 2020.
5. Asadollahi P and Li J, "Statistical analysis of modal properties of a cable-stayed bridge through long-term wireless structural health monitoring." *Journal of Bridge Engineering*, **22**(9), (2017): 04017051.
6. ASCE, "Infrastructure report card," *American Society of Civil Engineers*, https://infrastructurereportcard.org, (2021).
7. ASTM International, Standard test method for measurement of fracture toughness, ASTM International, (2015).
8. Bay H, Tuytelaars T and Gool LV, "Surf: Speeded up robust features," *European conference on computer vision,*Vol 3951, Berlin, Heidelberg, Germany, (2006), pp. 404-417.
9. Beardsley P, Torr P and Zisserman A, "3D model acquisition from extended image sequences," *European conference on computer vision,* Vol. 1065, Berlin, Heidelberg, Germany, (1996), pp. 683-695.
10. Bergen JR, Burt PJ, Hingorani R and Peleg S, "A three-frame algorithm for estimating two-component image motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**(9), (1992): 886-896.
11. Brown M and Lowe DG, "Automatic panoramic image stitching using invariant features," *International journal of computer vision*, **74**(1), (2007): 59-73.
12. Brown M and Lowe DG, "Recognising panoramas," *Proceedings of IEEE International Conference on Computer Vision (ICCV),* Vol 2, (2003), pp. 1218-1225.
13. Burt PJ, Hingorani R and Kolczynski RJ, "Mechanisms for isolating component patterns in the sequential analysis of multiple motion," *Proceedings of the IEEE Workshop on Visual Motion,* (1991), pp. 187-188.
14. Caetano E, Silva S and Bateira J, "A vision system for vibration monitoring of civil engineering structures," *Experimental Techniques*, **35**(4), (2011): 74-82.
15. Campbell LE, Connor RJ, Whitehead JM and Washer GA, "Benchmark for evaluating performance in visual inspection of fatigue cracking in steel bridges," *Journal of Bridge Engineering*, **25**(1), (2020): 04019128.
16. Cha YJ, Choi W and Büyüköztürk O, "Deep learning-based crack damage detection using convolutional neural networks," *Computer-Aided Civil and Infrastructure Engineering*, **32**(5), (2017): 361-378.
17. Coifman B, Beymer D, McLauchlan P and Malik J, "A real-time computer vision system for vehicle tracking and traffic surveillance," *Transportation Research Part C: Emerging Technologies*, **6**(4), (1998): 271-288.
18. Darrell T and Pentland A, "Robust estimation of a multi-layered motion representation," *Proceedings of the IEEE Workshop on Visual Motion,* (1991), pp. 173-174.
19. Dellenbaugh L., Kong X., Al-Salih H., Collins W. Bennett C., Li J. and Sutley E., "Development of Distortion-Induced Fatigue Crack Characterization Methodology using Digital Image Correlation," *Journal of Bridge Engineering*, 25(9), (2020), 04020063.

