VTRC
Virginia Transportation Research Council

We Bring Innovation to Transportation

Supplemental Data Collection and Processing for Bridge Safety Inspections Utilizing Mixed Reality and Artificial Intelligence

https://vtrc.virginia.gov/media/vtrc/vtrc-pdf/vtrc-pdf/25-R4.pdf

RODRIGO SARLO, Ph.D.
Assistant Professor

ALAN SMITH
Graduate Student

Civil and Environmental Engineering
Virginia Tech

JOSEPH GABBARD JR., Ph.D.0
Professor

JOHN LUKSAS
Graduate Student

Industrial Systems Engineering
Virginia Tech

Final Report VTRC 25-R4

VIRGINIA TRANSPORTATION RESEARCH COUNCIL
530 Edgemont Road, Charlottesville, VA 22903-2454
vtrc.virginia.gov

Standard Title Page - Report on Federally Funded Project

1. Report No.: FHWA/VTRC 25-R4	2. Government Accession No.:	3. Recipient's Catalog No.:
4. Title and Subtitle: Supplemental Data Collection and Processing for Bridge Safety Inspections Utilizing Mixed Reality and Artificial Intelligence		5. Report Date: January 2025
		6. Performing Organization Code: VTRC
7. Author(s): Rodrigo Sarlo, Ph.D., Joseph Gabbard Jr., Ph.D., Alan Smith, John Luksas		8. Performing Organization Report No.: VTRC 25-R4
9. Performing Organization and Address: Virginia Tech Blacksburg, VA 24061 Va. Transportation Research Council 530 Edgemont Rd Charlottesville, VA 22903		10. Work Unit No. (TRAIS):
		11. Contract or Grant No.: 120109
12. Sponsoring Agencies' Name and Address: Virginia Department of Transportation 1401 E. Broad Street Richmond, VA 23219 Federal Highway Administration 400 North 8th Street, Room 750 Richmond, VA 23219-4825		13. Type of Report and Period Covered: Final
		14. Sponsoring Agency Code:
15. Supplementary Notes: This is an SPR-B report		

16. Abstract:

This research work focused on evaluating both the technical and design elements necessary to perform augmented reality (AR) assisted bridge inspections. The inspection prototype developed by this project aims to demonstrate the potential benefits and limitations of incorporating head-worn displays into the inspection workflow. By virtue of switching to an electronic data collection method, many common quality assurance issues can be mitigated. Furthermore, AR has an additional advantage over tablet-based solutions in its hands-free operation, which can allow for increased safety and flexibility. Furthermore, data collected in an AR system is georeferenced locally to the bridge and shown to be adequately accurate to allow navigation to previously documented defects during the inspection. Finally, and most importantly, AR serves as a central platform within which many up-and-coming digital tools (such as artificial intelligence) can be integrated. To better understand this potential, the work also considered the use of AI for a specific use case (concrete crack measurements).

A prototype application was developed using the Microsoft HoloLens 2 headset, but the principles employed could be translated to other similar mixed reality platforms. The prototype targeted efficient integration into the entire field inspection workflow. Three key functionalities were identified and addressed as main challenges to this holistic inspection approach: 1) a structured data entry interface, 2) georeferenced annotations for visualizing historical data, and 3) computer vision for automated defect labeling. A unique research focus was how to address these challenges in imperfect, real-world field scenarios involving both novice and professional bridge inspectors. To this end, the research team explored a range of AR interaction techniques of various degrees of automation, drawing on human-computer interaction principles. The study's evaluation metrics focused on measurement accuracy, time on task, as well as the tool's impact on perceived workload and usability.

Overall, the research found that performing virtual measurements within the AR environment resulted in similar usability as traditional measurements (i.e. tape measure and crack ruler), while significantly reducing the cognitive workload. Results were mixed in terms of time on task and accuracy. In general, more automated measurement processes significantly reduced time on task, but tended to decrease accuracy. One key finding was that the best combination of metrics was found using hybrid automation approaches, where the inspector was able to refine automated defect labels. In terms of added value, less cognitive load could mean less errors in documentation process, saving time at the office. This would be added to the time saving in report generation due to the virtue of digital data collection. The platform also provides the ability to collect much more granular data, which could help improve overall data quality.

The research team was encouraged by the reactions of a small set of VDOT bridge inspectors during the field validation of the final prototype. These inspectors were impressed by the technology's capabilities and intuitive workflow, voicing support for the future potential for bridge inspection. While the performed work supports the functionality of the platform over the entire field inspection workflow, the current prototype still requires additional development to be fully implementation ready.

Supplemental files can be found at https://library.vdot.virginia.gov/vtrc/supplements

17 Key Words: Augmented Reality, Computer Vision, Bridge Inspection, Human-Computer Interaction, Automated Defect Detection, User Experience	18. Distribution Statement: No restrictions. This document is available to the public through NTIS, Springfield, VA 22161.

19. Security Classif. (of this report): Unclassified	20. Security Classif. (of this page): Unclassified	21. No. of Pages: 58	22. Price:

FINAL REPORT

SUPPLEMENTAL DATA COLLECTION AND PROCESSING FOR BRIDGE SAFETY INSPECTIONS UTILIZING MIXED REALITY AND ARTIFICIAL INTELLIGENCE

Rodrigo Sarlo, Ph.D.
Assistant Professor
Civil and Environmental Engineering
Virginia Tech

Joseph Gabbard Jr., Ph.D.
Professor
Industrial Systems Engineering
Virginia Tech

Alan Smith
Graduate Research Assistant
Civil and Environmental Engineering
Virginia Tech

John Luksas
Graduate Research Assistant
Industrial Systems Engineering
Virginia Tech

VTRC Project Manager
Amir Behravan, Ph.D., P.E., Virginia Transportation Research Council

In Cooperation with the U.S. Department of Transportation
Federal Highway Administration

Virginia Transportation Research Council
(A partnership of the Virginia Department of Transportation
and the University of Virginia since 1948)

Charlottesville, Virginia

January 2025
VTRC 25-R4

DISCLAIMER

The project that is the subject of this report was done under contract for the Virginia Department of Transportation, Virginia Transportation Research Council. The contents of this report reflect the views of the author(s), who is responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the Virginia Department of Transportation, the Commonwealth Transportation Board, or the Federal Highway Administration. This report does not constitute a standard, specification, or regulation. Any inclusion of manufacturer names, trade names, or trademarks is for identification purposes only and is not to be considered an endorsement.

Each contract report is peer reviewed and accepted for publication by staff of the Virginia Transportation Research Council with expertise in related technical areas. Final editing and proofreading of the report are performed by the contractor.

ABSTRACT

This research work focused on evaluating both the technical and design elements necessary to perform augmented reality (AR) assisted bridge inspections. The inspection prototype developed by this project aims to demonstrate the potential benefits and limitations of incorporating head-worn displays into the inspection workflow. By virtue of switching to an electronic data collection method, many common quality assurance issues can be mitigated. Furthermore, AR has an additional advantage over tablet-based solutions in its hands-free operation, which can allow for increased safety and flexibility. Furthermore, data collected in an AR system is georeferenced locally to the bridge and shown to be adequately accurate to allow navigation to previously documented defects during the inspection. Finally, and most importantly, AR serves as a central platform within which many up-and-coming digital tools (such as artificial intelligence) can be integrated. To better understand this potential, the work also considered the use of AI for a specific use case (concrete crack measurements).

A prototype application was developed using the Microsoft HoloLens 2 headset, but the principles employed could be translated to other similar mixed reality platforms. The prototype targeted efficient integration into the entire field inspection workflow. Three key functionalities were identified and addressed as main challenges to this holistic inspection approach: 1) a structured data entry interface, 2) georeferenced annotations for visualizing historical data, and 3) computer vision for automated defect labeling. A unique research focus was how to address these challenges in imperfect, real-world field scenarios involving both novice and professional bridge inspectors. To this end, the research team explored a range of AR interaction techniques of various degrees of automation, drawing on human-computer interaction principles. The study's evaluation metrics focused on measurement accuracy, time on task, as well as the tool's impact on perceived workload and usability.

Overall, the research found that performing virtual measurements within the AR environment resulted in similar usability as traditional measurements (i.e. tape measure and crack ruler), while significantly reducing the cognitive workload. Results were mixed in terms of time on task and accuracy. In general, more automated measurement processes significantly reduced time on task, but tended to decrease accuracy. One key finding was that the best combination of metrics was found using hybrid automation approaches, where the inspector was able to refine automated defect labels. In terms of added value, less cognitive load could mean less errors in documentation process, saving time at the office. This would be added to the time saving in report generation due to the virtue of digital data collection. The platform also provides the ability to collect much more granular data, which could help improve overall data quality.

The research team was encouraged by the reactions of a small set of VDOT bridge inspectors during the field validation of the final prototype. These inspectors were impressed by the technology's capabilities and intuitive workflow, voicing support for the future potential for bridge inspection. While the performed work supports the functionality of the platform over the entire field inspection workflow, the current prototype still requires additional development to be fully implementation ready.

Supplemental files can be found at https://library.vdot.virginia.gov/vtrc/supplements

INTRODUCTION

As bridge infrastructure continues to age and deteriorate, inspection practices have evolved to better assess their condition. This includes the adoption of element-level condition ratings and the digitization of data through the National Bridge Inventory (Federal Aid Highway Act, 1968). This shift towards digital data management enhances the quality of the data collected; however, it has introduced a gap between the traditional physical pen-and-paper field workflows and the new digital data requirements in the office. This gap often results in transcription errors, as highlighted in (Smith et al., 2022). Additionally, manual measurements in bridge inspections have been shown to be inherently subjective, often with over 100% variation in area measurements (Washer et al., 2020). Thus, there is growing interest in also digitizing the *field* workflow, both to improve consistency in measurements and to facilitate the conversion to office reporting requirements.

Various DOTs across the country have been working to incorporate modern technologies into their inspection workflows. We can divide current research efforts into five rough categories: Decision Support Systems, Augmented Reality (AR), Data Mining, Data Entry Systems, and Computer Vision.

Decision Support Systems are defined in this context to contain research which yields a tool that the inspector could query in the field or office to receive suggestions or additional information from references such as the American Association of State Highway and Transportation Officials (AASHTO) Manual for Bridge Element Inspection. Indiana and Illinois have both reported artificial intelligence systems for inspectors to reference inspection regulations and examples more easily, with Illinois being noteworthy due to its integration of Chat GPT3.5 and ability to produce example inspection photos such as "a CS2 crack on reinforced concrete" (Topcu, 2023, Xu et al., 2019).

With regards to Augmented Reality based bridge inspection, New Mexico and Rhode Island demonstrated AR's potential as a measurement tool (Falter, 2020, Moreu et al., 2019). Separately, Tennessee demonstrated AR as a data visualization tool by overlaying ground penetrating radar (GPR) results onto the scanned concrete surface to allow for a sort of X-Ray vision highlighting delaminations. This approach requires post processing of the GPR data and additional manual alignment (Hu et al, 2020). However, none of these systems have been aimed at assisting with the entire bridge inspection workflow, particularly regarding documentation and reporting. One particularly essential, yet overlooked, aspect of holistic bridge inspection via AR is world localization, the process of tracking and anchoring virtual objects in the correct physical location. If this localization is too coarse or often loses its knowledge of the current state and position of the world, virtual objects have a propensity to shift and drift over time from their original placement. This problem is further exacerbated as the processing power and sensor suite available to the headset to track itself is much less than the current state of the art, due to power and size limitations.

Data mining research generally seeks to accomplish "Infrastructure Condition State Prediction". This is accomplished by feeding many inspection reports into a model and having it

predict future condition states based on previous reports. This could be utilized for more effective allocation of resources based on anticipated deterioration. Iowa, Utah, Illinois, and Ohio DOTs have undertaken research into this area (Jung et al., 2022, Abukhalil et al., 2022, Mohammadi et al., 2023, Goa and Elzarka, 2021). While potentially promising, data mining is sensitive to the quality of collected data, thus documentation consistency is critical to such efforts.

Data entry methods include two main sets of software: Inspect X and Headlight. Both systems are tablet-based interfaces designed to replace the pen/paper data entry workflow of bridge inspection, digitizing the work immediately and potentially reducing transcription errors by connecting directly to the AASHTO Bridge Management system. Headlight was created as a collaboration between Washington DOT and the private company Pavia Systems, Inc. and has been pilot tested by inspectors in Washington, Minnesota, and Texas DOTs (Yamura et al., 2018). InspectX was developed by AssetIntel, a Florida based company. InspectX has been or is in the progress of being adopted by Tennessee, Utah, North Dakota, Georgia, Kansas, Louisiana, Virginia, Arkansas, Vermont, and Los Angeles DOTs (InspectX, 2023). In many cases, however, tablet interfaces are not preferred by inspectors, due to issues with glare and portability (Smith et al., 2022).

Computer vision (CV) algorithms have been identified as potential solutions to reduce measurement subjectivity and inspection times (Pathak, 2023). While numerous studies have validated the effectiveness of CV and machine learning for crack segmentation (Yang et al., 2018, Li et al., 2020, Munawar et al., 2022), the translation of these high accuracies to real-world conditions remains problematic, as demonstrated by (Bianchi and Hebdon, 2022). Even state-of-the-art segmentation algorithms, after training on millions of images, only manage about 63% accuracy on average (Wang et al., 2023). These findings indicate that, despite significant progress, the current precision of CV algorithms is insufficient for fully autonomous use in safety critical applications. Furthermore, defect segmentation algorithms are not designed to output the precise pixels corresponding to the affected area; this would require meticulously labeled training data, which is not feasible at large scale. Instead, these algorithms output the approximate boundary of a defect. In the case of cracks, this introduces a challenge when attempting to accurately measure crack thicknesses.

Of all these technologies, Augmented Reality (AR) stands out as a possible *centralized platform* within which the other technologies could be integrated. AR's hands-free interaction, combined with its spatial sensing capabilities, allow for virtual measurements and documentation directly on the structure instead of a tablet screen (Mascarenas et al., 2021). In addition, all defect information is georeferenced, meaning that its location on the bridge is automatically captured (Jakl et al., 2018). AR also can potentially offer access to historical data and predictive analytics (Data Mining). Finally, AR facilitates the deployment of computer vision (CV) algorithms directly in the field, enabling real-time collaboration between human inspectors and automated systems (Al-Sabbag et al., 2022). This collaboration is the key to leveraging the potential benefits of CV while ensuring dependable performance.

For more information on these technologies, the reader is referred to the extended literature review included as supplementary material.

PURPOSE AND SCOPE

This study's purpose was to design an Augmented Reality-based bridge inspection tool that facilitates the digitization of field documentation, while also improving its quality and consistency using assistive tools, like virtual measurements and Artificial Intelligence. A prototype application was developed using the Microsoft HoloLens 2 headset, a platform which projects virtual holograms onto the true physical environment. Nevertheless, many of the principles employed could be translated to other similar mixed reality platforms. The research team targeted efficient integration into the entire field inspection workflow as the primary goal, that is, the inspector should be able to use AR from start to finish in the field, while intuitively following a traditional inspection workflow. Within this context, this study sought to identify and address crucial research gaps regarding the deployment of AR tool in imperfect, real-world field scenarios involving both novice and professional bridge inspectors. The prototype development focuses on three key areas: 1) an inspector-centered AR interface design, 2) georeferenced annotations for visualizing historical data, and 3) computer vision for automated defect labeling. The study's evaluation metrics focused on measurement accuracy, time on task, as well as the tool's impact on perceived workload and usability.

The scope of the AR interface design holistically considered menu structure, structured data entry, user interaction, and the information architecture best suited to the inspectors' needs on site. The specific research objectives were to:

1.1. Create an intuitive interface that mimics the traditional inspection workflow as closely as possible,

1.2. Structure the data entry to match AASHTO component and element level reporting requirements,

1.3. Facilitate the incorporation of automation tools (e.g. computer vision) while easily allowing manual fallback options in case of malfunction.

The first phase of the interface design leveraged interviews of various Department of Transportation personnel across the country and used them to determine common issues with existing inspection procedures. The second phase of design was based on user experience experiments with multiple undergraduate students. The final phase involved a field test at a concrete T-beam bridge in Roanoke, Virginia with a team of three VDOT bridge inspectors. The final prototype version was created taking the feedback from the inspectors in the field test. The team limited the scope of the project to the field inspection process, so exporting report data was not included as a primary objective. However, the interface back-end data structure was intentionally designed to facilitate data export development in future phases of this project.

The scope of the research into georeferencing was aimed at understanding how well annotations persist in space over an inspection procedure as well as over multiple inspections, particularly in challenging outdoor environments. As will be discussed, the accuracy of the georeferencing is critical for informing how current and historical data should be visualized. The specific research objectives were to:

2.1. Determine how precisely live annotations can be projected onto the bridge structure. For example, do annotated crack outlines align with the physical crack?

2.2. Determine how well annotations persist over various bridge visits. That is, are defect annotations in the same place during a subsequent bridge visit?

The research in this case was limited to the Microsoft HoloLens 2 platform, which was tested on a local bridge at various points in the year. The results are strongly dependent on the headset's sensing capabilities and the integrated software that handles spatial mapping. While the team did investigate the use of special-purpose libraries such as Microsoft's World Locking Tools, the project did not attempt to create custom algorithms in this respect. Instead, future iterations of this technology could continue to improve in this regard.

Finally, the scope of the defect automation labeling focused on the development of a computer vision (CV) system that would enable manual intervention from an inspector. In this work, we define CV as the comprehensive processing of images into actionable measurements, encompassing both machine learning (ML) and image processing as critical steps in this process. The specific research objective in this area were to:

3.1. Develop a field-capable CV algorithm for obtaining crack measurements in real-time,

3.2. Determine what types of combined AR and CV interactions create the best user experience,

3.3. Quantify how these interactions impact inspection performance in a realistic inspection setting, compared to the traditional tape measure approach.

Because the broader goal of this work was to demonstrate the potential of AR to integrate across the entire inspection workflow, some key scope reductions were made. First, the team did not focus on creating the best-performing CV algorithm possible, but rather something capable enough to enable the exploration of human-in-the-loop interaction. In other words, developing a functional manual fallback for automation malfunction was deemed more critical than maximizing the performance of the automation itself. It is also a much less studied area in the literature. Second, the team chose to focus on concrete cracking. This enabled in-depth development to demonstrate the full extent of a narrow workflow, rather than partial development of a broad workflow. Concrete cracking generally has the most stringent documentation requirements and so was deemed the most interesting candidate for research.

METHODS

Overview

The tasks performed as part of this project are as follows:

1. Augmented reality interface design
2. World localization and world persistence

3. Computer vision technical development
    a. Field architecture development
    b. Projection of computer vision results
    c. Crack detection and documentation development
4. User interaction experiment
5. Qualitative field validation

Augmented Reality Interface Design

This task specifically addressed objectives 1.1, 1.2, and 1.3. In the preliminary stages, extensive research data was gathered to comprehend the operational domain thoroughly. Engagements with bridge division professionals across various hierarchical levels were conducted to delineate the user interactions with the proposed AR system. These interactions were diversified across personal interviews, surveys, and direct observation during bridge inspections.

