USE OF PHOTOGRAMMETRY AIDED DAMAGE DETECTION FOR RESIDUAL STRENGTH ESTIMATION OF CORROSION DAMAGED PRESTRESSED CONCRETE BRIDGE GIRDERS

Yeshwanth Sai Neeli

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science
in
Civil Engineering

Carin L. Roberts-Wollmann, Chair

Matthew Hardy Hebdon

Ioannis Koutromanos

June 18, 2020

Blacksburg, Virginia

Keywords: Full-Scale Prestressed Concrete Bridge Girders, Corrosion Damage, Visual Inspections of Bridges, Photogrammetry, Structure from Motion, 3D Point Clouds, Textured Mesh Models, Crack Detection, Fully Convolutional Network, Spall Detection, 3D Damage Maps, Residual Capacity Estimation, Destructive Testing on Corrosion Damaged Bridge Girders

USE OF PHOTOGRAMMETRY AIDED DAMAGE DETECTION FOR RESIDUAL STRENGTH ESTIMATION OF CORROSION DAMAGED PRESTRESSED CONCRETE BRIDGE GIRDERS

Yeshwanth Sai Neeli

ABSTRACT

Corrosion damage reduces the load-carrying capacity of bridges which poses a threat to passenger safety. The objective of this research was to reduce the resources involved in conventional bridge inspections which are an important tool in the condition assessment of bridges and to help in determining if live load testing is necessary. This research proposes a framework to link semi-automated damage detection on prestressed concrete bridge girders with the estimation of their residual flexural capacity. The framework was implemented on four full-scale corrosion damaged girders from decommissioned bridges in Virginia. 3D point clouds of the girders reconstructed from images using Structure from Motion (SfM) approach were textured with images containing cracks detected at pixel level using a U-Net (Fully Convolutional Network). Spalls were detected by identifying the locations where normals associated with the points in the 3D point cloud deviated from being perpendicular to the reference directions chosen, by an amount greater than a threshold angle. 3D textured mesh models, overlaid with the detected cracks and spalls were used as 3D damage maps to determine reduced cross-sectional areas of prestressing strands to account for the corrosion damage as per the recommendations of Naito, Jones, & Hodgson (2011). Scaling them to real-world dimensions enabled the measurement of any required dimension, eliminating the need for physical contact.

The flexural capacities of a box beam and an I-beam estimated using strain compatibility analysis were validated with the actual capacities at failure sections determined from four destructive tests conducted by Al Rufaydah (2020). Along with the reduction in the cross-sectional areas of strands, limiting the ultimate strain that heavily corroded strands can develop was explored as a possible way to improve the results of the analysis. Strain compatibility analysis was used to estimate the ultimate rupture strain, in the heavily corroded bottommost layer prestressing strands exposed before the box beam was tested. More research is required to associate each level of strand corrosion with an average ultimate strain at which the corroded strands rupture. This framework

was found to give satisfactory estimates of the residual strength. Reduction in resources involved in current visual inspection practices and eliminating the need for physical access, make this approach worthwhile to be explored further to improve the output of each step in the proposed framework.

USE OF PHOTOGRAMMETRY AIDED DAMAGE DETECTION FOR RESIDUAL STRENGTH ESTIMATION OF CORROSION DAMAGED PRESTRESSED CONCRETE BRIDGE GIRDERS

Yeshwanth Sai Neeli

GENERAL AUDIENCE ABSTRACT

Corrosion damage is a major concern for bridges as it reduces their load carrying capacity. Bridge failures in the past have been attributed to corrosion damage. The risk associated with corrosion damage caused failures increases as the infrastructure ages. Many bridges across the world built forty to fifty years ago are now in a deteriorated condition and need to be repaired and retrofitted. Visual inspections to identify damage or deterioration on a bridge are very important to assess the condition of the bridge and determine the need for repairing or for posting weight restrictions for the vehicles that use the bridge. These inspections require close physical access to the hard-to-reach areas of the bridge for physically measuring the damage which involves many resources in the form of experienced engineers, skilled labor, equipment, time, and money. The safety of the personnel involved in the inspections is also a major concern. Nowadays, a lot of research is being done in using Unmanned Aerial Vehicles (UAVs) like drones for bridge inspections and in using artificial intelligence for the detection of cracks on the images of concrete and steel members.