20. Dong C, Li L, Yan J, Zhang Z, Pan H and Catbas FN, “Pixel-level fatigue crack segmentation in large-scale images of steel structures using an encoder–decoder network,” *Sensors*, **21**(12), (2021): 4135.
21. Dung CV, Sekiya H, Hirano S, Okatani T and Miki C, “A vision-based method for crack detection in gusset plate welded joints of steel bridges using deep convolutional neural networks,” *Automation in Construction*, **102,** (2019): 217-229.
22. Feng D and Feng MQ, “Experimental validation of cost-effective vision-based structural health monitoring,” *Mechanical Systems and Signal Processing*, **88,** (2017): 199-211.
23. Feng MQ, Fukuda Y, Feng D and Mizuta M, “Nontarget vision sensor for remote measurement of bridge dynamic response,” *Journal of Bridge Engineering*, **20**(12), (2015): 04015023.
24. Glišić B, Yao Y, Tung STE, Wagner S, Sturm JC and Verma N, “Strain sensing sheets for structural health monitoring based on large-area electronics and integrated circuits,” *Proceedings of the IEEE*, **104**(8), (2016): 1513-1528.
25. Goldstein A and Fattal R, “Video stabilization using epipolar geometry,” *ACM Transactions on Graphics (TOG)*, **31**(5), (2012): 1-10.
26. Grundmann M, Kwatra V, Castro D and Essa I, “Calibration-free rolling shutter removal,” *IEEE international conference on computational photography (ICCP),* Seattle, USA, (2012), pp. 1-8.
27. Haghani R, Al-Emrani M and Heshmati M, “Fatigue-prone details in steel bridges,” *Buildings*, **2**(4), (2012): 456-476.
28. Harris C and Stephens M, “A combined corner and edge detector,” *Alvey vision conference,* **15**(50), (1988): 10-5244.
29. Holusha J and Chang K , “Engineers see dangers in aging infrastructure,” *New York Times*, August 2, (2007), http://www.nytimes.com/2007/08/02/us/01cnd-engineer.html.
30. Igarashi T, Moscovich T and Hughes JF, “As-rigid-as-possible shape manipulation,” *ACM transactions on Graphics (TOG)*, **24**(3), (2005): 1134-1141.
31. Ihn JB and Chang FK, “Detection and monitoring of hidden fatigue crack growth using a built-in piezoelectric sensor/actuator network: I. Diagnostics,” *Smart materials and structures*, **13**(3), (2004): 609.
32. Irani M and Anandan P, “A unified approach to moving object detection in 2D and 3D scenes,” *IEEE transactions on pattern analysis and machine intelligence*, **20**(6), (1998): 577-589.
33. Iyer S and Sinha SK, “Segmentation of pipe images for crack detection in buried sewers,” *Computer-Aided Civil and Infrastructure Engineering*, **21**(6), (2006): 395-410.
34. Jahanshahi MR and Masri SF, “A new methodology for non-contact accurate crack width measurement through photogrammetry for automated structural safety evaluation,” *Smart materials and structures*, **22**(3), (2013): 035019.
35. Jang S, Li J and Spencer Jr BF, “Corrosion estimation of a historic truss bridge using model updating,” *Journal of Bridge engineering*, **18**(7), (2013): 678-689.
36. Kalal Z, Mikolajczyk K and Matas J, “Forward-backward error: Automatic detection of tracking failures,” *20th international conference on pattern recognition,* (2010), pp. 2756-2759.
37. Kharroub S, Laflamme S, Song C, Qiao D, Phares B and Li J, “Smart sensing skin for detection and localization of fatigue cracks,” *Smart Materials and Structures*, **24**(6), (2015): 065004.
38. Khuc T and Catbas FN, “Computer vision-based displacement and vibration monitoring without using physical target on structures,” *Structure and Infrastructure Engineering*, **13**(4), (2017): 505-516.
39. Kong X and Li J, “Vision-based fatigue crack detection of steel structures using video feature tracking,” *Computer-Aided Civil and Infrastructure Engineering*, **33**(9), (2018): 783-799.
40. Kong X and Li J, “Vision-based fatigue crack detection of steel structures using video feature tracking,” *Computer-Aided Civil and Infrastructure Engineering*, **33**(9), (2018): 783-799.
41. Kong X, Li J, Bennett C, Collins W and Laflamme S, “Numerical simulation and experimental validation of a large-area capacitive strain sensor for fatigue crack monitoring,” *Measurement Science and Technology*, **27**(12), (2016): 124009.