Data gathering encompassed 20 virtual discussions with Department of Transportation employees nationwide and the distribution of two distinct online surveys targeted at VDOT bridge inspectors and their supervising engineers. While the survey for engineers received only six responses, the inspectors' survey yielded 41 responses, highlighting diverse perspectives from relatively junior staff members. Additionally, the authors participated in inspection “shadowing” visits, observing the inspection processes of three bridges and a culvert firsthand. The insights from these engagements were instrumental in forming a cohesive model of the operational workflow, illustrating the interactions between different roles and their tasks within the system. Following this, a hierarchical task analysis was conducted, refining the collected data into an organized structure of task flows, which outlined the sequential steps inherent in the bridge inspection routine.

World Localization and World Persistence

This task specifically addressed objectives 2.1 and 2.2. To provide a consistent, repeatable, and savable 3D experience for bridge inspectors between different bridges, our AR application employs a world localization system and a world persistence system. These systems work together to provide the ability for the system to recognize a particular bridge location it has been before and then load the previously placed virtual objects (like virtual defect markers).

Figure 1. WLT with anchors visualized (green and blue dots on ground).

Microsoft, the manufacturer of the HoloLens 2 device, reports a ±10% error when using the standard HoloLens 2 tracking system, meaning that a user could walk 10 meters and the system could think they walked up to 11 meters or as little as 9 meters (fast-slow-still). Undoubtedly, this ±10% error is not acceptable to accurately track defects and other bridge information, especially since the virtual objects are tightly associated with a specific location, so if an object is not accurately located in the world, location as an information encoding modality is lost. We attempt to mitigate this "large" potential tracking loss by utilizing Microsoft's World Locking Tools (WLT). The WLT software system (Microsoft, 2024), as shown in Figure 1, utilizes a mesh of automatically placed spatial anchors, which are virtual objects constantly tracking and anchoring themselves onto a specific physical world feature (like a corner of a concrete sidewalk) to stabilize the entire application. World Locking Tools, in theory, allows the application to significantly correct for the ±10% error from the standard HoloLens 2 tracking system by sampling the closest spatial anchors to the user and seeing how the current virtual position of the user differs from the correct position (based on the spatial anchors).

In addition to the WLT tracking system, we also tested the World Locking Tool's world persistence system. This persistence system aims to recognize unique bridges the headset has traveled to and assist with loading the tracking system setup for that specific location so that the previously saved virtual objects can be accurately placed.

Computer Vision Technical Development

This task addressed objective 3.1, which was highly technical, focusing on automating the concrete crack documentation process in the field. It was determined that Computer Vision

(CV) and Machine Learning (ML) models were the most suitable for this task. However, three primary issues were identified with this approach, guiding three parallel subtasks.

The first was the inability to run these models directly on the HoloLens 2 device without significant modification and consequent performance degradation. In response to this issue, a server-client pipeline was implemented, inspired by the approach described by Al-Sabbag et al. (2022).

The second issue identified was the inaccuracy of established techniques in projecting crack centerlines, which hindered effective collaboration between human and machine-generated results. To overcome this, a novel, application-specific projection algorithm was developed.

The third issue arose from the models producing an image segmentation mask, where black pixels represented the background and white pixels indicated cracked regions. This output format was found to be incompatible with the data traditionally recorded by inspectors, which included the length of the bridge element affected by the crack, the maximum crack thickness and a condition state rating. To resolve this, the model outputs were post-processed with a custom image processing routine, converting the results into data that could be directly utilized by inspectors.

Each of these issues was addressed in parallel as described in the subtasks below.

Field Architecture

To lay the foundation for this research, we first designed a system which supports CV-based crack documentation within the AR space while also in an internet-denied environment. The Microsoft HoloLens 2 was chosen as the AR platform for this research. The bridge inspection application was written in Unity C# using Visual Studio 2022 and Microsoft's Mixed Reality Toolkit (MRTK) 2. The Computer Vision code was written in Python 3.6 and utilized the crack detection machine learning model of Bianchi and Hebdon (2022). Additional post processing of the machine learning result was written in Python 3.6.

As these machine learning models cannot be run natively on the HoloLens 2, a server was utilized for computer vision tasks. This led to the system architecture seen in Figure 2. The server was defined to be a mid-range laptop, with the only hardware requirement being the Nvidia graphics card (GPU). A server-client communication pipeline was developed using HTTP requests between python and Unity, following the examples presented in Al-Sabbag et al., (2022). This allowed the HoloLens 2 device to send images to the server and receive processed crack detection results back. As the overall goal of the project was to function in GPS- and internet-denied environments, this communication was sent through an offline router (Netgear) and was verified to function identically with a phone offline hot-spot (Pixel 6) in lieu of the router.

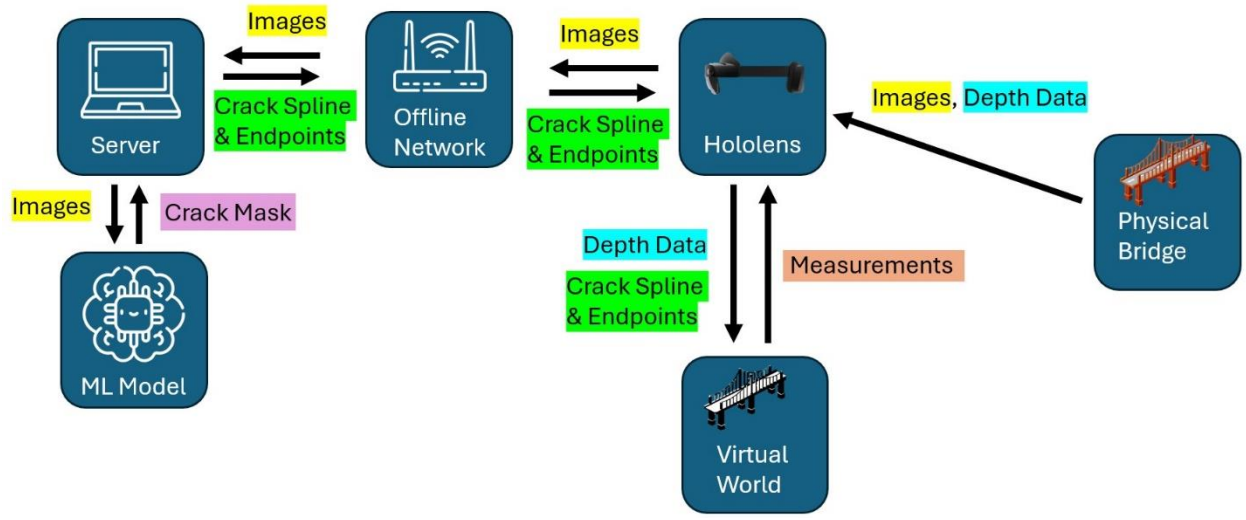

Figure 2. Field architecture utilized in this study.

Projection of CV Results

A major challenge we encountered in our research was the accuracy of projecting CV-generated annotations onto the environment through the HoloLens 2. Machine learning and computer vision operate in the 2D image space (x,y), whereas humans interact with physical objects in 3D (X,Y,Z). To enable meaningful interaction between humans and computers in a crack annotation task, the transformation between the image space (x,y) and the physical space (X,Y,Z) needs to be accurate to the millimeter scale. This is a significantly stricter requirement than for generic AR applications, where errors of a few inches might be considered acceptable. Previous research into projection in other fields (such as computer graphics) often utilized the "camera projection formula," as depicted in Figure 3, where **C** is the camera coordinates in pixels, **I** is the intrinsics matrix which corrects for lens distortion and scales the image to pixel dimensions, **E** is the extrinsic matrix which aligns world coordinates to the camera's local coordinate system, and **W** is the world coordinates of the projected point. This formula allows for the mapping of world coordinates (X,Y,Z) to pixel coordinates directly (x,y), while the inverse operation is more complex but follows similarly. However, due to issues like engineering tolerances on the camera components, lens distortions, and the eye calibration of the HoloLens 2, the standard camera projection formula results in a projection error of 2-3 in. This error led to the need for a different approach in this work.

$$\underbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\textbf{Camera Coordinates}} = \underbrace{\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\textbf{Intrinsics}} \underbrace{\begin{bmatrix} r_{00} & r_{01} & r_{02} & t_0 \\ r_{10} & r_{11} & r_{11} & t_1 \\ r_{20} & r_{21} & r_{22} & t_2 \end{bmatrix}}_{\textbf{Extrinsics}} \underbrace{\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}}_{\textbf{World Coordinates}}$$

Figure 3. Traditional camera projection formula

Other groups, such as Mohammadkhorasani et al. (2023) and Malek et al., (2023) had developed more accurate projection approaches; however, these approaches still necessitated additional, user-specific eye calibration steps. Our method, drawing inspiration from the approach mentioned in Farasin et al., (2020), utilized Holographic Image Blend shader code (Fogerty, 2016). This technique took into account the existing eye calibration within the HoloLens 2 to project an image onto a floating canvas, as shown in Figure 4. This canvas provided a direct mapping from image coordinates (x,y) to canvas coordinates (Cx, Cy, Cz).

By applying basic linear algebra, we were able to draw a line from the known focal point of the camera (Fx, Fy, Fz) through the canvas coordinate (Cx, Cy, Cz), and then raycast this vector onto the environmental mesh. Raycasting in this context is defined as finding the intersection of this vector with the environmental mesh, where this mesh is the HoloLens 2's interpretation of nearby surfaces based on its built-in depth sensor. This intersection is indicated with a physical point (X, Y, Z) corresponding to the specific pixel coordinate (x,y).

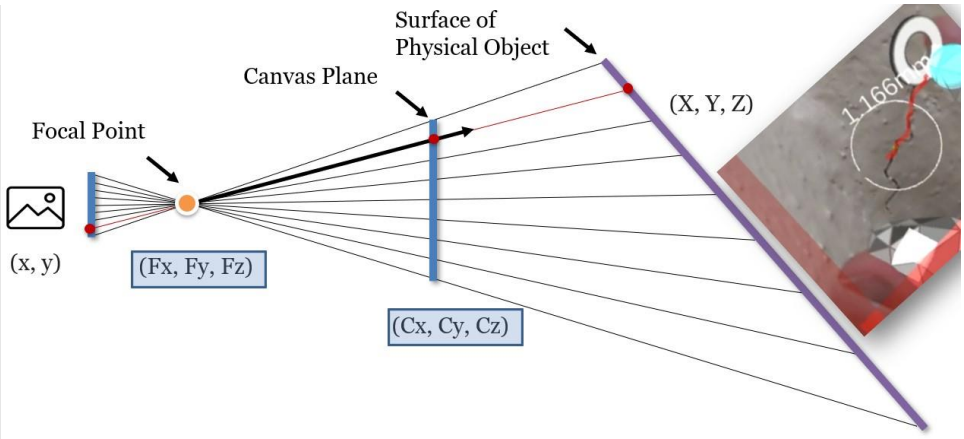

Figure 4. Example projection methodology. A temporary canvas plane is generated to leverage the HoloLens 2 capabilities for image projection. The algorithm results (e.g. crack outline) are then projected through the canvas plane onto the physical surface.

Crack Detection and Documentation

Given a mapping from image coordinates (x,y) to physical space (X,Y,Z), we were able to project the results of a machine learning algorithm onto the physical surface. However, this projection was not useful on its own since standard machine learning algorithms return an image "mask" where white pixels correspond to regions likely to contain a crack, and black pixels do not. In order to be consistent with standard inspection practices, the automated crack documentation algorithm should output three main measurements, illustrated in Figure 5: 1) The height of the cracked region, measured vertically in the direction of gravity; 2) The length of the cracked region, measured horizontally in alignment with the length of an element; 3) The width of the largest crack within this region. The cracked region is defined as total area on a bridge element affected by cracking, whether it be a single crack or multiple ones. We note that

AASHTO reports are only required to specify the length and crack width measurements (not height), however, many inspectors do measure height or crack area (length times height) as an additional piece of information, hence it's inclusion in our documentation methodology.

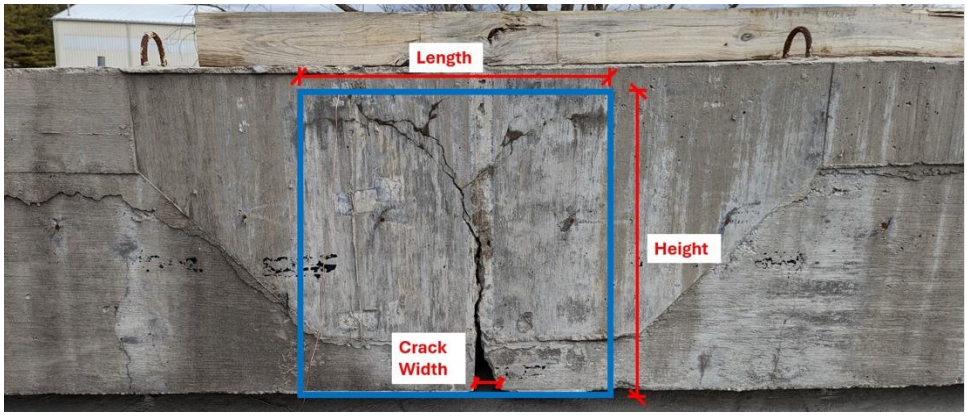

Figure 5. Crack measurements for standard inspections.

To derive these three measurements from the "image mask" returned by the machine learning algorithm, a custom image processing methodology was developed, as detailed in Figure 6. The advantage of deploying these techniques with HoloLens 2 is that these results are returned in real-time. Our crack evaluation code expanded upon previous computer vision (non-ML) focused research in crack detection and evaluation (Nguyen et al., 2014). The innovation of this research lay in the integration of machine learning results with traditional edge detection techniques. Traditional edge detection tends to produce a very noisy image. Although this noise could be calibrated for different distances and textured surfaces, such as in Malek et al., (2023), it remains a major challenge of such approaches. In contrast, the image mask generated by machine learning was significantly less noisy but lacked the detail provided by edge detection. By performing a "bitwise AND" operation on the two results, we effectively filtered the noise from the edge detection results ("filtered edges") while preserving the higher level of detail, thereby enhancing the overall accuracy of the crack evaluation.

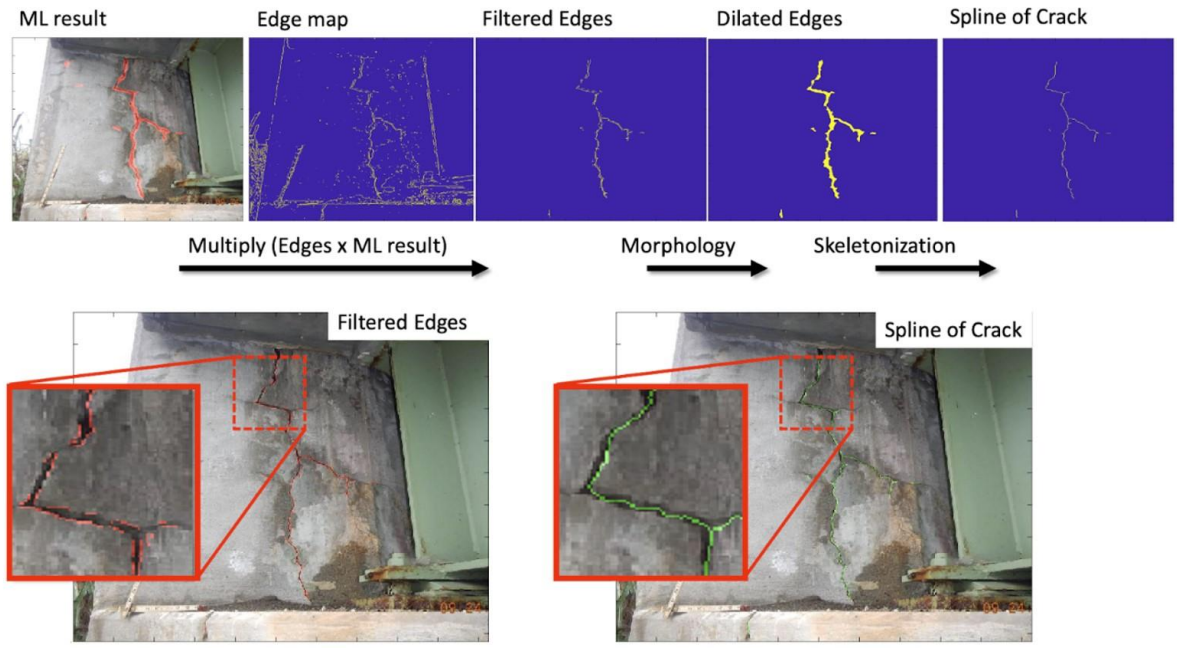

Figure 6. Post processing machine learning mask into crack spline.

From this set of filtered edges, we utilized Gaussian morphological operators such as "opening" and "closing" to eliminate any remaining noise and to close gaps in the results. Following this, we extracted the "skeleton" of the result, producing a single-pixel spline that traced the centerline of the crack. From this spline, we extracted the endpoints for use in our crack documentation workflow. For the "crack width" measurements, we selected a given pixel on the centerline and extracted a nearby region of pixels. Inspired by Nguyen et al. (2014), a third-order polynomial curve was fitted to this centerline. Then, utilizing the "filtered edges" data, we divided the same region into edges on one side of the centerline and edges on the other side. The polynomial curve, fitted to the centerline, was then offset perpendicular to the centerline in each of the two directions, using least squares optimization to determine the optimal offset that minimized the distance from the edges on that side. The two offsets were combined to yield a "crack width" measurement in pixels.

At this point, the crack endpoints were in pixel coordinates (x,y), and the crack thickness was quantified in pixels. This data, along with the pixel coordinates of every pixel on the crack centerline, was converted to JavaScript Object Notation (JSON) format and transmitted from the server to the HoloLens 2 via the previously mentioned HTTP pipeline and offline router. The HoloLens 2 then unpacked the JSON data, projecting the centerline of the crack as a "line" object with the specified coordinates and creating "crack point" objects at each endpoint.

For the crack width measurements, given a point on the crack centerline and the thickness computed by the server, we calculated the "edge points" in pixel space above and below the crack as simply the centerline point ± thickness/2. For each of these "edge points", we projected them onto the physical surface and then computed the distance between them in 3D space. This method circumvented potential errors associated with attempting to derive a "pixels to inches"

conversion factor, which would not only require a planar surface but also be susceptible to minor errors in the points used to define that planar surface.

User Interaction Experiment

A user study was conducted to investigate how varying degrees of automation impacted the usability, difficulty, and efficiency of crack documentation tasks, addressing objectives 3.2 and 3.3. Twenty participants were recruited from the student population at Virginia Tech, with an average age of 25.4 years (standard deviation = 3.5), comprising 15 males, 4 females, and 1 non-binary individual, representing novice inspectors. The majors of the users, their familiarity with inspection, and their familiarity with augmented reality (AR) were recorded and summarized in Figure 7.