Girders or beams in a bridge are the primary longitudinal load carrying members. Concrete inherently is weak in tension. To address this problem, High Strength steel reinforcement (called prestressing steel or prestressing strands) in prestressed concrete beams is pre-loaded with a tensile force before the application of any loads so that the regions which will experience tension under the service loads would be subjected to a pre-compression to improve the performance of the beam and delay cracking. Spalls are a type of corrosion damage on concrete members where portions of concrete fall off (section loss) due to corrosion in the steel reinforcement, exposing the reinforcement to the environment which leads to accelerated corrosion causing a loss of cross-sectional area and ultimately, a rupture in the steel. If the process of detecting the damage (cracks, spalls, exposed or severed reinforcement, etc.) is automated, the next logical step that would add great value would be, to quantify the effect of the damage detected on the load carrying capacity

of the bridges. Using a quantified estimate of the remaining capacity of a bridge, determined after accounting for the corrosion damage, informed decisions can be made about the measures to be taken. This research proposes a stepwise framework to forge a link between a semi-automated visual inspection and residual capacity evaluation of actual prestressed concrete bridge girders obtained from two bridges that have been removed from service in Virginia due to extensive deterioration.

3D point clouds represent an object as a set of points on its surface in three dimensional space. These point clouds can be constructed either using laser scanning or using Photogrammetry from images of the girders captured with a digital camera. In this research, 3D point clouds are reconstructed from sequences of overlapping images of the girders using an approach called Structure from Motion (SfM) which locates matched pixels present between consecutive images in the 3D space. Crack-like features were automatically detected and highlighted on the images of the girders that were used to build the 3D point clouds using artificial intelligence (Neural Network). The images with cracks highlighted were applied as texture to the surface mesh on the point cloud to transfer the detail, color, and realism present in the images to the 3D model. Spalls were detected on 3D point clouds based on the orientation of the normals associated with the points with respect to the reference directions. Point clouds and textured mesh models of the girders were scaled to real-world dimensions facilitating the measurement of any required dimension on the point clouds, eliminating the need for physical contact in condition assessment. Any cracks or spalls that went unidentified in the damage detection were visible on the textured mesh models of the girders improving the performance of the approach. 3D textured mesh models of the girders overlaid with the detected cracks and spalls were used as 3D damage maps in residual strength estimation.

Cross-sectional slices were extracted from the dense point clouds at various sections along the length of each girder. The slices were overlaid on the cross-section drawings of the girders, and the prestressing strands affected due to the corrosion damage were identified. They were reduced in cross-sectional area to account for the corrosion damage as per the recommendations of Naito, Jones, & Hodgson (2011) and were used in the calculation of the ultimate moment capacity of the girders using an approach called strain compatibility analysis. Estimated residual capacities were compared to the actual capacities of the girders found from destructive tests conducted by Al Rufaydah (2020). Comparisons are presented for the failure sections in these tests and the results were analyzed to evaluate the effectiveness of this framework. More research is to be done to determine the factors causing rupture in prestressing strands with different degrees of corrosion. This framework was found to give satisfactory estimates of the residual strength. Reduction in resources involved in current visual inspection practices and eliminating the need for physical access, make this approach worthwhile to be explored further to improve the output of each step in the proposed framework.

ACKNOWLEDGEMENTS

I would like to extend my deepest gratitude to everyone who helped and supported me throughout this research project. Firstly, I would like to thank my adviser, Dr. Roberts-Wollmann not only for this challenging opportunity that taught me a new skillset but also for her vision, expertise, and for all her support throughout the project. I would like to thank my committee members, Dr. Hebdon and Dr. Koutromanos for their guidance.

I would like to extend my gratitude to Dr. Marc Maguire, Assistant Professor at University of Nebraska-Lincoln, and Dr. Sattar Dorafshan, Assistant Professor at University of North Dakota for sharing their pixel level labeled dataset of concrete wall and bridge deck images, that added great value to our training dataset to train the Neural Network.