42. Kong X, Li J, Collins W, Bennett C, Laflamme S and Jo H, “A large-area strain sensing technology for monitoring fatigue cracks in steel bridges,” *Smart Materials and Structures*, **26**(8), (2017): 085024.
43. Lawn JM and Cipolla R, “Robust egomotion estimation from affine motion parallax,” *European Conference on Computer Vision,* Vol. 800, Berlin, Heidelberg, Germany, (1994), pp. 205-210.
44. Lee YC, Tseng KW, Chen YT, Chen CC, Chen CS and Hung YP, “3D Video Stabilization with Depth Estimation by CNN-based Optimization,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* (2021), pp. 10621-10630.
45. Lee, J., Jeong, S., Lee, Y. J., & Sim, S. H.. “Stress Estimation Using Digital Image Correlation with Compensation of Camera Motion-Induced Error, ” *Sensors*, 19(24), (2019), 5503.
46. Liu F, Gleicher M, Wang J, Jin H and Agarwala A, “Subspace video stabilization,” *ACM Transactions on Graphics (TOG)*, **30**(1), (2011): 1-10.
47. Liu S, Yuan L, Tan P and Sun J, “Bundled camera paths for video stabilization,” *ACM Transactions on Graphics (TOG)*, **32**(4), (2013): 1-10.
48. Longuet-Higgins HC and Prazdny K, “The interpretation of a moving retinal image,” *Proceedings of the Royal Society of London. Series B. Biological Sciences*, **208**(1173), (1980): 385-397.
49. Lowe DG, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, **60**(2), (2004): 91-110.
50. Lucas BD and Kanade T, An iterative image registration technique with an application to stereo vision, Vol. 81, (1981), pp. 674-679.
51. Malek, Kaveh, Ali Mohammadkhorasani, and Fernando Moreu. "Methodology to integrate augmented reality and pattern recognition for crack detection." Co*mputer-Aided Civil and Infrastructure Engineering,* (2022).
52. Mascarenas, David DL, JoAnn P. Ballor, Oscar L. McClain, Miranda A. Mellor, Chih-Yu Shen, Brian Bleck, John Morales et al. "Augmented reality for next generation infrastructure inspections." Structural Health Monitoring 20, no. 4 (2021): 1957-1979.
53. Moreu, Fernando, Kaveh Malek, Elijah Wyckoff, and Ali Mohammadkhorasani. "Augmented Reality: Existing Capabilities and Future Opportunities." TR News 340 (2022).
54. Napolitano, Rebecca, Zachary Liu, Carl Sun, and Branko Glisic. "Combination of image-based documentation and augmented reality for structural health monitoring and building pathology." Frontiers in Built Environment 5 (2019): 50. 10.3389/fbuil.2019.00050
55. Nishikawa T, Yoshida J, Sugiyama T and Fujino Y, “Concrete crack detection by multiple sequential image filtering,” *Computer-Aided Civil and Infrastructure Engineering*, **27**(1), (2012): 29-47.
56. Outay, Fatma, Nafaa Jabeur, Hedi Haddad, and Zied Bouyahia. "A Framework for Self-Inspection Buildings Based on Augmented Reality Agents." Computing and Informatics 41, no. 2 (2022): 479-502.
57. Park JW, Lee JJ, Jung HJ and Myung H, “Vision-based displacement measurement method for high-rise building structures using partitioning approach,” *NDT & E International*, **43**(7), (2010): 642-647.
58. Rieger JH and Lawton DT, “Processing differential image motion,” *J. Opt. Soc. Am. A*, **2**(2), (1985): 354-359.
59. Roberts T and Talebzadeh M, "Acoustic emission monitoring of fatigue crack propagation," *Journal of constructional steel research*, **59**(6), (2003): 695-712.
60. Rosten E and Drummond T, “Fusing points and lines for high performance tracking,” *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, Vol. 2, (2005), pp. 1508-1515.
61. Sadhu, Ayan, Jack E. Peplinski, Ali Mohammadkhorasani, and Fernando Moreu. "A Review of Data Management and Visualization Techniques for Structural Health Monitoring Using BIM and Virtual or Augmented Reality." Journal of Structural Engineering 149, no. 1 (2023): 03122006.
62. Safdarnejad SM, Liu X and Udpa L, “Robust Global Motion Compensation in Presence of Predominant Foreground,” *Proceedings of the British Machine Vision Conference (BMVC)*, (2015), pp. 21.1-21.11.