Aside from this cohort, two practicing inspectors also participated, albeit with the limitation that they did not engage in the 'CV Refines Human' and 'Fully Automated' interactions due to time constraints. To avoid confusion with another group of inspectors who participated in the field validation, these inspectors will be referred to as “lab inspectors.” The lab inspector data was excluded from statistical analyses. These analyses, which involved calculating means, standard deviations, and conducting various significance tests (including ANOVA, t-tests, tests for homogeneity of variance, and evaluations of normal distribution), relied solely on data from the student population. This exclusion of inspector data from the analyses was deliberate, aiming to ensure that the statistical conclusions accurately represented the characteristics of a broader, non-expert population and avoided any potential biases that might arise from including the expert subset.

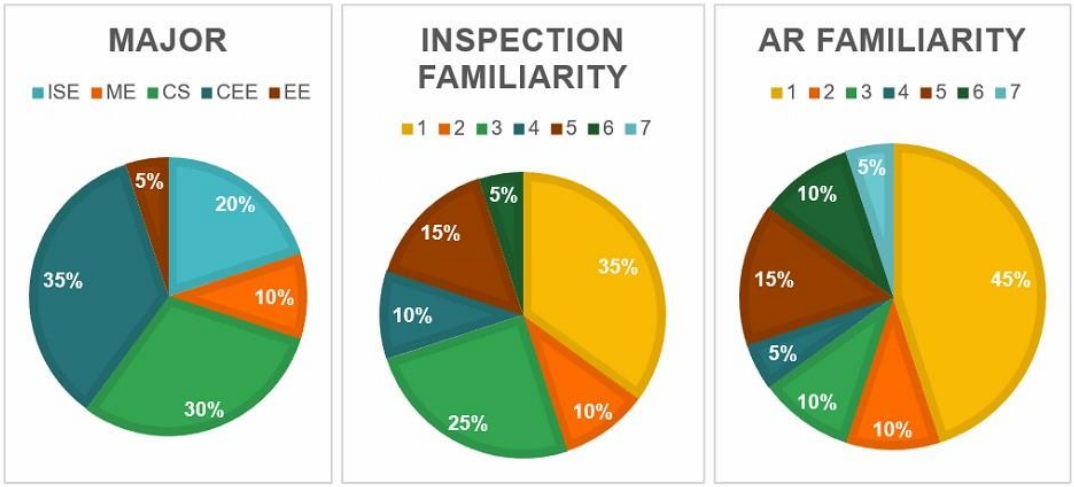

Figure 7. Student population major and inspection/AR familiarity. Majors are: Electrical Engineering (EE), Industrial and Systems Engineering (ISE), Mechanical Engineering (ME), Computer Science (CS), and Civil and Environmental Engineering (CEE). For the familiarity plots, 7 represents extremely familiar while 1 represents no experience at all with the subject.

Experiment Design

Our decision to primarily involve novice inspectors was based on several considerations. Firstly, although the final version of the inspection tool is intended for professional inspectors,

the purpose of this study was to assess the impact of specific interaction techniques for defect detection automation (detailed below). Novice users, being untrained, represent an unbiased group of subjects, as they are equally unfamiliar with both traditional and AR-based inspection techniques. As the results will demonstrate, practicing inspectors tend to have a strong preference for the Tape measurement approach, which could bias the findings. Secondly, it is much easier to recruit novice participants from the student population, which assists in achieving a larger study size.

The experiment featured two independent variables. The first was the interaction method, with five different methods being evaluated, explained in detail in the following section. The second independent variable was the cracked surface documented by the users, with a total of five different surfaces, as shown in Figure 8. These surfaces included three full-sized concrete bridge girders and two smaller beams that were tested to failure at the Virginia Tech Structures lab. Consequently, many of the cracking patterns observed were more severe than what would typically be found in an existing bridge. Additionally, one of the cracked surfaces (Figure 8, Photo 4) was reinforced with carbon fiber on the upper half of the region of interest for this study, while another (Figure 8, Photo 5) had carbon fiber present to the left of the region of interest but was not part of the region.

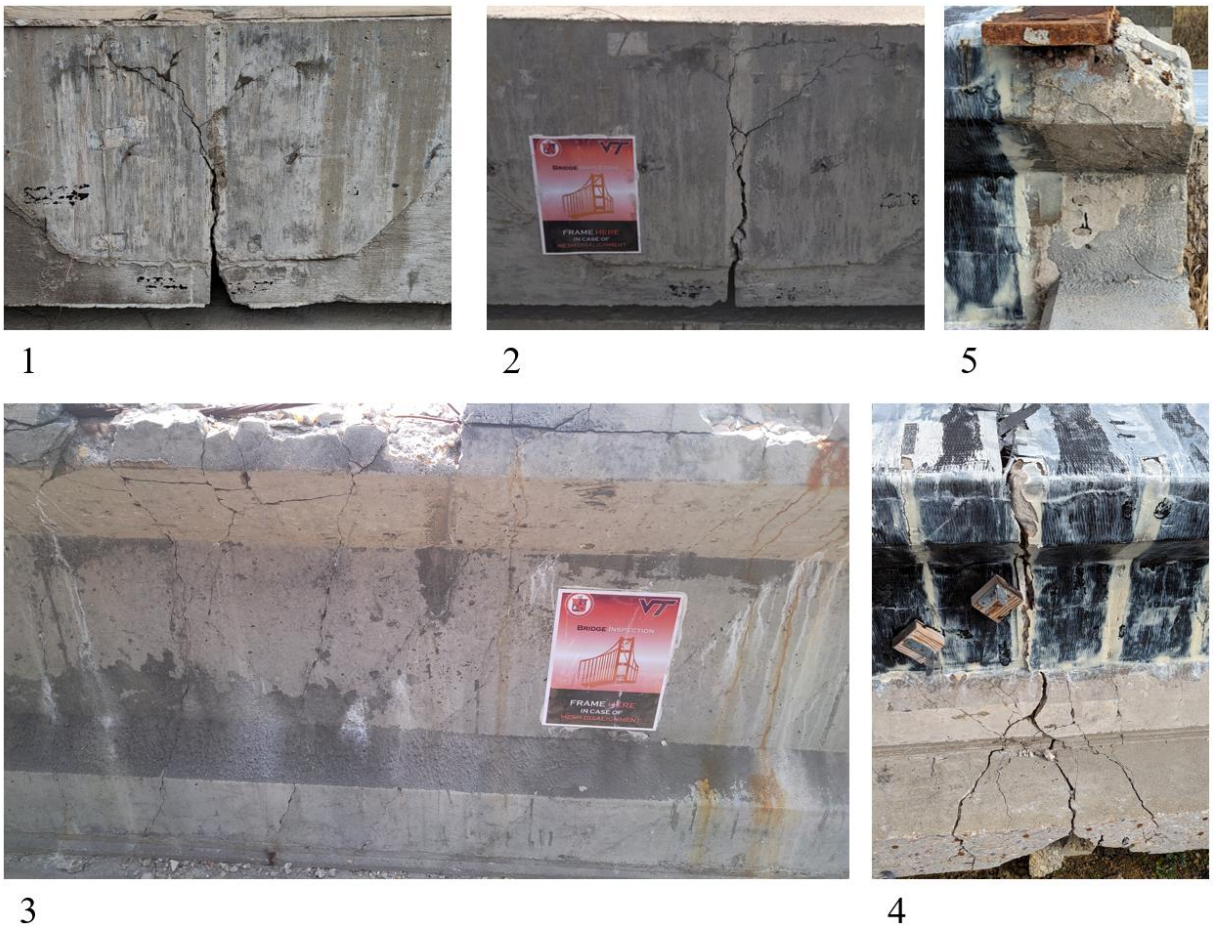

Figure 8. Cracked concrete girder regions (labeled 1–5) used in this study.

The order of the experiment tasks was randomized using a Balanced Latin Square design (Montgomery, 2020). Before beginning the experiment, each user was introduced to its contents and trained on how to measure cracks. Following this introduction, the user completed the pre-test survey, which collected demographic information and assessed their level of familiarity with the tasks. Next, the HoloLens 2 was calibrated to fit the user's eyes. Then, the user was taken outside to complete a "tutorial" on the various interaction techniques. The tutorial was considered complete when the user confirmed they felt comfortable with the techniques and demonstrated the capability to utilize all available interactions. Based on the balanced Latin Square Design, the user would start with a given interaction technique and document each of the five cracked surfaces, which were ordered randomly according to the balanced Latin square design. After completing the 5th task, the user was asked to complete the NASA-Task Load Index (TLX) and System Usability Scale (SUS) surveys before repeating the documentation process and surveys for the next interaction technique.

The NASA-TLX is a multidimensional survey that measures perceived workload through six dimensions: mental demand, physical demand, temporal demand, performance, effort, and frustration. Participants rate each category on a scale of 0 to 100 and then perform pairwise comparisons between the dimensions to weigh them according to their perceived relevance to the task at hand. SUS is a ten-item questionnaire that provides a quick measure of the usability of a system, where participants respond on a five-point Likert scale from "strongly disagree" to "strongly agree." Upon the completion of all five interaction techniques, the user was given the post-test survey. The total duration of the experiment was approximately three hours.

In addition to SUS and TLX, the team also recorded time on task and measurement error. Although participants took crack length, height, and width measurements (see Figure 5), we chose to condense this into area (length * height) and width for conciseness of the results. Identifying differences between interaction methods for the area and thickness measurements required transformation into an error metric prior to aggregation for each interaction method, due to the distinct 'ground truth' values associated with each task. Two different error metrics are analyzed in this study 1) Self Error is defined to be relative to each participant's own measurements using the Tape method, and 2) Ground Truth (GT) Error is defined to be relative to the "correct" global ground truth. The first approach-error relative to user's respective Tape measurements-controlled for the lack of inspection experience among the student participants, however this method prevented comparative analysis between the AR interactions and the Tape interaction. The second approach, error relative to a ground truth, allowed for comparison with the Tape interaction but could include a bias due to the user's inexperience. By including both analyses, we can identify and account for these biases (if present) by observing the differences between the two sets of results.

In this study, for the first approach the error metrics for each task (area and crack width) were defined using the following formula:

$$V_{i,errorSelf} = V_i - V_{Tape} \tag{1}$$

In this equation, V_i represents the measurement (area or crack width) obtained using the interaction method, where i encompasses the following methods: Manual, Human Corrects CV, CV Corrects Human, and Fully Automated. The term V_{Tape} refers to the corresponding measurement obtained using the Tape method.

For the second approach, the error metrics for each task (area and crack width) were defined using the following formula:

$$V_{i,errorGT} = V_i - V_{GT} \quad (2)$$

where the term V_{GT} refers to the Tape measurement obtained by the researchers to establish the ground truth result for each cracked region.

Interaction Modes

Building on existing literature on "adaptable automation" and "graceful degradation", the researchers split the AR interface design into four unique levels of automation (in addition to the baseline existing measuring Tape workflow) (Wickens et al., 2021, Aniculaesei et al., 2019). The goal was to be able to first test these levels of automation separately to determine the benefits and limitations of each, as compared to the Tape baseline. Based on those results, a final interface could be designed that allows the user to seamlessly swap between levels of automation as needed, while always allowing more manual fallbacks for when automation fails. The five levels of automation were designated as 1) Tape, 2) Manual AR, 3) Human Refines CV, 4) CV Refines Human, and 5) Fully Automated. Manual AR in this context refers to the set of AR interactions for crack measurement without any automation, no physical tools are used in the manual for this experiment. According to this philosophy, methods (3) and (4) are a hybridization of methods (2) and (5). For this reason, we do not describe the methods in order, but rather leave methods (3) and (4) until the end, as these build on the others.

For this study's purposes, a Tape baseline interaction was defined to best mimic the existing workflow of a bridge inspector documenting a crack defect on a concrete element. A folding ruler was used for measuring linear and area measurements. A crack comparator was used for measuring crack widths. A pen and notepad were provided for recording measurements taken.

The first AR interaction mode, shown in Figure 9, was called the "Manual AR" case and represented the least automated of the AR interaction modes. Based on the interview and survey results from the inspectors, it was determined that crack endpoints are something that inspectors are interested in tracking for crack growth. For the first Manual AR interaction technique, a user tapped the bridge element's surface with their finger and the device created a holographic red sphere at that location. As the user marked more points, a blue bounding box was automatically generated which enclosed the points and automatically calculated and recorded the associated area of the bounding box. Marking all crack endpoints naturally generated a bounding box that enclosed the entire cracked region. The second Manual AR interaction technique allowed the user to move any of the red spheres by simply "pinching and dragging" them along the physical surface of the element. This interaction was locked to the 2D plane of the surface. The third Manual interaction technique allowed the user to virtually measure crack widths by "pinching and pulling the sphere towards themselves." As the sphere was locked to the surface of the

element, this action caused the sphere to grow larger. The diameter of this sphere was used for measuring crack widths; the user could make the sphere larger or smaller until the edges lined up with the edges of the crack. The last interaction allowed the user to move the spheres towards or away from the surface by first pinching and holding the sphere for 5 seconds, at which point it would turn from red to yellow to designate an "unlocked" state in which it can be moved freely in 3D.

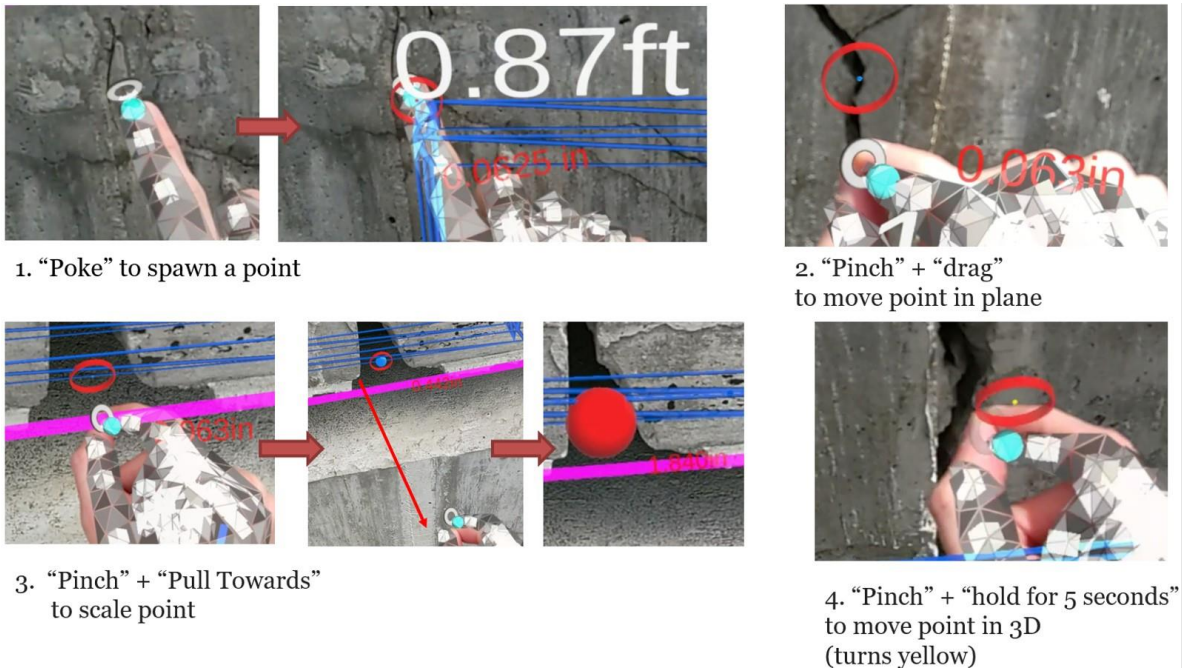

Figure 9. Manual interaction techniques: 1) Point creation 2) Point translational movement 3) Point scaling for crack width measurements 4) 3D movement point movement.

The most automated AR interaction workflow was named "Fully Automated," shown in Figure 10. In this interaction, the user began by taking a picture of the cracked region. This picture was sent to the server and processed into a line tracing the centerline of the crack and crack endpoints. Red spheres were automatically placed at the calculated crack endpoints and the blue bounding box was generated and used to calculate the affected area. Crack width values were automatically calculated by the CV and the largest crack width was recorded for the calculation of condition state. If the inspector deems the result unsatisfactory based on his experience, they could retake the picture from a different distance/angle to attempt to get a better result. In a field scenario, the user could manually override these results, but to evaluate the performance of specific levels of automation, this was not permitted in this mode. As the ML model results are variable based on distance and angle of the photo, the photo retake option allows the user to attempt to get a better result. Generally, taking a picture closer would recognize smaller cracks, while taking a picture further away recognized larger cracks. We identified a need for scale invariance in these models as a point of future work.

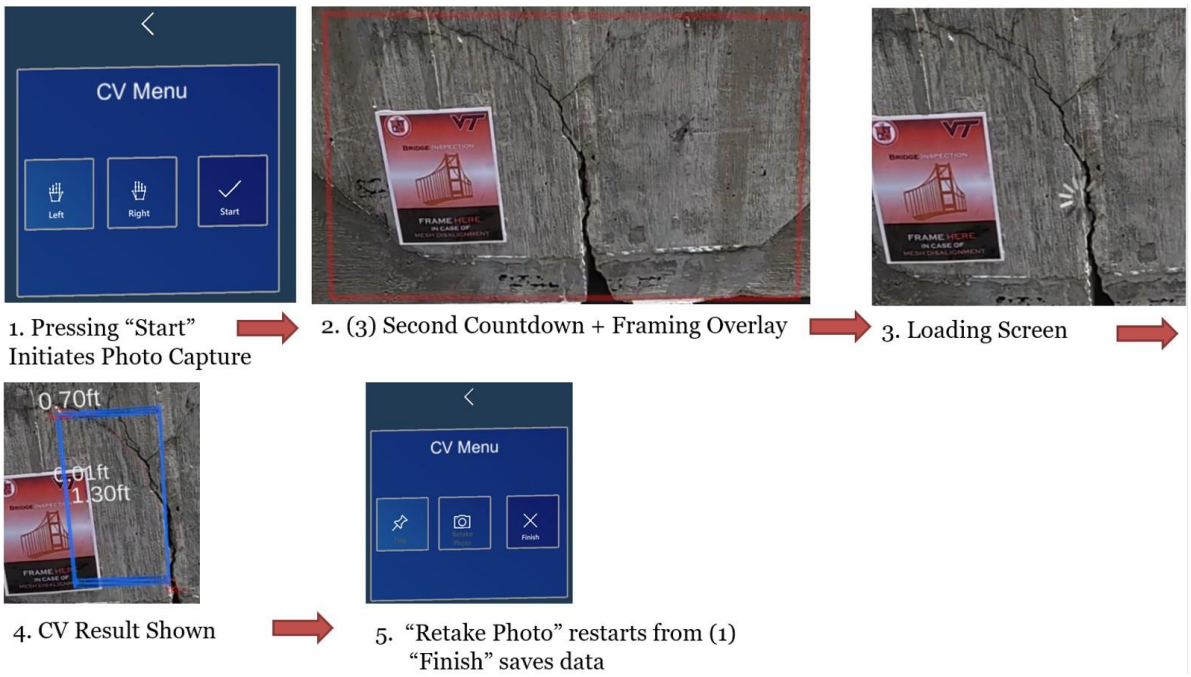

Figure 10. Fully Automated interaction techniques 1) Photo capture initiation 2) Photo framing and countdown 3) Loading screen 4) CV results projected 5) Photo recapture and finish options. Note that the result shown in (4) does not capture the full extents of the crack, representing a case where the photo should be retaken.