I would like to thank my colleagues, Abdullah Al Rufaydah and Ali Alfailakawi for their help and support in making my research successful. I would like to extend my gratitude to my friend Sandhya Bhaskar for her help in the implementation of open-source Deep Learning model for crack detection. I would like to thank the team at Advanced Research Computing (ARC) at Virginia Tech for their help with using High Performance Computing Resources. I would like to thank to my colleague, Alan Smith for the programs that he developed for region growing and slicing the point cloud, that were used in this research with slight modifications. I would like to thank to my colleague, Eric Bianchi for his interest and support throughout the project. I would like to thank my professors and colleagues in Structural Engineering and Materials program area for sharing their knowledge and supporting me throughout my time here at Virginia Tech. I would also like to thank my parents and friends. None of this would have been possible without their support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

k	Neighborhood Size around a Point
n_s	Normal of the Seed Point
n_p	Normal of each neighboring point
θ_{th}	Threshold Angle
r_{th}	Residual Threshold
α	Multiplier to standard deviation in Statistical Outlier Removal algorithm
f'_c	Compressive strength of concrete
f_{ps}	Stress in the Prestressing Steel at the nominal strength of the member
f_{se}	Effective prestress in the strands after all the losses
L_d	Development Length of the Prestressing Strands
d_b	Nominal Diameter of the Prestressing Strands
ε_{ps}	Total Strain in the Prestressing Steel
ε_1	Strain in Prestressing Steel Caused by the Effective Prestress
ε_2	Compressive Strain in Concrete f the only Force Applied was the Prestressing Force
ε_3	Strain in concrete at level of prestressing steel when section develops its nominal capacity
ε_{cu}	Ultimate Strain in Concrete
c	Depth to Neutral axis of the cross- from the extreme compression fiber
T	Tensile Force at a section
C	Compressive Force at a Section

LIST OF ABBREVIATIONS

HRBT	Hampton Roads Bridge Tunnel
FHWA	Federal Highway Administration
AASHTO	American Association of State Highway and Transportation Officials
ACI	American Concrete Institute
ASCE	American Society of Civil Engineers
2D	Two-Dimensional
3D	Three-Dimensional
LiDAR	Light Detection and Ranging
IR	Infrared
UBIT	Under-Bridge Inspection Truck
UAV	Unmanned Aerial Vehicle
GPS	Ground Positioning System
DSM	Digital State Model
SfM	Structure from Motion
PSR	Poisson Surface Reconstruction
PCA	Principal Component Analysis
SVM	Support Vector Machine
CNN	Convolutional Neural Networks
FCN	Fully Convolutional Network
TP	True Positives
TN	True Negatives
FP	False Positives
FN	False Negatives
DI	Damage Indices
IoU	Intersection over Union
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
GUI	Graphical User Interface
LRFD	Load and Resistance Factor Design
SOR	Statistical Outlier Removal
IW	Inspection Window
CUDA	Compute Unified Device Architecture
ARC	Advanced Research Computing (Virginia Tech)
HPC	High Performance Computing
C	Cracks
S	Spalls

CHAPTER 1 INTRODUCTION

1.1. Background and Motivation

The durability of infrastructure components, such as bridges, is significantly affected by long-term deterioration due to corrosion. Corrosion is a significant concern for bridges as it reduces their load-carrying capacity which can cause a great threat to passenger safety. In concrete bridges, the ingress of agents such as chlorides through pores in the concrete causes corrosion of the reinforcing steel, which causes cracking in the concrete and accelerates the process of corrosion. As per the 2017 American Society of Civil Engineers (ASCE) Report Card on America's infrastructure, almost four in ten bridges in the United States are 50 years or older (ASCE, 2017). Current bridges are designed for a design life of 50 years. As per the National Bridge Inventory (NBI) records of 2016 by Federal Highway Administration, 4790 prestressed concrete bridges out of a total of 129,362 were classified as Structurally Deficient, which means that at least one of the bridge components is in poor or worse condition (FHWA, 2016). As per the same records of 2018, among all types of bridges, the condition of 7.6% of bridges out of a total of 616,096 bridges is reported to be in a poor condition (FHWA, 2018). So, there is a need for inspection, maintenance, repair, and rehabilitation of the existing bridges.