63. Schaefer S, McPhail T and Warren J, “Image deformation using moving least squares,” *ACM Trans. Graph,* **25**(3), (2006): 533-540.
64. Schaffalitzky F and Zisserman A, “Multi-view matching for unordered image sets, or “How do I organize my holiday snaps?”,” *European conference on computer vision,* Vol. 2350, Berlin, Heidelberg, Germany, (2002), pp. 414-431.
65. Shi J, “Good features to track,” Proceedings of IEEE conference on computer vision and pattern recognition (CVPR), (1994), pp. 593-600.
66. Shi Z, Shi F, Lai WS, Liang CK and Liang Y, “Deep Online Fused Video Stabilization,” *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision,* (2022), pp. 1250-1258.
67. Sirca Jr GF and Adeli H, "System identification in structural engineering," *Scientia Iranica*, **19**(6), (2012): 1355-1364.
68. Snavely N, Seitz SM and Szeliski R, “Photo tourism: exploring photo collections in 3D,” *ACM siggraph,* 25(3), (2006): 835-846.
69. Ta DN, Chen WC, Gelfand N and Pulli K, “Surftrac: Efficient tracking and continuous object recognition using local feature descriptors,” *IEEE conference on computer vision and pattern recognition (CVPR),* (2009), pp. 2937-2944.
70. Tomasi C and Kanade T, “Detection and tracking of point,” *Int J Comput Vis*, **9,** (1991): 137-154.
71. Torr PH and Murray DW, “Stochastic motion clustering,” *European Conference on Computer Vision,* Vol. 801, Berlin, Heidelberg, Germany, (1994), pp. 328-337.
72. Torr PH, Zisserman A and Maybank SJ, “Robust detection of degenerate configurations while estimating the fundamental matrix,” *Computer vision and image understanding*, **71**(3), (1998): 312-333.
73. Wahbeh AM, Caffrey JP and Masri SF, “A vision-based approach for the direct measurement of displacements in vibrating systems,” *Smart materials and structures*, **12**(5), (2003): 785.
74. Wang, Shaohan, Sakib Ashraf Zargar, and Fuh-Gwo Yuan. "Augmented reality for enhanced visual inspection through knowledge-based deep learning." Structural Health Monitoring 20, no. 1 (2021): 426-442.
75. Xu, Jiaqi, Derek Doyle, and Fernando Moreu. "State of the Art of Augmented Reality (AR) Capabilities for Civil Infrastructure Applications." arXiv preprint arXiv:2110.08698 (2021).
76. Yamaguchi T, Nakamura S, Saegusa R and Hashimoto S, “Image-based crack detection for real concrete surfaces,” *IEEJ Transactions on Electrical and Electronic Engineering*, **3**(1), (2008): 128-135.
77. Yeum CM and Dyke SJ, “Vision-based automated crack detection for bridge inspection,” *Computer-Aided Civil and Infrastructure Engineering*, **30**(10), (2015): 759-770.
78. Yoon H, Elanwar H, Choi H, Golparvar-Fard M and Spencer Jr BF, “Target-free approach for vision-based structural system identification using consumer-grade cameras,” *Structural Control and Health Monitoring*, **23**(12), (2016): 1405-1416.
79. Yu J and Ramamoorthi R, “Learning video stabilization using optical flow,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* (2020), pp. 8159-8167.
80. Zoghlami I, Faugeras O and Deriche R, “Using geometric corners to build a 2D mosaic from a set of images,” *Proceedings of IEEE computer society conference on computer vision and pattern recognition,* (1997), pp. 420-425.

APPENDIX A: RESEARCH RESULTS

Sidebar Info
Program Steering Committee: NCHRP IDEA Program Committee
Month and Year: December, 2022
Title: Fatigue Crack Inspection Using Computer Vision and Augmented Reality
Project Number: NCHRP IDEA 20-30/IDEA 223
Start Date: January, 1, 2020
Completion Date: December, 31, 2021
Product Category: New tool
Principal Investigator: Jian Li, Associate Professor
Institution: University of Kansas
E-Mail: jianli@ku.edu
Phone: +1 (785) 864 6850

TITLE: Fatigue Crack Inspection Using Computer Vision and Augmented Reality
SUBHEAD: This project proposed human-machine collaboration by integrating computer vision and augmented reality for robust fatigue crack inspection of steel bridges.

WHAT WAS THE NEED?

Fatigue cracks developed under repetitive traffic loads are a major threat to maintaining the structural integrity of steel bridges. In fact, it is believed that 50~90% of all mechanical failures in metallic structures are fatigue-related. The collapse of the Silver Bridge and the Mianus River Bridge are among the most well-known bridge failures directly caused by fatigue. Human visual inspection is currently the de facto approach for fatigue crack detection because as of today, administrations strongly rely on the experience of the experts to maintain their bridges. Federal mandate requires typical visual inspection of fatigue cracks. However, due to human limitations and the complex nature of bridge structures, fatigue crack inspections are time consuming, labor intensive, and lack reliability. Although non-destructive testing (NDT) techniques such as ultrasonic testing and acoustic emission have been used as supplemental methods to human visual inspection, they require complex testing equipment, and thus are not broadly used. As a result, inspecting the large steel bridge inventory in the United States remains a great challenge due to the lack of a human-centered, efficient, and cost-effective methodology for detecting, tracking, and documenting fatigue cracks. On the other hand, if crack inspections could inform the inspector in the field, more reliable, efficient, and accurate assessment of the inventory could be achieved and documented.

WHAT WAS OUR GOAL?

The objective of the research was to develop augmented reality software integrating vision-based crack detection algorithm to enable human-machine collaboration for bridge inspectors to perform robust fatigue crack detection, characterization, tracking, and documentation in the field.

WHAT DID WE DO?

Global motion compensation methods were first proposed to remove camera motion from 2D and 3D videos recorded using non-stationary platforms such as mixed reality headsets. Global motion compensation is necessary to accurately analyze surface motion for fatigue crack detection using feature point displacements. The displacement-based approach of crack detection was validated in the laboratory setup for fatigue crack detection in 2D and 3D videos. However, the computational cost was very high making it unsuitable for near real-time crack detection. To perform near real-time crack detection, a novel approach based on tracking distance between feature point pairs was developed to avoid the need for GMC. As a result, the computation time was significantly reduced, making it suitable for near-real time crack detection. Augmented reality software was developed to project the crack detection result as holograms on top of the real-world structure. An innovative automatic anchoring system (AAS) was developed to anchor the hologram in the right location and orientation. The software also enables human-machine interaction for selecting the optimal crack detection result.