The first of the two hybrid modes was called "Human Refines CV" and was designed to be a less automated interaction in which the human gets the final say in the result. In this interaction, it began like the fully automated one in that the user took a photo and the CV projected the resulting crack endpoints and area onto the surface. For this interaction, the user could not retake the photo but instead could fully modify the results of the CV as if those points were created by the user in the Manual interaction mode.

The second of the two hybrid modes was called "CV Refines Human" and was designed to be a more automated interaction where the human gave the CV a starting point which the CV could then modify. This type of interaction was supported by literature such as Fitts' list which states that computers are better than humans at fine details while humans excel at big picture thinking (Fitts, 1951). This functionality was accomplished by implementation of a "snapping" functionality akin to Autodesk's O-Snap functionality. In this interaction mode, the user still began by taking a picture, but the CV did not create any points, it only projected a line designating where it determined the crack centerline to be. The user then created points, similar to the manual interaction mode, however, any points created or moved close to the crack centerline would automatically snap to the centerline and update the sphere size to the crack width at that location as determined by the CV. The user was not responsible for crack width measurements in this mode.

Qualitative Field Validation

A field validation of the AR bridge inspection tool was completed at a concrete T-beam bridge in Roanoke, Virginia with a team of three inspectors. These were different from those engaged in the user interaction experiment and are referred to as "field inspectors." The final version of the interface was used as described in the results and discussion, section "Augmented Reality Interface Design". The interface utilized for field validation featured a hierarchical element selection process in the AASHTO format, along with manual area and condition state documentation capabilities for all defects. For crack defects, automatic length and height measurement capabilities were provided. Crack width values were measured using a physical crack comparator and input through a virtual keyboard. A recommended condition state was offered for defects like cracking, which had a quantitative metric (e.g., crack width) corresponding to a specific condition state. Condition state definitions were provided for all defects, and example condition state photos were made available for cracking, corrosion, and spall defects.

Each inspector was given a short tutorial on how to operate the interface. Afterwards, each inspector documented a single crack defect on a reinforced concrete bridge girder. While using the AR interface, the inspector was continuously encouraged to give comments on what they liked and disliked about the interface. Once all three inspectors had documented a single crack defect, they were timed while documenting a different crack defect. A freeform collective discussion was conducted after the conclusion of the field test, which included suggestions from the inspectors on features they would like to see incorporated into a future version of the interface as well as their general thoughts of the AR inspection tool itself.

RESULTS AND DISCUSSION

Augmented Reality Interface Design

The first and most important result of the data collection and bridge inspection shadowing phase was understanding the bridge inspection workflow. Typically, an inspection commences with a review of the prior report, which is then physically carried to the inspection site. This document, averaging 20 pages, predominantly features narrative descriptions of previous findings concerning defects or areas of concern. Inspection teams, generally comprising two members, leverage these reports to guide their inspection, with one member annotating and assessing physical conditions on-site and relaying findings to their partner for documentation. Inspectors typically follow a hierarchical process when selecting which elements to inspect: they first focus on one component (e.g. deck, superstructure) then inspect all elements of the same kind (e.g. barrier rail, concrete girders) before moving on to the next element. Post-inspection, these annotations and photographic evidence are incorporated into a digital report, which subsequently requires manual data entry into various software systems, including structural analysis and bridge management tools utilized by different departments.

The challenges identified with this process were primarily attributed to the reliance on traditional, narrative-based note-taking methods. This practice, lacking a unified system for data

organization, necessitated redundant manual data entries into disparate systems, often leading to inaccuracies and issues with data integrity. Additionally, the subjective nature of manual measurements and the difficulty in locating defects based on narrative descriptions were identified as significant obstacles. Furthermore, insights from inspector surveys indicated a cautious stance towards adopting new technological solutions, rooted in previous unsuccessful attempts to integrate modern devices into their workflow.

The final bridge inspection augmented reality interface took all these lessons into account, from the initial stakeholder surveys to user experience experiments. It also considered results from the user experiment, which will be discussed later. The interface is intended to completely digitize the inspection process while maintaining the familiar workflow of traditional pen-and-paper inspections. Another key feature of the interface is that it allows inspectors to seamlessly switch between manual, digital, and automated measurements. This not only allows inspectors to work around potential issues with automation, but also allows them to work with the digital tools to the extent that they are comfortable.

Figure 11 shows an overview of the AR interface workflow until arriving at the data panel, the main menu from which measurements, notes, and photos are recorded. Figure 12 shows the flow of actions available withing the data panel.

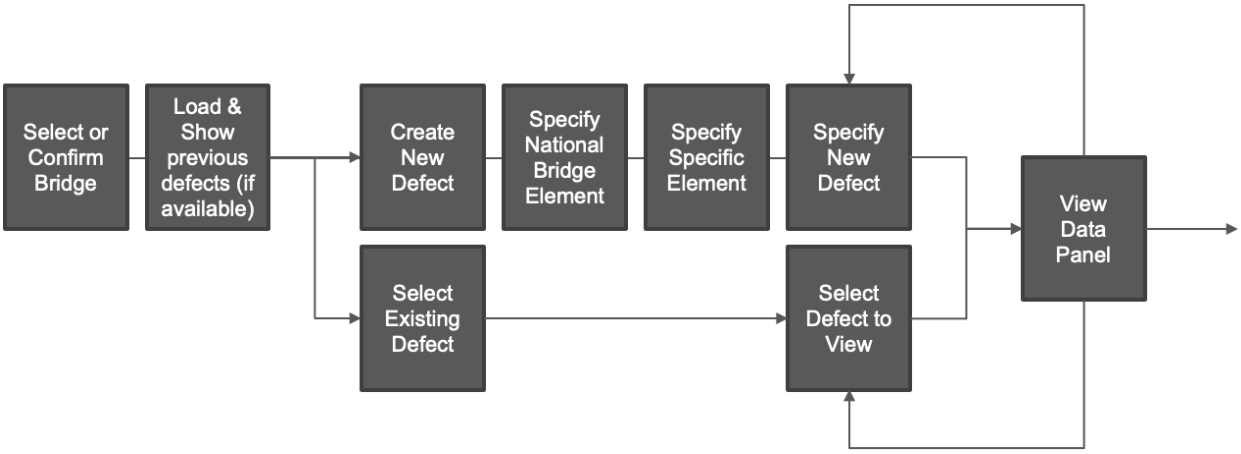

Figure 11. Flow chart of the AR interface workflow to arrive at the data panel.

The following subsections show each stage (i.e., menu) of the interface to explain each of the diagram elements in detail. We note that the actual inspector's view through the headset is slightly different than the images shown below due to limitations of the AR display's video/image capturing features. For example, some virtual elements are visually offset from the real-world in the image, but not when viewed through the AR headset. Further, to expedite the reporting process, the images shown below were captured at a site with a series of concrete blocks. While not a bridge, the two environments are identical from the perspective of the interface (unknown outdoor location, without internet or GPS). Lastly, some of the images show a "virtual glove" over the inspector's hands to visualize the HoloLens 2 hand tracking feature. Doing so is optional, but can help the user adjust their gestures by understanding where the HoloLens 2 perceives the hand to be.

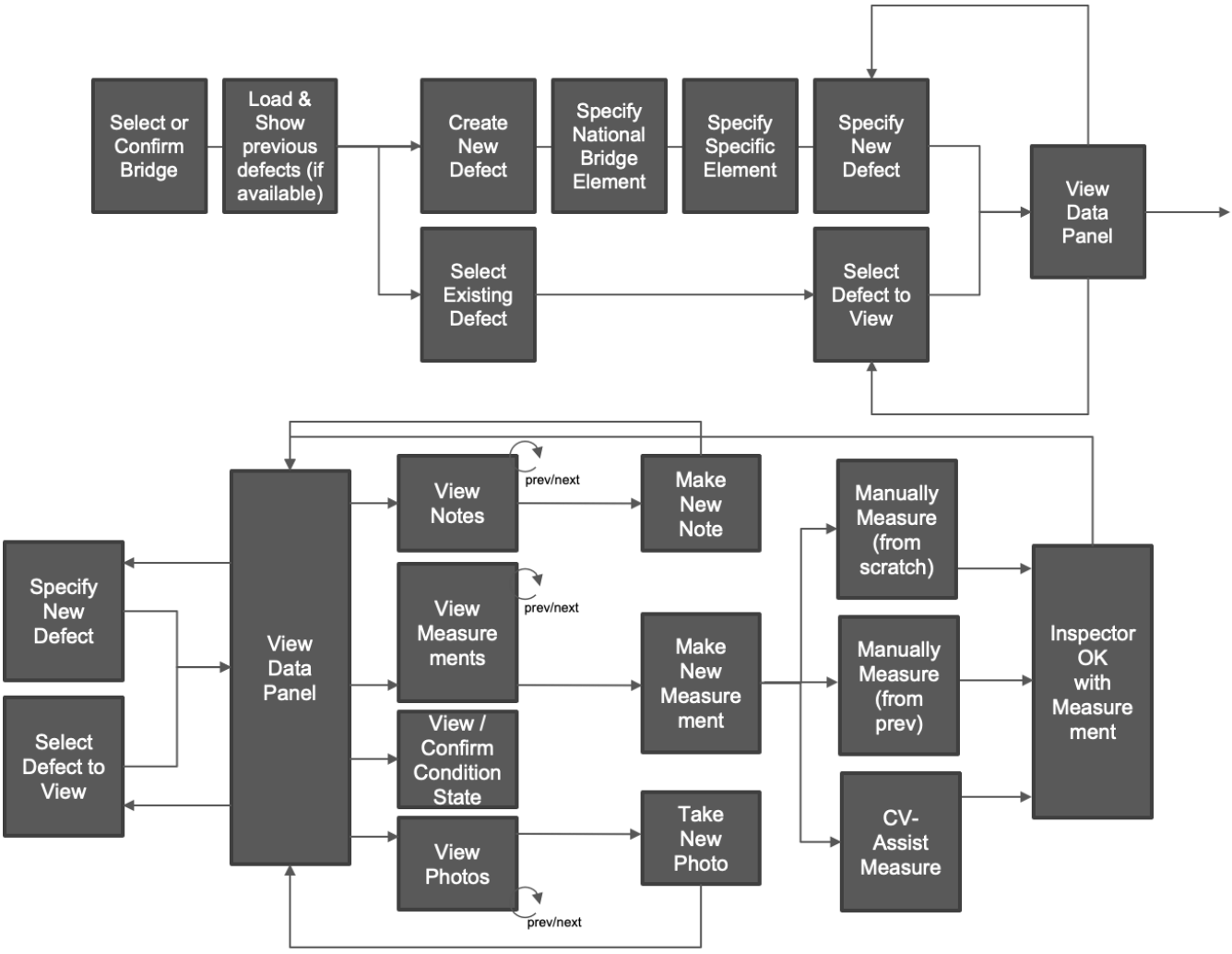

Figure 12. Flow chart of the actions available within the data panel.

Bridge Selection Menu

The first step is to select the bridge for which the inspection will be performed, as shown by the bridge selection menu in Figure 13. This could be a new bridge or a previously inspected bridge. Currently, when accessing a previous bridge, the interface searches for and loads bridge data based on a unique bridge identifier (specified when inspectors create a "new bridge" for the first time in the system). We anticipate the future versions will detect the inspector's location and present a list of one (or more) bridge(s) in that area, whereby inspectors would simply confirm the system's bridge selection upon app startup.

The previous bridge environment (the 3D geometry used for localization) is saved within the HoloLens 2 itself and can be transferred to another HoloLens 2 if needed. The documentation data (images, measurements, and notes) are stored in the laptop server.

Figure 13. AR user interface showing the first screen upon first starting the app.

Defect Creation/Edit Menu

After inspectors load a bridge (new or existing), the system allows access to a menu whereby inspectors can "Create New Defect." The interface shown in Figure 14 contains placeholders (labeled <Empty>) where additional defects can be listed, either from the previous inspection or the current one. In our running task example, the inspector chooses "Create New Defect".

When inspectors select "Create New Defect" instructive text appears to guide inspectors in the workflow, as shown in Figure 15. We anticipate that this text will be refined, and or optional as we learn more about how inspectors wish to work. Continuing this task, an inspector would physically touch the bridge surface to denote the location of the new bridge defect. This interaction provides haptic feedback and is easier to use than a "point and pinch" gesture. The research team chose this under the assumption that all inspections are conducted at arm's length, as is the current AASHTO requirement. Options for annotating at distances of several meters are possible and should be considered as alternatives where arm's reach is not possible. However, it is important to note that implementing such a strategy would also require revisions to the inspection guidelines.

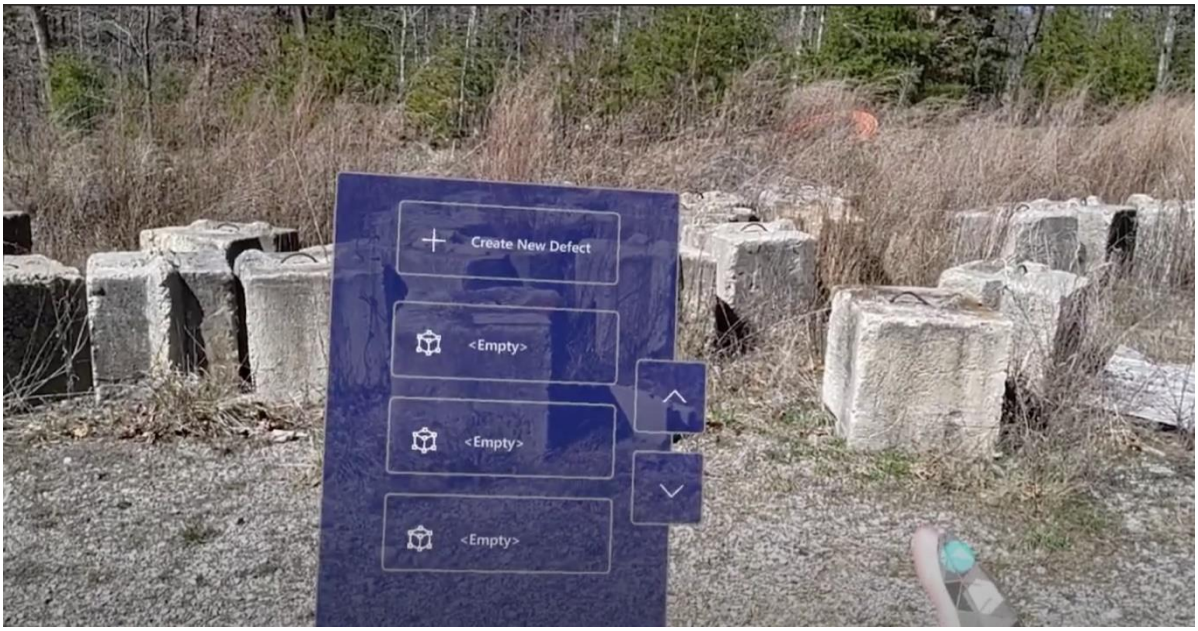

Figure 14. Defect selection menu.

Figure 15. Instructions on how to record the defect location.

Defect Specification Menu

After inspectors touch the surface of the bridge (at a defect location), a blue orb (i.e., sphere) appears, as shown in Figure 16. In subsequent bridge inspections, the orb signifies that a defect has been documented, and inspectors looking out across a bridge section will see 3D virtual orbs at all physical locations where defects had been previously noted (top right of Figure 16). In this example, we are creating a new defect, whereby the system prompts inspectors for structural meta data regarding the defect location. The first menu (Figure 16), prompts the user to

select from a list of general elements (sometimes called components) such as Decks and Slabs, Railings, Superstructure, etc. Inspectors' selection at this level leads to another menu (not shown). In this example, the inspector chooses "Decks and Slabs" as the general element (top left corner of the menu in Figure 16).

Figure 16. General bridge element selection menu.

This opens another menu of specific element options (Figure 17), which fall under the general element category (e.g. reinforced concrete deck, prestressed concrete deck, etc.). Note that these elements are coded based on the AASHTO bridge inspection manual. In this example, the inspector chooses "12: Reinforced concrete deck." Subtle, but present, is a "back arrow" located above the menu in the center (i.e., a chevron, or < symbol). This feature allows inspectors to return to the previous menu in cases where the incorrect bridge element was selected in the previous menu (Figure 17).

After selecting the specific element from the previous menu, the system presents inspectors with a list of possible defects that can be associated with reinforced concrete decks according to AASHTO (Figure 18). As before, we present the AASHTO numerical codes to assist inspectors in noting defects consistently and documenting defects in compliance. Not visible in the image above is the same "back button" described earlier. Here the inspector is noting that the defect is related to "1130: Cracking (RC and Other)".

The hierarchical selection process ensures that minimum effort is expended recalling the correct element/defect codes or double checking that a defect code is allowed under a certain element. Overall, defect specification process takes only a few seconds to complete.

Figure 17. Specific bridge element selection menu.

Figure 18. Defect selection menu.

Data Panel

After specifying the defect type (in our example we are using Cracking (RC and Other), a detailed data panel appears with several options available (Figure 19). Some options are general to all defects while others are specific to the defect type.

Figure 19. Defect data panel. Note that the menu title is incorrect: It should read Cracking (RC and Other), as this relates to the available data entry options.

The left side of the panel is reserved for previous inspection data. The right side of the panel is reserved for current inspection data. Inspectors work with just the right-hand side when documenting a new defect. When working with a previously noted defect, the left-hand side is available for reference. In the top center, there is a button that allows inspectors to optionally copy data from a previous inspection to the current inspection (for example, to use as a starting point for new data entry). In the center middle of the panel are buttons in which inspectors can change the mode of the overall panel, which in turn specifies what type of information (as well as permissible actions) are available in both the previous (left) and current (right) sides of the panel, allowing inspectors to view/manage: (1) measurements, (2) photos, and (3) notes.

Measurements Mode

In this mode (also Figure 19), inspectors can view historical measurements as well as document new measurements, specific to the defect. Further, inspectors can annotate the scene using virtual tools to measure a defect and/or access AI/CV tools to assist in measurements. While in measurement mode, inspectors have access to the following functions:

<u>Previous Inspection Data</u>

The last inspection date is presented at the top of left side, and controls are located at the bottom of the left side to allow inspectors to access previous reports. The number of reports and

the currently selected report (shown as 0/0 for new defects) is presented to help inspectors navigate large measurement sets. As inspectors work their way back through previous reports, the date is updated as is the currently selected report. When there is no data available, the panel clearly states this as depicted in Figure 19..

Current Inspection Data

The right side contains the current inspection data. The current inspection date is located at the top of the right side, and controls located at the bottom of the right side allows inspectors to specify/confirm a condition state, view details about condition state definitions and see visual examples, as well as start an AR annotation. In the case of cracking, inspectors can enter data describing the: (1) crack length, (2) crack height, (3) crack width, and (4) crack spacing. If measurements are performed manually (e.g. via a tape measure or crack ruler), the data can be entered directly using a virtual keyboard (Figure 20) by simply selecting the field to the right of the label. Alternatively, the inspector can access digital measurement tools via the "Start Annotation" button in the bottom right (see subsection 5 below). Importantly the data panel also allows the specification of the defects condition state (CS1-CS4). The white bounding box indicates the software's suggestion based on the specified measurements, but the inspector is expected to make the final selection. In order to assist with this process, particularly for junior inspectors, a "View condition state details" button is available to view the AASHTO manual guidelines for defect type selected (see subsection 6 for more details).

Figure 20. Virtual keyboard entry via the stock Microsoft HoloLens 2 keyboard. We are working to develop a specialized number pad (essentially the right side of the keyboard shown above), that will consume less visual space and be easier to work numerical data. Also shown in the figure, are controls above the number pad to allow inspectors to use "speech to text" functions.

Photos Mode

In this mode (Figure 21), inspectors can view historical photos as well as create new photos (including photos with AR annotation). While in photos mode, inspectors have access to the following functions:

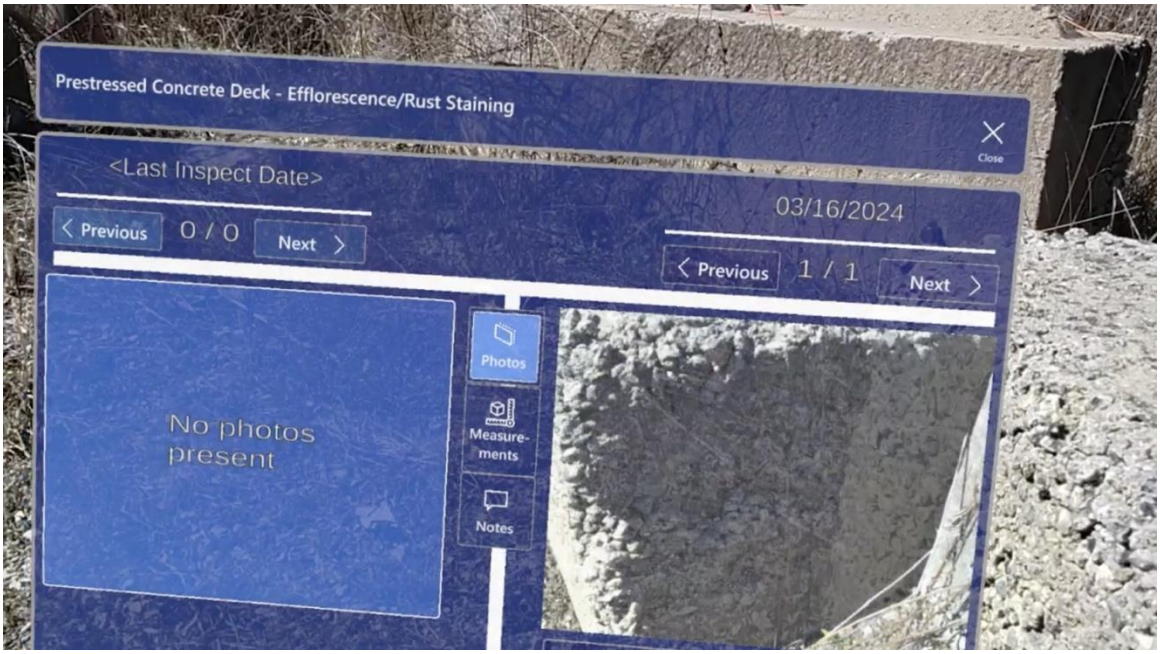

Figure 21. Example of Photos Mode where one photo has been taken during the current inspection.

Previous Inspection Data

The last inspection date is presented at the top of left side, and controls are located at the top of the left side to allow inspectors to access previous photos. The number of photos as well as the currently selected photo (shown as 0/0 for new defects) is presented to help inspectors navigate large photo sets. As inspectors work their way back through previous photos, the date is updated as is the currently selected photo. When there are no photos available, the panel clearly states this as depicted in the Figure below.

Current Inspection Data

The right side contains the current inspection photos. The current inspection date is located at the top of the right side. The number of photos as well as the currently selected photo (shown as 0/0 for new defects) is presented to help inspectors navigate large photo sets. Below the image panel (not visible in Figure 21), inspectors can also take a new photo or start a new annotation (see subsection *5. Start Annotation*, below).

Notes Mode

In this mode (Figure 22), inspectors can view historical notes as well as create new notes. At this time, notes mode is still a work in progress. However, we anticipate inspectors will have access to the following functions:

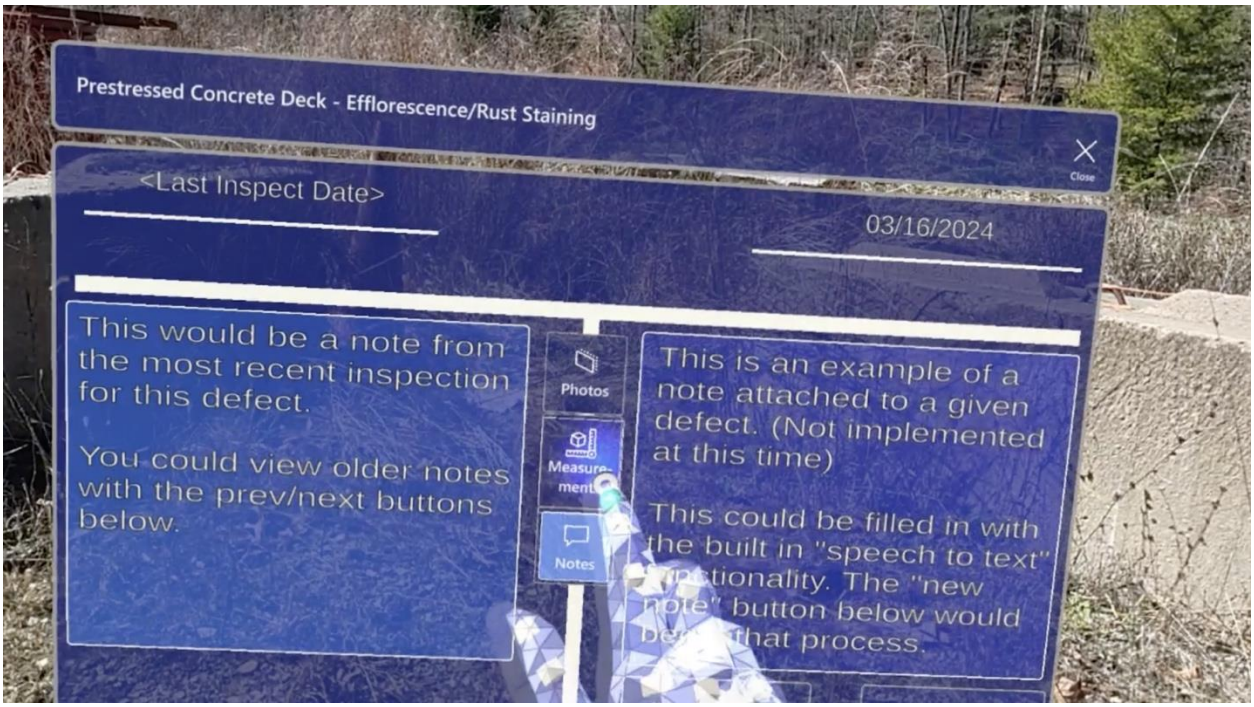

Figure 27. Virtual objects at Virginia Tech test bridge used to ensure the persistence of virtual objects in the tracking system.

Previous Inspection Data

The last inspection date is presented at the top of the left side. We anticipate adding controls located at the bottom of the left side to allow inspectors to access previous notes. The number of notes (n) as well as the currently selected note (m) will be represented as m/n. As inspectors work their way back through previous notes, the date will be updated as is the currently selected note.

Current Inspection Data

The right side contains the current inspection notes. The current inspection date is located at the top of the right side. Below the image panel (not visible in Figure 22), inspectors can also create a New Note, as well as start a new annotation (see section above on Start Annotation). Rather than typing notes virtually, it is expected that inspectors will leverage the built-in speech-to-text feature to accelerate the process.

Start Annotation

This menu allows inspectors to utilize aspects of our prototype that feature novel world-relative AR and CV features. In short, this feature allows inspectors to work directly with the bridge surface to visually annotate the defect (e.g., crack), and take virtual measurements while also affording access to CV assistance to expediate the annotation process. With training, performing the process virtually can expedite the data entry process and also collect much more granular data on a defect.

The Start Annotation process begins by first assessing if a photo exists for this inspection at this specific defect. If no photo exists, the inspector is prompted to frame a shot of the defect, and a countdown begins prior to taking a photo. The concept of operations is that a photo is needed for any CV operation and this operation can take a few seconds to process on the server side. Thus, we initiate the process with a photo such that inspectors can continue to use the tool while the photo image is being processed. If a photo already exists for this inspection, then the CV has already performed its calculations for this defect and instead of prompting the inspector take a photo, the annotation menu (Figure 23) is directly presented to inspectors.

The annotation menu is a key component of the annotation process. In short, the menu allows inspectors to manage how the annotation is performed. The options are to: (1) select nothing in this menu and immediately begin virtual annotation (the default action), whereby inspectors touch the bridge surface directly in order to collect quantitative data (see Virtual Annotation below); (2) Start a new annotation using data from the previous inspection (initiated using Start from Prev button); or (3) Start a new annotation using the data derived from the CV process (initiated using Start from CV button). In all cases, the visual result is a series of virtual points (i.e., red spheres) placed initially by the inspector (manual interaction), the previous inspection, or the CV. Once points are placed, inspectors can move them to match the current defect condition (see Taking AR Measurements below). As points are being manipulated, the annotation menu further allows inspectors to Undo and Redo point placement actions using buttons located across the top. Finally, inspectors can re-take a photo (both to document the defect as well as to provide new input to the CV components) and finalize the annotation using the Finish button.

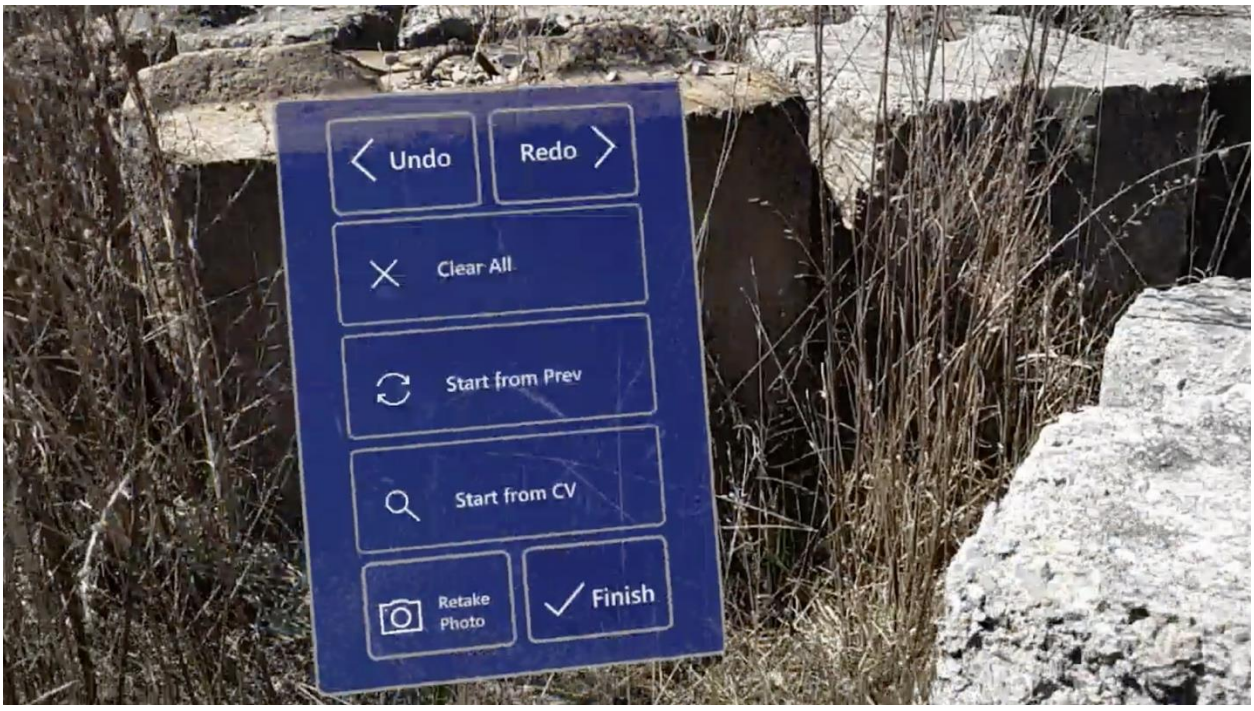

Figure 23. Annotation Menu.

Virtual Annotation

As mentioned above, once inspectors start an annotation the default app behavior allows inspectors to directly place points on a bridge surface to both denote and measure a defect's quantitative parameters of interest (e.g., for cracks, this would be length – as well as maximum crack width). These points can also be initially placed by the system using the previous annotation or CV, with the inspector being able to subsequently move the points to positions of their liking.

To place a point, inspectors simply extend their index finger and touch the bridge surface (Figure 24). A red point/sphere appears, which can then be moved using a pinch gesture (index finger and thumb). When any outer point is being moved, the dimensions of the overall bounding box are updated in real-time for inspectors to see. Once the pinch is released, the point stays in place.

It is important to note that inspectors can place as many points as they wish per defect, with the minimum number of points being two. Thus, for example, some inspectors may wish to place points to mark multiple areas of interest in a defect (e.g., end points and/or branching of cracks that diverge). These points not only assist in measurement, but also mark key features of a defect that will be visually available (i.e., visually overlaid onto the bridge) during subsequent inspections.

Once inspectors are confident in the placement of points (again, either manually, or assisted via previous and/or CV assistance), the quantitative measurement data shown is automatically copied into the measurement panel, and inspectors can then "close out" of that

specific defect. This close-out action automatically saves all measurements, photos and notes to the AR device for future export.

Figure 24. Taking AR measurements is done using bare-handed interaction, whereby inspectors simply touch the bridge surface to establish points of interest

View Condition State Details

As mentioned above, the current inspection side of the panel also affords access to reference data regarding condition state. This feature provides textual descriptions (Figure 25) and photographs (Figure 26) of example defects at a given condition state. The information shown in this panel is defect-specific and available on-demand to assist inspectors in recalling quantitative and qualitive AASHTO guidelines regarding defect condition state designation. At this time, the prototype has textual information for all defects, but images only for cracking, spalling, and corrosion defect states.

Figure 25. Example of information displayed when "View Condition State Details" is chosen in the context of a Reinforced Concrete Deck – Cracking (RC and Other).

Figure 26. Shows example photos provided when "View Condition State Details" is chosen in the context of Reinforced Concrete Deck – Cracking (RC and Other). Photos obtained from the AASHTO Bridge Reference Manual.

General Observations on the Interface

A majority of the interface is operated via button clicks, a simple AR interaction which is intuitive and requires virtually no experience to learn. Creating and moving objects in 3D is slightly more difficult and requires practice. However, we have observed that the learning curve is shallow, and most novice users become proficient in less than an hour. The manipulation of objects can be affected by the aforementioned issues in the hand tracking or spatial mapping, but these are not considered interface issues per se, but rather hardware and operation issues.

One practical limitation of AR interfaces in general is the typing of text and/or numbers using a virtual keyboard. This can be cumbersome and slow to execute. In our interface, this is not considered a major obstacle since number pads could be replaced with sliders and keyboards with speech to text, limiting the need for direct text/number entry.

World Localization and World Persistence

To test our world localization and world persistence systems we discussed earlier, World Locking Tools was implemented into the application and several tests were performed to determine its usefulness. Firstly, we traveled to a steel-girder bridge near the Virginia Tech campus and placed virtual cubes all around the bridge in conspicuous, easily recalled locations and then tested the system's ability to world persist by returning to the same bridge several times during the semester where each time the application was started, and the location and type of objects was ensured. When we tested the system, it was able to successfully recall 100% of virtual objects (correct object and correct location). Figure 27 below shows the red cubes, which were our chosen virtual object, to test world persistence and tracking, overlayed onto our test bridge near Virginia Tech.

Figure 27. Virtual objects at Virginia Tech test bridge used to ensure the persistence of virtual objects in the tracking system.

In addition to the real-life persistence testing performed at a bridge site, we had previously performed a more general tracking error experiment where we walked around the Virginia Tech campus (approximately 1,100 ft straight-line distance) using an application utilizing the World Locking Tools software system and tested to see the overall tracking error incurred over that large distance (Figure 28). Tracking error on an AR headset can manifest itself as a virtual object shifting or drifting from its previously placed position. In this test, we placed virtual red cubes on specific physical environment places (like a bike rack) and then walked back and forth along the 1,100 ft straight-line distance and then returned to each virtual object to see if its position had shifted since its initial placement. What we found was that over that 1,100-ft path, the objects had shifted, on average, only about 6 in from their original position. This result is much better than the standard ±10% error discussed earlier and further proved that World Locking Tools was indeed a viable world tracking and persistence tool which could handle tracked environments at the general size we required.

It should be noted that the 6-inch error noted here does not extend to the crack measurements as these are fundamentally different processes. World localization is intended for use in placing the defect orbs (Figure 16) at the approximate defect location and recalling them during subsequent visits. Measurements use the local environment mesh generated by the HoloLens 2. Measurement accuracies are discussed in the following two sections.

Figure 28. Test on Virginia Tech campus of the World Locking Tools system to see the average tracking error (virtual object drift) incurred over around 1,100 feet of Straight-line headset movement tracking.

Computer Vision Development

Field Architecture

A limitation of the HoloLens 2 headset is that it is optimized for indoor use, so its functionality may be reduced in outdoor settings. The research team did not expend significant effort trying to improve operation performance, as this is linked primarily to hardware and software development within the device. The researchers believe that outdoor performance will steadily improve with new hardware releases as the mixed reality outdoor application space continues to grow. For completeness, we mention some of observed hardware limitations below.

Some of the features that leverage the headset's infrared cameras (also referred to as long- and short-throw sensors) could be affected by sunlight. This included the hand tracking feature, which could sometimes suffer from an offset to the physical hand position. At worst, this created an inconvenience, but could be resolved by visualizing the hand-tracking feature, allowing the user to adjust for the offset. The current system can run continuously for 1-2 hours, depending on outside temperature, which may not be sufficient for a full inspection. This could be extended via an external power bank and will likely improve in subsequent HoloLens generations.