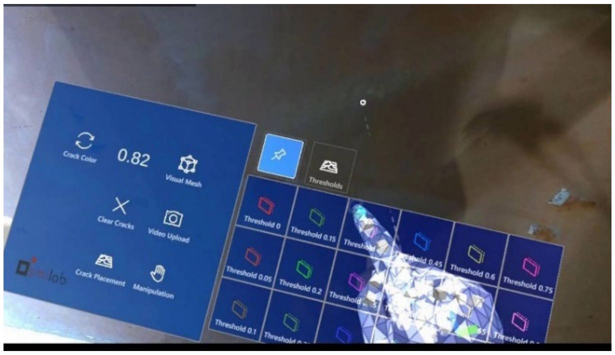


Figure 1: The developed AR software interface

WHAT WAS THE OUTCOME?

The validation of proposed integrated tool was performed both on individual components and as a whole system. The vision-based fatigue crack detection approach was validated in the laboratory for in-plane and out-of-plane fatigue crack detection. The feature point displacement-based approach successfully detected fatigue crack after applying the global motion compensation. The novel distance-based crack detection method achieved higher accuracy of crack detection in significantly less time compared to the displacement-based algorithm. The accuracy of the detected crack was as high as 0.77 indicating effectiveness of the proposed algorithm. The automatic anchoring system was first validated through field testing on a simulated crack. The system successfully anchored the hologram of the detected crack on the created 2D and 3D crack benchmarks. The software as a whole was tested in the laboratory bridge girder setup. This proof-of-concept testing shows that the developed AR-based fatigue crack inspection tool integrated with computer vision techniques is functioning seamlessly and smoothly, achieving the originally proposed goals. The field test further demonstrated the seamless flow of the developed AR-based software integrated with computer vision algorithms and many features to support human-centered bridge inspections. The result proves the effectiveness of the proposed crack detection concept by detecting differential surface motions. The field testing also highlights the challenge for field implementation especially for fatigue cracks that experience extremely small openings under operational loadings.

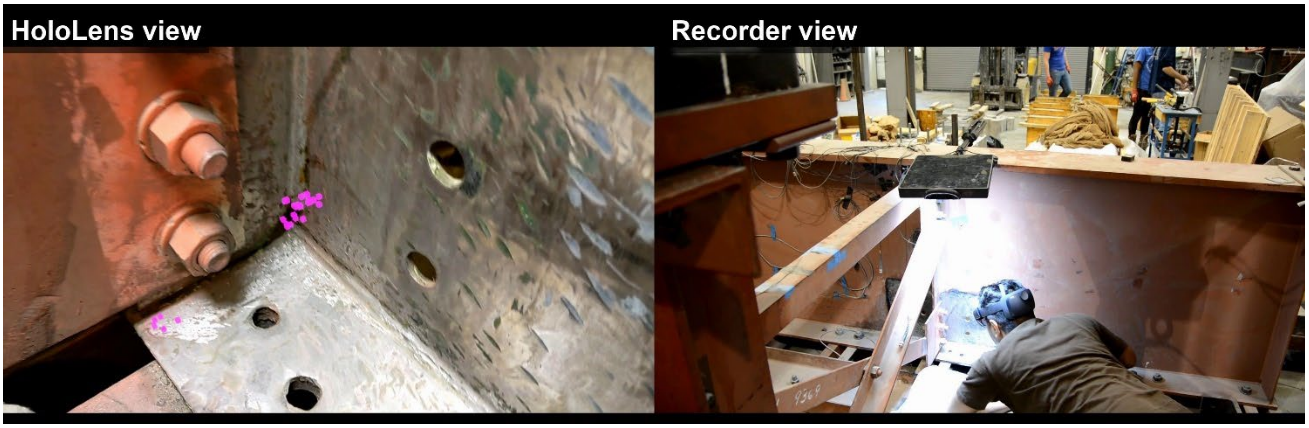


Figure 2: Fatigue crack inspection using developed AR-based tool

WHAT IS THE BENEFIT?

This project developed and validated a human-machine collaboration to empower bridge inspectors to perform robust fatigue crack detection, characterization, and tracking. It can help bridge inspectors improve the efficiency and accuracy of fatigue crack inspection in the field.