As the existing hardware could not facilitate direct deployment of ML models, a server-client data transfer system was employed. This system was tested extensively for data transfer speeds and reliability with different distances and environments. In addition to qualitative testing at bridges, Figure 29 shows the results of a quantitative data transfer test in and around a concrete building on the Virginia Tech campus. The server was set up in an office and a small text file was sent to the HoloLens 2 every 1 second. For each transfer, the physical location of the HoloLens was recorded as well as the total data transfer time in milliseconds. For Figure 29a, the transfer time is plotted against the straight-line distance from the HoloLens 2 to the server. Figure 29b shows the physical position of the HoloLens 2 plotted in 3D space as a point cloud, where each point is color coded by the transfer time.

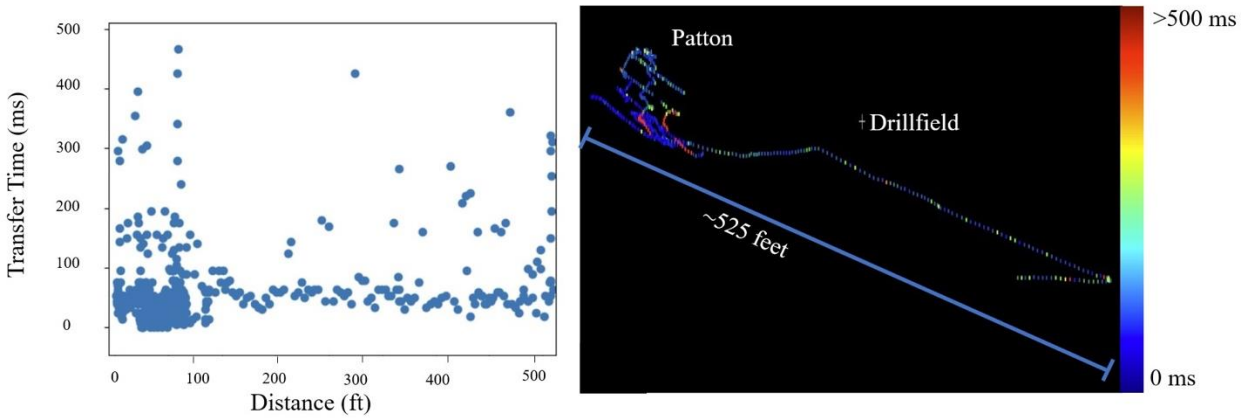

Figure 29. Data transfer speeds for text files. a) Plotted as transfer time vs distance and b) Plotted spatially in 3D with color representing transfer time; red points are failed transfers in stairwells of the building.

This experiment showed that the chosen server-client setup could send data with transfer times ranging from 100-500 milliseconds for distances up to 525 ft. As this test involved small text files, this can be interpreted as the "server response time" or the overhead delay to any data transfer. The experiment was repeated for low resolution images of approximately 100 KB and showed similar transfer times. Full resolution images of size 3 MB were found to have transfer times ranging from 1.8 - 6.6 seconds. For reliability, all transfers outside of the building were successful even at the maximum 500 ft range tested. However, inside of the building there were clusters of failed transfers in the stairwells, implying that some structural materials may block communication.

Projection of Computer Vision Results

When testing existing methods of projecting computer vision results onto physical surfaces, we observed translational offsets of approximately 2-4 inches. This level of precision is adequate for many computer vision tasks such as attaching labels to detected objects but was found to be inadequate for projecting crack annotations. Figure 30a shows an example of the precision of existing projection approaches, using a basic blob detection algorithm where the centroids of black dots are recognized in the image and projected onto the physical surface as green spheres.

Through our novel projection method, we observed translational offsets on the order of 0.02 in for crack centerline projections, demonstrating the precision of our approach. Figure 30b shows the precision for our projection approach using the same blob detection test case with red spheres. Figure 31 shows various additional projection results for crack centerlines.

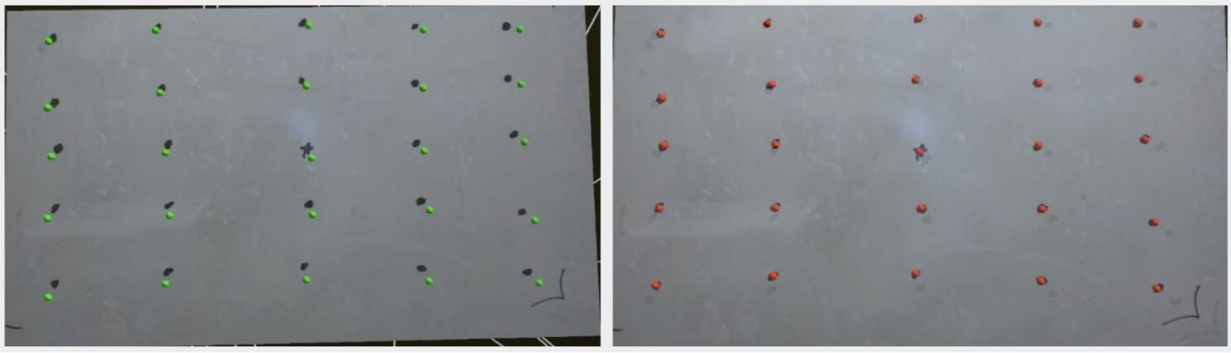

Figure 30. Precision of projection approaches plotted using a blob detection algorithm for detecting black dots on a white background. a) Precision of existing projection approaches, green spheres are the projection results and should be centered on the black dots and b) Precision of our projection approach.

Crack Detection and Documentation

One of the motivations for incorporating ML based automation into the field environment was to identify and mitigate the variability of these algorithms. Taking a photo from a different distance or angle can drastically affect the result, as shown in Figure 32. This is the motivation for incorporating inspector collaboration with these algorithms rather than independent automation.

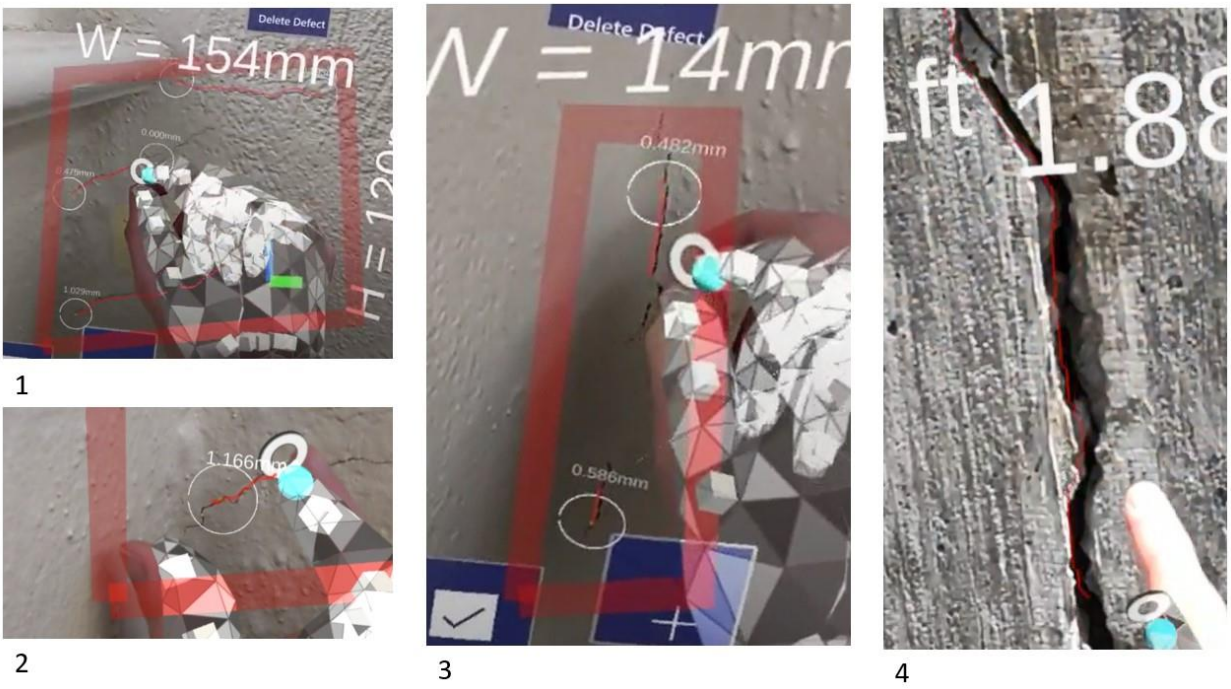

Figure 31. Precision of our projection result for aligning crack centerlines over various physical cracks. Images 1-3 show hairline cracks ranging from 0.02-0.04 inches in width (labeled as 0.5-1 mm), with observed offsets of approximately 0.02 inches. Image 4 shows the precision for a larger crack of approximately 0.5 inches in width.

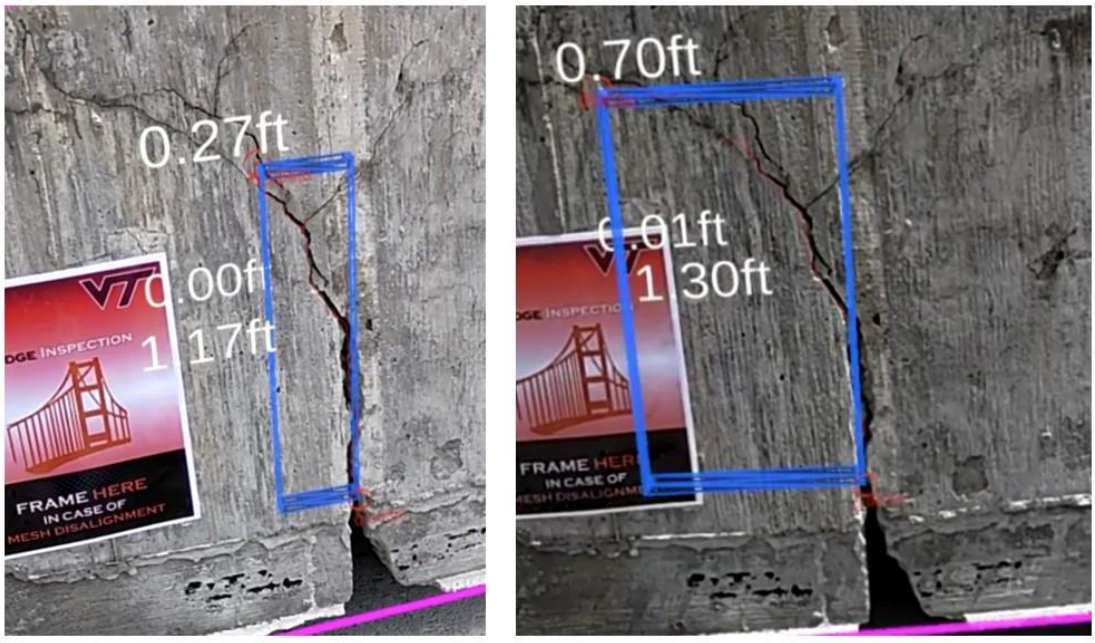

Figure 32. Example of variability in the computer vision crack measurement algorithm. Taking the photo from different distances and angles can lead to variable performance due to the underlying ML variability. Note that both a) and b) underestimate the affected crack area.

The model used in this study takes approximately 0.2 seconds to evaluate an image. Post processing of this result takes 1-2 seconds. As previously mentioned, image data transfer was found to take 1.8-6.6 seconds based on distance. The final result was returned to the user approximately 3-8 seconds after taking the picture (based on distance from server), this included data transfer times between the HoloLens 2 and the server, as well as all image processing steps on the server. We found this approach to work well for smaller cracks, including hairline cracks as small as 0.02 inches when the photo was taken close to the surface. However, the ML model struggled to detect very large cracks (0.5 inches and larger), which prevented this approach from measuring those cracks accurately.

User Interaction Experiment

The goal of the user interaction experiment was to understand how the use of defect documentation automation impacted the inspection performance. We studied both objective and subjective data. Objective data collected included the time spent on each crack documentation task and measurements of crack height, width, and thickness. Subjective data encompassed perceived workload, assessed through the NASA-Task Load Index (TLX) survey, and system usability, measured via the System Usability Scale (SUS). Furthermore, a post-study questionnaire was administered, which gathered users' preferred interaction method and free-form comments on the overall experience. All statistical analysis results can be found in the supplementary materials.

Subjective Results

For the perceived workload as measured by the NASA TLX (Figure 33), a repeated-measures ANOVA testing was conducted to assess the impact of the type of interaction method on perceived workload. The scores were found to be normally distributed, as determined by Shapiro-Wilk's test, and there was homogeneity of variances, as assessed by Levene's test ($p = 0.265$). The means and standard deviations for all data are presented in the supplementary materials. The repeated-measures ANOVA revealed that the mean perceived workload scores differed significantly across the five interaction methods ($p = 0.0002$). Paired samples t-tests were then conducted to compare differences in perceived workload for each interaction method. All of the AR interactions were found to have lower mean perceived workload than the Tape baseline, however, only the two most automated interactions (CV refines Human and Fully Automated) were significantly lower than Tape. In this study, the Bonferroni correction was applied to adjust the alpha value used for determining significance in the t-tests. This adjustment was made using the formula:

$$\alpha^* = \frac{0.05}{5} = 0.01 \tag{3}$$

where α^* represents the adjusted threshold. This computation was performed for the five interaction methods under consideration, ensuring a more stringent criterion for statistical significance.

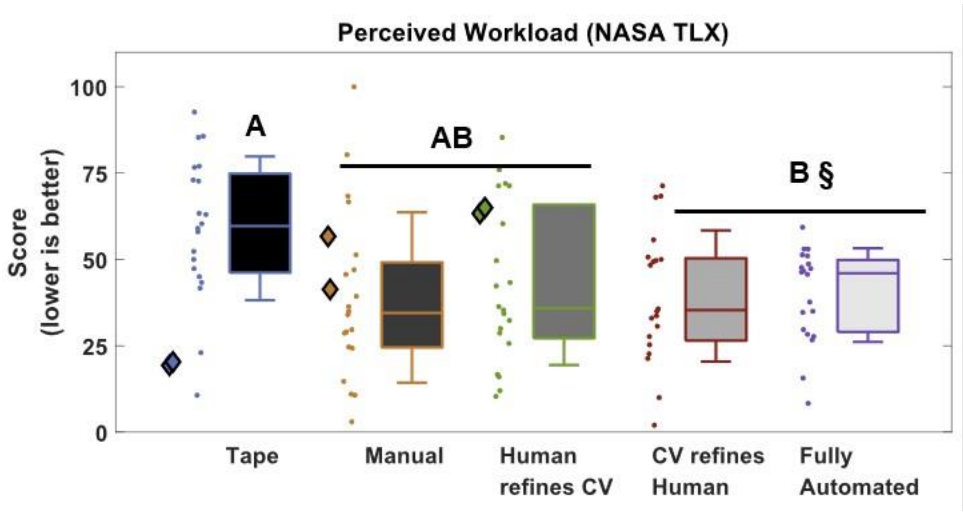

Figure 33. Box plot and associated scatter data of Perceived Workload (TLX) - Diamonds represent lab inspector data (not collected for the last two methods), while dots represent student data. Error bars are standard deviation. Groups which do not share a letter have statistically significant differences. Groups with the § symbol are significantly different from the baseline (Tape).

Referencing the perceived workload data in Figure 33, the two most automated interaction methods (CV refines Human and Fully Automated) were associated with significantly lower perceived workloads compared to the use of Tape, indicating a better outcome as lower workload scores are preferable. The less automated methods (Manual AR and Human refines CV) exhibited mean perceived workloads comparable to the more automated approaches, though they were not significantly lower than the Tape baseline ($p = 0.015$ and $p = 0.025$, respectively, against a threshold of $\alpha^* = 0.01$). This suggests that a larger participant sample might have produced significant differences. The inspector data (diamonds in Figure 11) show the opposite trend: increased perceived workload with increasing automation, indicating a worse outcome. This is to be expected and showcases the inspectors show a bias towards the Tape baseline condition due to years of experience using that interaction.

Similar to the analysis for perceived workload, a repeated-measures ANOVA was conducted to assess the effect of the interaction method type on system usability (Figure 34). The scores were found to be normally distributed, as determined by Shapiro-Wilk's test. However, the assumption of homogeneity of variances was not met, as indicated by Levene's test ($p = 0.013$). Given the violation of homogeneity of variances, the Greenhouse-Geisser adjustment was applied to the repeated-measures ANOVA. This adjustment led to the conclusion that the mean system usability scores did not significantly differ across the five interaction methods ($p = 0.683$, $p(GG) = 0.596$).

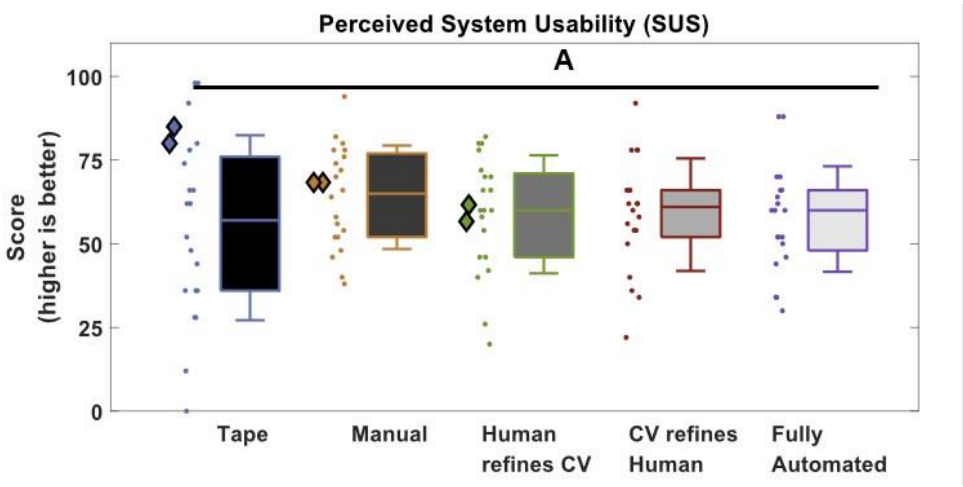

Figure 34. Box plot and associated scatter data of Perceived Usability (SUS) - Diamonds represent inspector data, while dots represent student data. Error bars are standard deviation. All methods share the letter "A", indicating no statistically significant differences.

Regarding the perceived system usability data presented in Figure 34, all interaction methods were deemed similarly usable, with the Manual method scoring slightly higher than the others (indicative of better system performance and user satisfaction), albeit not to a statistically significant degree ($p = 0.596$). This outcome is particularly encouraging for AR interactions, suggesting that despite common issues such as glare and hand tracking difficulties, the convenience offered by AR systems appears to offset these challenges. As with the workload data, inspector usability results show a slight decreasing trend (worse outcome) with increasing automation.

Last, as part of the post study questionnaire the user's preferred interaction method was recorded (Figure 35). A portion of this questionnaire included free-form comments from the users; the Manual interaction stands out as the favorite, followed by Human refines CV, then Tape. The student users mentioned doubts of the measurement accuracy and complaints of the CV refines Human interaction's O-Snap functionality. One lab inspector preferred Tape, while the other lab inspector preferred the Manual interaction. The first inspector mentioned that the CV was impressive when conditions were cooperative, while the second inspector said the system showed a "great deal of potential." Both lab inspectors also mentioned that the CV was accurate when it worked, but often would produce failed crack detections. This response aligns with our conclusion of needing a "Human in the Loop" for these types of automation as the human can correct the result when needed, while the automation can save time and reduce workload when correction is unnecessary.

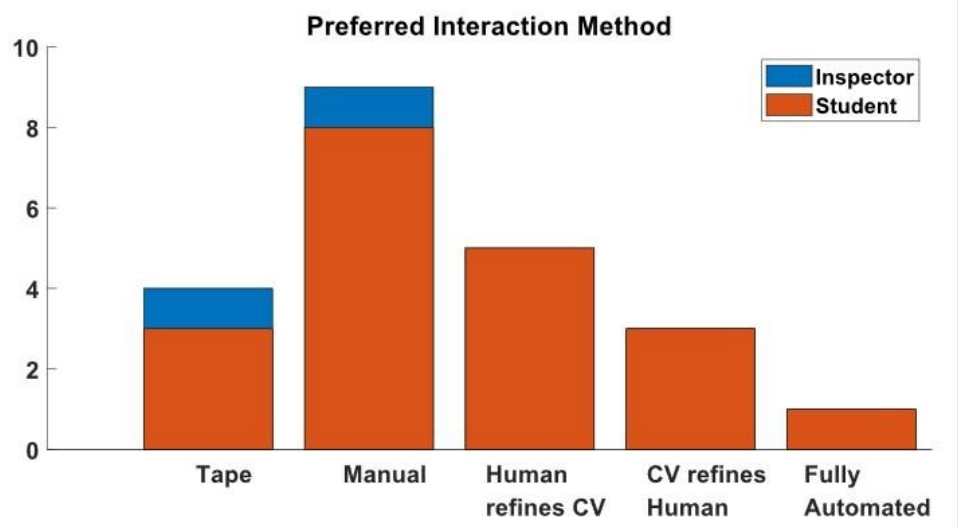

Figure 35. Student and inspectors' preferred interaction method.

Objective Results

Completion time was measured in seconds for each individual crack documentation task. These times were aggregated across the five tasks for each interaction method. Times were not normally distributed as assessed by Shapiro-Wilk's test. After applying a natural log transformation to the completion time data, the transformed times were found to be normally distributed as assessed by Shapiro-Wilk's test. The transformed times were found to have homogeneity of variances as assessed by Levene's test ($p = 0.221$). A repeated-measures ANOVA determined that the mean transformed times differed significantly across the five interaction methods ($p = 5.4e\text{-}06$).

A paired samples t-test was performed to compare differences in transformed completion time for each interaction method. Manual and Human refines CV did not have significantly different times to Tape. CV refines Human and Fully Automated were found to be significantly faster than Tape, although CV refines Human was not significantly different than Manual.

For completion time (Figure 36), the less automated interactions (Manual and Human refines CV) showed similar completion times to traditional Tape interaction. For the more automated interactions (CV refines Human and Fully Automated) both are shown to be significantly faster than traditional Tape ($p = 0.007$ and $p = 0.005$ respectively). This result is expected as both of those interactions required less input from the user; neither required the user to measure Crack Width, and Fully Automated only required a photo. Inspector data shows slower completion times on average with AR compared to Tape.

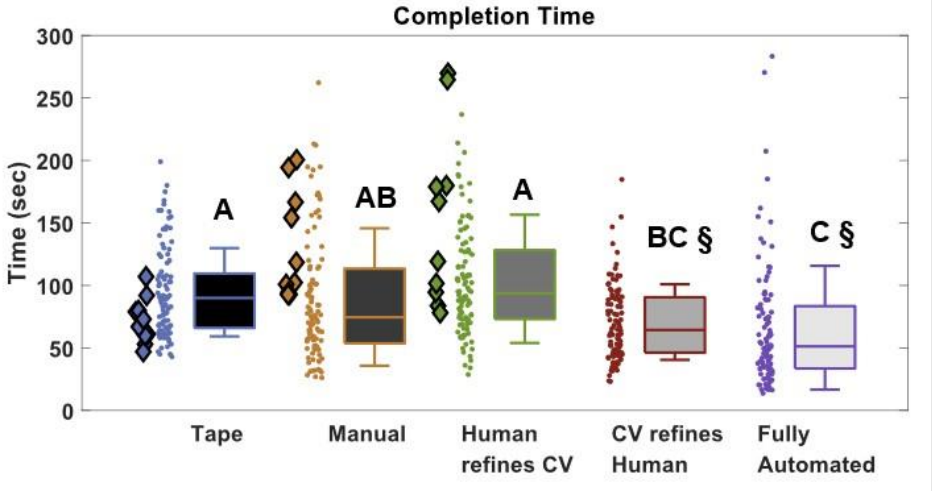

Figure 36. Box plot and associated scatter data for completion time for each of five cracked regions - Diamonds represent inspector data, while dots represent student data. Error bars are standard deviation. Groups which do not share a letter have statistically significant differences. Groups with the § symbol are significantly different from the baseline (Tape).

The measurement data collected in this study encompassed the length and height of the affected crack region, along with the largest crack width observed within this region. To facilitate statistical analysis, the length and height were multiplied to derive an area metric. For both the area and crack width datasets, the coefficient of variation was calculated for each interaction method, shown in Figure 37. Each data point corresponds to a distinct crack region from Figure 8 (five total). The data from these regions were not aggregated at this stage. The dataset's small size, with just five crack regions, limits further statistical analysis due to insufficient data points.

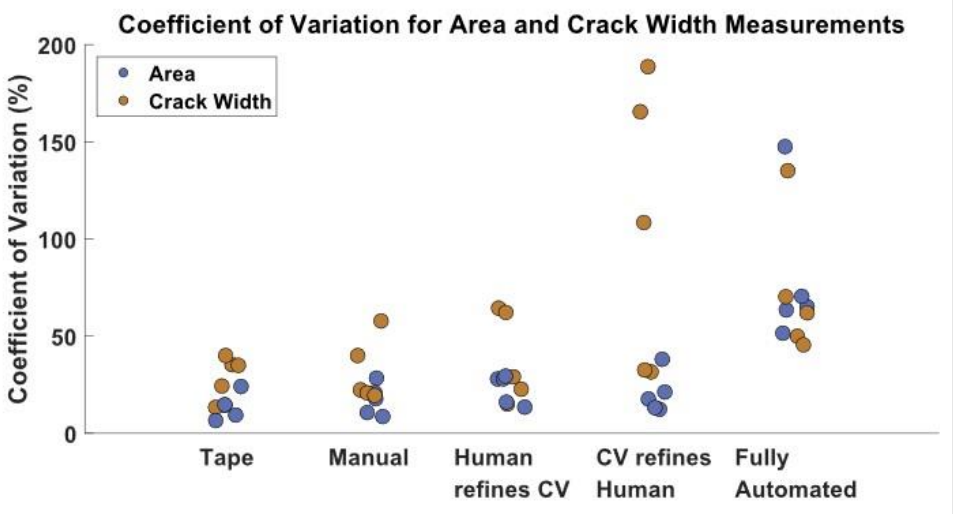

Figure 37. Coefficient of variation for area and crack width measurements.

A previous study by Washer et al., (2020) with 40 inspectors assessing the same bridge found area measurement variations ranging from as low as 18% to often exceeding 100%. Their methodology, requiring inspectors to both locate and measure cracked regions, introduced more variability than our approach, which focused solely on measuring predefined regions. As depicted in Figure 37, our experiment's variation for area and crack width measurements under the Tape condition ranged from 10-40%, aligning closely with the findings of Washer et al., (2020), albeit at the lower end. Due to the limited dataset (five regions), a comprehensive statistical analysis was not feasible; however, we observed consistent variation across all interaction methods for area measurements, except for the Fully Automated condition, which exhibited a significantly higher variation, suggesting lesser reliability. Similarly, crack width measurements showed consistent variation for Tape, Manual, and Human refines CV methods. In contrast, methods automating crack width measurement (CV refines Human and Fully Automated) revealed a substantial increase in variation, indicating lower reliability for these measurements. Coefficient of variation was not computed for inspector data due to the small sample size (two inspectors).

Figure 38 illustrates the dataset analyzed for area error relative to users' Tape (self error), which demonstrated a normal distribution according to Shapiro-Wilk's test, although Levene's test indicated a variance homogeneity violation ($p = 0.001$). The Friedman rank sum test demonstrated significant discrepancies in mean area errors across five interaction methods ($p = 2.3e-08$), with the Wilcoxon signed rank test identifying a notable difference between the Fully Automated interaction and other methods. No significant variances were observed among Manual, Human refines CV, or CV refines Human interactions.

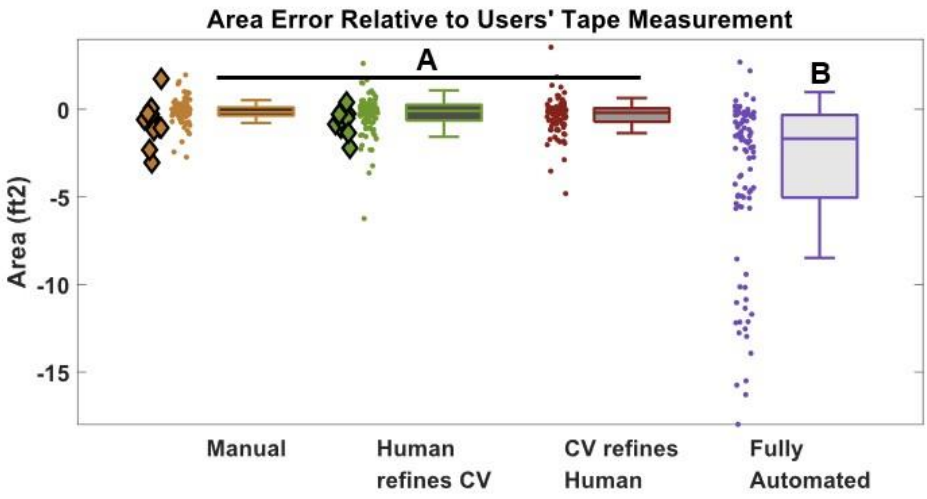

Figure 38. Crack Area Error relative to users' Tape - Diamonds represent inspector data, while dots represent student data. Error bars are standard deviation. Groups which do not share a letter have statistically significant differences.

Similarly, for area error relative to ground truth (Figure 39), normal distribution was confirmed, alongside a variance homogeneity violation (p value = 0.015). The Friedman test highlighted significant mean area error differences across interaction methods ($p = 4.29e-09$),

with the Wilcoxon test pinpointing a significant discrepancy between the Fully Automated interaction and others, including Tape. No notable differences were detected among Tape, Manual, Human refines CV, or CV refines Human interactions.

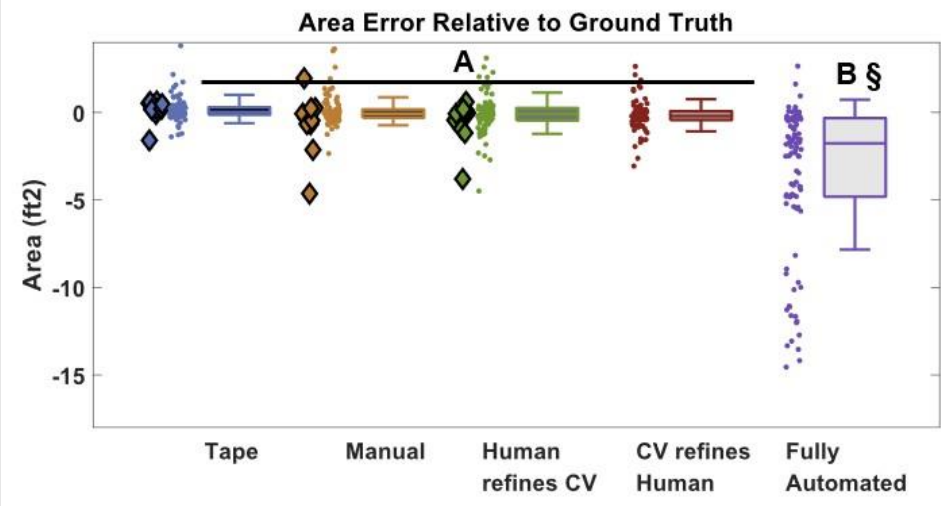

Figure 39. Crack Area Error relative to ground truth - Diamonds represent inspector data, while dots represent student data. Error bars are standard deviation. Groups which do not share a letter have statistically significant differences. Groups with the § symbol are significantly different from the baseline (Tape).

Figure 40 illustrates the dataset analyzed for crack width error relative to user's Tape, which revealed a normal distribution, confirmed by Shapiro-Wilk's test, as well as homogeneity of variances by Levene's test (p value = 0.942). A repeated measures ANOVA showed significant differences in mean crack width error relative to users' Tape (p = 3.15e-11), with Pairwise T-test revealing significant differences between the CV refines Human and Fully Automated interactions compared to Manual and Human refines CV interactions.

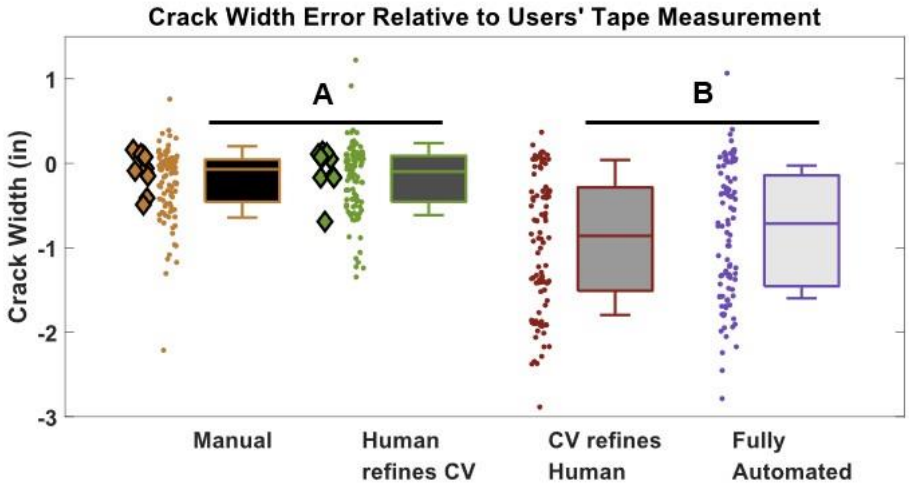

Figure 40. Crack Width Error relative to users' self Tape - Diamonds represent inspector data, while dots represent student data. Error bars are standard deviation. Groups which do not share a letter have statistically significant differences.

For crack width error relative to ground truth (Figure 41), the Shapiro-Wilk's test indicated a violation of normality (p = 3.57e-07), while homogeneity of variance was confirmed by Levene's test. The Friedman rank sum test demonstrated significant discrepancies in mean crack width errors across the five interaction methods (p = 4.71e-36), with the Wilcoxon signed rank test identifying larger mean crack width error between CV refines Human and Fully Automated conditions as compared to all other interactions. Additionally, Manual and Human refines CV were found to have significantly more error than Tape, but less error than CV refines Human and Fully Automated interactions. It should be noted that crack width error is exaggerated in these results since some cracks were extreme, as wide as 3 inches.

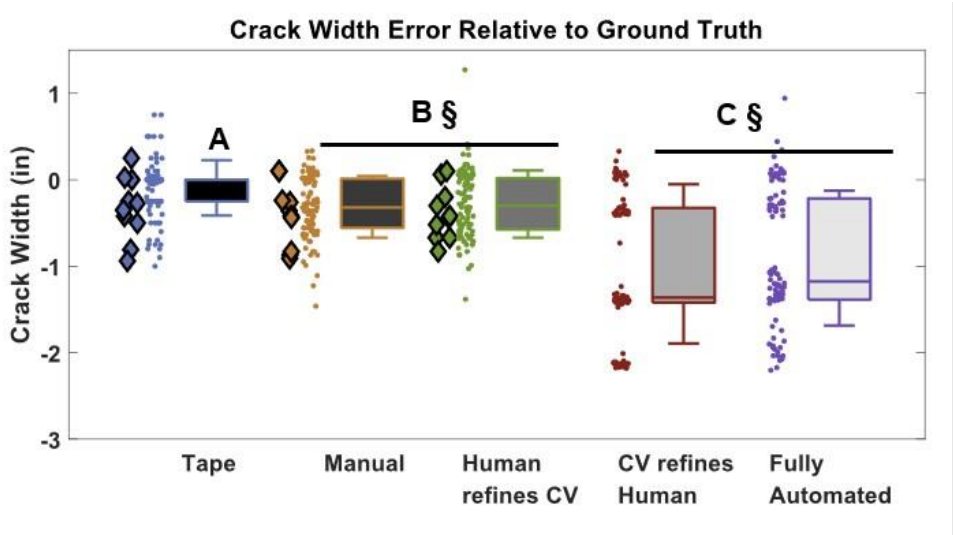

Figure 41. Crack Width Error relative to ground truth - Diamonds represent inspector data, while dots represent student data. Error bars are standard deviation. Groups which do not share a letter have statistically significant differences. Groups with the § symbol are significantly different from the baseline (Tape).

For the area and crack width error metrics, we observed similar trends between the errors relative to the user's own Tape and those relative to ground truth. This suggests that the user's inexperience with crack documentation tasks did not significantly impact the experiment's results. The discussion will henceforth focus on the error relative to ground truth data results, enabling comparison with the baseline Tape condition. In line with the trends seen in the coefficient of variation data, area error relative to ground truth did not significantly differ among interaction methods, except for the Fully Automated interaction, which was associated with significantly more area error. Specifically, the Fully Automated condition had a mean area error of approximately 2.5 ft² below ground truth, representing an unconservative underestimation. This finding indicates that AR area measurements can effectively substitute for a physical Tape measure and can be *partially* automated without a significant increase in error. Conversely, measurements of crack width showed an increase in error for Manual and Human refines CV interactions (~0.2 inches additional mean error compared to Tape) while the CV refines Human and Fully Automated showed a larger increase in error (~0.8 inches additional mean error compared to Tape). Both of these results represent crack widths smaller than the actual width,

which is an unconservative result. This suggests a discrepancy in participants' visual interpretation of a holographic sphere for crack width measurement tasks, indicating that this measurement should continue to use the physical crack comparator for optimal accuracy. Inspector data shows similar trends to student data for area and crack width error metrics, once again implying that the student user's lack of inspection experience did not significantly affect the error metric results.

The large variation in crack width error in both CV refines Human and CV only interaction techniques were driven largely by the practical limitations of the implemented CV algorithm in the experimental conditions. In particular, the crack detection portion of the algorithm struggled to detect both thin and thick cracks at the extremes, leading to generally shorter crack outlines. This was influenced by distance to the crack: smaller cracks would be more likely detected by taking a close-up photo, whereas large cracks would be more likely detected by taking a photo further away. This is a known problem with crack detection algorithms, as they generally work best for cracks within a certain range of pixel widths. If a crack was detected, however, the measured crack widths via our custom approach were generally not sensitive to distance from the crack. Large crack width measurement errors would occur when the ground truth maximum crack width was not detected by the algorithm.

Qualitative Field Validation

Observations

For the bridge subjected to this field test, only one section exhibited cracking within arm’s reach, located on the inner side of a concrete bridge girder. The area under the bridge was poorly lit, necessitating the use of a flashlight to view the crack, which could only illuminate a small portion of the crack at any given time. Consequently, the CV could only identify the illuminated portion of the crack. A broader light source was recognized as necessary for utilizing this feature under similar conditions.

The simulated field inspection found that each crack documentation took 2-3 minutes per inspector; however, this duration included delays due to various minor interface bugs. Given that the training time was short, approximately 15 minutes per inspector, it is reasonable to expect that this time would decrease with extended training and experience.

Inspectors found the interface to be intuitive and easy to learn, reporting no safety concerns and appreciating the hands-free nature as safer compared to traditional tablet workflows. The inclusion of condition state example definitions and photos was well-received, with inspectors highlighting their potential benefits for junior inspectors. While the button activation generally functioned well, a preference was expressed for interface elements to be positioned closer to the user and off to the side. Hand tracking exhibited occasional issues near bridge surfaces. The HoloLens 2 device caused reflections on the lenses from areas behind the user, akin to a welding mask, which were sometimes distracting. One inspector, having

undergone hand surgery, had trouble performing the “flat palm” gesture required for adjusting the virtual menus.

One inspector suggested incorporating functionality seen in their tablet application, which displays the difference between measurements from previous inspections and the current one, into the AR inspection tool. Additionally, the ability to dynamically customize font and window sizes was identified to better accommodate inspectors with reduced eyesight. The need for a feature to delete defects, whether mistakenly created or remediated, was emphasized for future versions, along with the request for a virtual Tape measurer.

Potential future applications of this type of bridge inspection tool were identified, including scour area and riverbed profile measurements. The documentation of warping measurements on steel elements subjected to impact damage, requiring six-foot measurements, was noted as a task that would be significantly facilitated by the 3D sensing capabilities of the AR device. Moreover, the ability to place points in AR to mark sound/unsound locations was suggested as a helpful feature for documenting delaminated regions identified through concrete sounding.

Cost Benefit Analysis

As of 2024, a hard-hat integrated HoloLens 2 head set costs roughly $5,000. For automation integration, an additional $2,000 would be required to purchase a laptop computer and Wi-Fi router to run the machine learning models.

The cost of training is likely to equal up to 3 hours of personnel time, depending on familiarity with mixed reality devices. During user experience testing, which took 2-3 hours per participant, it was observed that novice user proficiency increased substantially over this time period.

The cost of storage per bridge is largely dependent on the amount of data desired. Assuming 100 documented defects per bridge, with each defect having 3 MB of associated images and documentation, each bridge will generate roughly 0.3 GB of data per inspection. Assuming cloud storage at $2 per month per 100 GB, this results in approximately $0.07 per year per inspection. This would compound roughly every two years as inspections are repeated.

The speed of use of the system was not significantly different from the use of a traditional measuring tape. However, significant time savings are expected in the office regarding report generation and quality assurance. In this regard, we estimate that the system could save approximately one hour per inspection.

The added value of data is difficult to estimate at this point in the research and would require trial implementations for real-life use cases. Thus, for now this is not considered, but it will certainly be positive considering the added granularity that the HoloLens 2 data provides.

Taking these conservative estimates into account and assuming that 1 personnel hour equals $30 (factoring in salary, travel, etc.), it would take approximately 240 bridge inspections

to break even with the initial investment for one system. This is less than a year's worth of inspections, factoring in only one hour saved per inspection. If additional efficiencies and added value can be identified, the break-even number of inspections could decrease dramatically.

CONCLUSIONS

AR interface design

1. *Menu-based data entry via an AR interface was an effective and intuitive way to record inspection information in a digital format.* The interface was reviewed in the field by a team of three inspectors, who agreed that it conformed well to the AASHTO reporting requirements and that it was easier to navigate than the current InspectX system. (Objective 1.1, 1.2)

2. *A majority of issues with AR operation were related to hardware limitations in outdoor settings, like poor hand-tracking and sensor saturation.* These should be carefully considered but should become less burdensome as the technology matures. (Objective 1.1, 1.2)

3. *AR-based manual length and height measurements were effective, although slightly less accurate than physical tape measurements; width measurements using a variable-width spheres were not user-friendly and decreased accuracy, thus are not recommended*. (Objective 1.3)

World Localization

1. *The HoloLens 2 world mapping tools can be leveraged to project computer vision results onto the physical space with sub-millimeter precision.* This allows precise overlaying of crack segmentation outlines, measurements, and area bounds. (Objective 2.1)

2. *The position of an annotation will drift as the user moves about the physical space, the larger the space, the larger the drift. This can be mitigated through the use of World Locking Tools to less than a 6-inch deviation over 1,100 ft.* Practically, this means that if an inspector leaves a defect location and comes back (either in the same or subsequent inspection), the annotations could have shifted by a few inches. This is not a concern for marking the general defect location but could impact the accuracy of historical overlays. The latter case could be addressed by a computer vision-based correction when new annotations are generated. (Objective 2.2)

Automated Defect Annotations

1. *The current AR prototype supports real-time projection of CV results in the field via a laptop server with GPU and local Wi-Fi network (router or hotspot).* The hardware requirements are necessary to avoid the limitations of running ML models on the HoloLens 2 platform. (Objective 3.1)

2. *The current AR prototype will support automated virtual crack length and height measurements, but requires occasional manual intervention for best results.* In particular, the automated crack detection algorithm can miss the extremes of a crack, generally leading to underestimation. (Objective 3.1)

3. *The current AR prototype will support automated virtual crack width measurements, however inconsistent results lead us to recommend a manual approach.* Crack width measurements require both crack detection and edge detection, making them more sensitive to lighting conditions and other sources of noise. Errors tended to be sporadic in nature (either very accurate or completely off). Nevertheless, the method developed in this research is one of the first demonstrations of crack width measurements in true field conditions (versus simply crack detection). Further research is required to drive this capability forward. (Objective 3.1)

4. *For novice users, all AR techniques reduced the perceived workload to traditional tape measurements and were generally preferred. The opposite was true for inspectors who perceived a higher workload with AR and preferred the tape interaction more.* Due to the inspectors' existing familiarity with the tape measure method, this result confirms that training and experience play a large role in preference and cognitive load. Taking the novice users as an unbiased case (no experience in either), the research suggests that AR has the potential to alleviate the cognitive load of inspection tasks. (Objectives 3.2, 3.3)

5. *With respect to time and accuracy, experienced inspectors were better than novice users using traditional techniques, but similar using the AR-based techniques.* Again, this suggests that experience is an important factor in quantifying efficiency gains. The performance metrics recorded in this work are the result of approximately 30 minutes of AR training, so it is expected that there is large room for improvement. (Objective 3.2, 3.3)

6. *High levels of automation could speed up documentation tasks by about 20% but in their current state incur unacceptable levels of error.* Improvements will require additional research into more robust computer vision, which was not a primary goal of this research. (Objective 3.2, 3.3)

RECOMMENDATIONS

1. *Refinement of AR interactions and hardware architecture/setup for remote field use.* We recommend that VTRC should work with a select group of VDOT inspectors to fine-tune both the data entry interface format and the hardware architecture required for running AI assistance tools. The former should focus on further user experience studies with a broader inspector base and involve adjustments and improvements to the existing interface. The latter should focus on trialing hardware solutions that would require

minimal setup time: ideally by turning on a small, networked PC server housed inside a work vehicle and walking to the bridge site with only the AR headset on.

2. *Develop a set of standards and procedures for exporting inspection data, streamlining report generation, and integrating with existing software* (*InspectX, AASHTOWare Bridge, or others).* This recommendation focuses on extending the AR-based workflow to the office to explore its potential as an end-to-end solution. This work would leverage the already-developed hierarchical data structure to facilitate ease of integration with reporting systems. As in the previous recommendation, we suggest that VTRC work with a select group of VDOT stakeholders to validate the entire inspection to report workflow and assess potential time savings.

3. *Expand the scope of automated defect annotations to include corrosion, section loss, spalling, delamination, and others.* This recommendation is aimed at broadening the scope of AR-enabled tasks to demonstrate AR's potential as a complete bridge inspection tool. This would entail further research into state-of-the-art defect detection algorithms as well as custom development of measurement extraction approaches like the one developed in this work for crack area and width extraction. Additional research could focus on optimizing these algorithms for accuracy in variable field conditions.

IMPLEMENTATION AND BENEFITS

Researchers and the technical review panel (listed in the Acknowledgements) for the project collaborate to craft a plan to implement the study recommendations and to determine the benefits of doing so. This is to ensure that the implementation plan is developed and approved with the participation and support of those involved with VDOT operations. The implementation plan and the accompanying benefits are provided here.

Implementation

With regards to *Recommendation 1*, the Structure and Bridge Division does not believe that an augmented reality platform is sufficiently ready for bridge safety inspections in real-world conditions. Furthermore, the results of this research pertain to a single proprietary system. However, the Division sees the potential for further refining the hardware architecture and safety inspectors' interactions with a non-proprietary augmented reality system to support the full range of data collection following the *Nation Bridge Inspection Standards* and referencing relevant VDOT libraries containing archival reports and plan sets. Because these functionalities are inherently specific to VDOT, the Division feels that additional development is required for a platform that focuses on VDOT's Structure and Bridge Division's specific needs. Thus, this topic will be presented at the next Bridge Research Advisory Committee meeting where research priorities are considered for funding.

Regarding *Recommendation 2*, the Structure and Bridge Division certainly sees the need to incorporate the complete set of required inspection data gathered using generic augmented

reality (AR) and artificial intelligence (AI) tools into standard bridge safety inspection reports. However, full integration of this novel data collection into the Structure and Bridge Division's data system (such as *InspectX*) cannot take place until the results of *Recommendation 1* are fully implemented. Prior to this integration phase, the focus should be on understanding and addressing the hardware requirements necessary to create a local network for the AR device (headset). This includes coordination with the Virginia Information Technology Agency (VITA) and VDOT's Information Technology Division (ITD) concerning the procurement, deployment, storage, and management of essential equipment such as computers, virtual headsets, routers, or other network devices needed for field setups. A better understanding of the hardware and security requirements is vital to ensure the setup can be standardized for use by multiple teams across all districts in Virginia. Additionally, because this project only considered one virtual platform, there needs to be an exploration of non-proprietary options that cater to the Structure and Bridge Division's specific safety inspection needs. Assigning inspection teams to test hardware functionality will be part of this process. Thus, this topic will be presented at the next Bridge Research Advisory Committee meeting where research priorities are considered for funding.

Similarly, implementing *Recommendation 3* is not feasible until *Recommendations 1* and *2* are implemented. Thus, the Structure and Bridge Division will not act on *Recommendation 3* at this time. However, the Division does recognize that many more structural defect assessment capabilities need to be incorporated into the augmented reality / artificial intelligence platform in order to cover the range of scenarios bridge safety inspectors come across on a typical basis. Thus, the Division will encourage academia and the industry to expand capabilities with defect recognition and quantification so that VDOT can consider including them in a future inspection platform. Provided this future development and successful implementation stemming from *Recommendations 1* and *2*, the Structure and Bridge Division will support additional evaluation of the newly available technology compared with VDOT's inspection needs.

Benefits

While there are no immediate benefits to be gained from implementing the results of this particular research, this project has shown promise for the development of tools in the future that could improve the process by which VDOT conducts its bridge safety program, by way of:

- decreased duration of individual inspections, and thus improved safety for inspectors as well as less disruptions to the traveling public
- increased accuracy of the data collected during the inspections;
- improved deterioration tracking over time;
- automated inspection report generation;
- automated integration of inspection data with bridge management software.

Further developments of the AR and AI technology within the industry will help to actualize these benefits.

ACKNOWLEDGEMENTS

The research team gratefully acknowledges the individuals who supported this research through their essential insights. In particular, the technical review panel that guided this project through participation in regular review meetings and comments on the final report and implementation plan. This panel was composed of: Christopher R. Williams (project champion), P.E., Assistant State Bridge Engineer; James M. Lightfoot, P.E., Senior Inspection QA Engineer; Christopher A. Roberts, P.E., Assistant District Bridge Engineer, Hampton Roads District; and D. Spencer Estes, Senior Bridge Safety Inspector, Culpeper District.

Also, Eric Bianchi, for his help in implementing and troubleshooting the machine learning algorithms used in this study; Brett Frazer , for his participation in initial prototype testing and indispensable insight into the minds of inspectors; as well as David Kelly, David Smith, Benson Williams, and the rest of the Salem District Bridge inspection team for facilitating the many field visits that the team undertook. Our depth of understanding into the inspection workflow would not have been possible without their critical insight and advice. Lastly, all VDOT personnel who participated in the surveys and interviews.

REFERENCES

Al-Sabbag, Zaid Abbas, Yeum, Chul Min, and Narasimhan, Sriram. *Enabling Human–Machine Collaboration in Infrastructure Inspections through Mixed Reality.* Advanced Engineering Informatics, vol. 53, 2022a, 101709.

Al-Sabbag, Zaid Abbas, Yeum, Chul Min, and Narasimhan, Sriram. *Interactive Defect Quantification through Extended Reality*. Advanced Engineering Informatics, vol. 51, 2022b, 101473.

Aniculaesei, Adina, et al. *Graceful Degradation of Decision and Control Responsibility for Autonomous Systems Based on Dependability Cages*. 5th International Symposium on Future Active Safety Technology toward Zero, Blacksburg, Virginia, USA, 2019.

Bianchi, Eric, and Hebdon, Matthew. *Development of Extendable Open-Source Structural Inspection Datasets.* Journal of Computing in Civil Engineering, vol. 36, no. 6, 2022, 04022039.

fast-slow-still. *World Locking Tools Concepts*. Learn.microsoft.com, 19 July 2022, learn.microsoft.com/en-us/mixed-reality/world-locking-tools/documentation/concepts.

Federal-Aid Highway Act of 1968. 82 Stat. 815, Public Law 90-495, 1968, https://www.govinfo.gov/app/details/STATUTE-82.

Fitts, Paul M. *Human Engineering for an Effective Air-Navigation and Traffic-Control System*. 1951.

Fogerty, Brandon. *Holographic Photo Blending with PhotoCapture*. Unity Forums, 2016, https://forum.unity.com/threads/holographic-photo-blending-with-photocapture.416023/.

Gao, C., and Elzarka, H. *The Use of Decision Tree Based Predictive Models for Improving the Culvert Inspection Process.* Advanced Engineering Informatics, vol. 47, 2021, 101203.

Hu, D., Hou, F., Blakely, J., and Li, S. *Augmented Reality Based Visualization for Concrete Bridge Deck Deterioration Characterized by Ground Penetrating Radar.* Construction Research Congress 2020, American Society of Civil Engineers, Reston, VA, 2020, pp. 1156–1164.

InspectX — AssetIntel. https://www.assetintel.co/inspectx. Accessed 1 Nov. 2023.

Jakl, Andreas, et al. *Augmented Reality for Industry 4.0: Architecture and User Experience*. FMT, 2018, pp. 38–42.

Jung, Y., Kang, M., Jeong, M. M., and Ahn, J. *Network and Cluster Analysis on Bridge Inspection Reports Using Text Mining Algorithms.* Construction Research Congress 2022, 2022, pp. 492–501.

Li, Gang, et al. Automatic Tunnel Crack Detection Based on U-Net and a Convolutional Neural Network with Alternately Updated Clique. Sensors, vol. 20, no. 3, 2020, 717.

Malek, Kaveh, Mohammadkhorasani, Ali, and Moreu, Fernando. *Methodology to Integrate Augmented Reality and Pattern Recognition for Crack Detection.* Computer-Aided Civil and Infrastructure Engineering, vol. 38, no. 8, 2023, pp. 1000–1019.

Mascarenas, David DL, et al. *Augmented Reality for Next Generation Infrastructure Inspections.* Structural Health Monitoring, vol. 20, no. 4, 2021, pp. 1957–1979.

Microsoft. *World Locking Tools Documentation.* Learn.microsoft.com, 2024 learn.microsoft.com/en-us/mixed-reality/world-locking-tools/.

Mohammadi, P., Rashidi, A., Malekzadeh, M., and Tiwari, S. *Evaluating Various Machine Learning Algorithms for Automated Inspection of Culverts*. Engineering Analysis with Boundary Elements, vol. 148, 2023, pp. 366–375.

Montgomery, Douglas C. *Design and Analysis of Experiments*. 10th ed., Wiley, 2021.

Munawar, Hafiz Suliman, et al. *Civil Infrastructure Damage and Corrosion Detection: An Application of Machine Learning.* Buildings, vol. 12, no. 2, 2022, 156.

Nguyen, Hoang-Nam, Kam, Tai-Yan, and Cheng, Pi-Ying. *An Automatic Approach for Accurate Edge Detection of Concrete Crack Utilizing 2D Geometric Features of Crack.* Journal of Signal Processing Systems, vol. 77, 2014, pp. 221-240.

Pathak, J. Y. *Evaluating Trust in AI-Assisted Bridge Inspection through VR*. 2024.

Smith, A., Duff, C., Sarlo, R., and Gabbard, J. L. *Wearable Augmented Reality Interface Design for Bridge Inspection.* 2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), IEEE, 2022, pp. 497–501.

Topcu, O. K. *An AI-Assisted Bridge Inspection System: Ontology-Based Visual Question-Answering Methodology Using Large Language Models*. Ph.D. dissertation, Southern Illinois University at Edwardsville, 2023.

Wang, Wenhai, et al. *Internimage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions.* Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 14408–14419.

Washer, G. A., Hammed, M. M., Jensen, P., and Connor, R. J. *Quality of Element-Level Bridge Inspection Data.* Transportation Research Record, vol. 2674, no. 2, 2020, pp. 252–261.

Wickens, Christopher D., et al. *Engineering Psychology and Human Performance*. Routledge, 2021.

Xu, X., Yuan, C., Zhang, Y., Cai, H., Abraham, D. M., and Bowman, M. D. *Ontology-Based Knowledge Management System for Digital Highway Construction Inspection.* Transportation Research Record, vol. 2673, no. 1, 2019, pp. 52–65.

Yang, Xincong, et al. *Automatic Pixel-Level Crack Detection and Measurement Using Fully Convolutional Network.* Computer-Aided Civil and Infrastructure Engineering, vol. 33, no. 12, 2018, pp. 1090–1109.