Currently, visual inspections, Non-Destructive testing, and live load tests are conducted on existing bridges to assess their performance and get an estimate of the residual strength of damaged bridges. There are several Non-Destructive evaluation methods available such as tap tests, ultrasonic wave velocity, impact-echo, half-cell potential, ground penetration radar (GPR), and Structural Health Monitoring techniques that use sensors to assess the condition of the infrastructure (Chang, Flatau, & Liu, 2003). However, visual inspections are still an important tool in condition assessment of bridge structures. Federal Highway Administration requires bridge inspections to be conducted once every twenty-four months as per the requirements of National Bridge Inspection Standards (NBIS). These visual inspections, which usually involve walking on the bridge deck and reaching the areas underneath the deck that are difficult to reach, are labor-intensive and require a lot of resources in the form of experienced engineers, skilled labor, equipment, time, and money. Conventional visual inspections involve equipment like ladders, scaffolding, boats, climbers, rigging, etc. and vehicles like bucket trucks, Under-Bridge Inspection Truck (UBIT), scissor lifts, etc. for accessing the hard-to-reach areas. These equipment and vehicles require skilled personnel for their operation. UBITs have high initial costs and require regular maintenance (Dorafshan & Maguire, 2018). The results depend on many human-related factors. Thus, there is a need for automation in this process of condition assessment. These days, considerable research is being carried out to use artificial intelligence and Unmanned Aerial Vehicles (UAVs) to aid bridge inspections and reduce human involvement.

The overarching objective of the current research is to evaluate the feasibility of using photogrammetric methods and semi-automatic detection of visible damage on prestressed concrete bridges. The detected damage could be used to estimate the residual strength of corrosion damaged girders. The goal is to semi-automate the process of damage mapping on the bridge girders. Visual damage maps provide information about the corrosion damage that affects the flexural capacity of a bridge girder at any section. This information can be used for arriving at an estimate of the residual capacity of the bridge girders so that bridge inspectors can make an informed decision to conduct a more in-detail inspection and decide if live load testing is necessary.

Recently, there has been some research in the estimation of residual strength of concrete bridge girders from observed visual damage. Naito, Jones, & Hodgson (2011) proposed some recommendations to estimate the residual capacity of corrosion damaged prestressed concrete box beams. From visual inspections on decommissioned box beams, they observed that the level of corrosion damage on the prestressing strands is associated with the longitudinal cracking observed and that it reduces the tensile capacities of the strands. They quantified the strength reduction in strands caused by different levels of strand corrosion damage. From destructive evaluations of box beams, Naito et al. (2011) studied the probabilities of different levels of corrosion on strands with respect to the locations of longitudinal cracking. The concept behind their recommendations was that the strands under a longitudinal crack or strands exposed to the environment or even severed (due to corrosion damage) can be reduced in cross-sectional area, either partially or fully, for calculating moment strength of a girder. Strength thus calculated should be the residual capacity of the girder as the damage was already accounted for.

Current research attempts to link the semi-automated visual inspection of bridges with the residual strength estimation. It is aimed at developing a visual tool to implement the approach proposed in Naito et al. (2011). The visual tool facilitates the process of the damage mapping, identification of the damage that affects the capacity of a bridge girder at a particular section without necessitating physical access to the girders at every step. This can be achieved by mapping the damage affecting the strength at a particular section, from the semi-automatically generated damage maps on to a cross-section drawing of the bridge girder. The cross-section drawing superimposed with the locations of cracks and spalls facilitates the identification of prestressing strands that are exposed to the environment or located adjacent to longitudinal cracks. Recommendations similar to the ones by Naito et al. (2011) can be used to arrive at an estimate of the residual capacity of a bridge girder. This step can either be done manually or can be semi-automated. But it is important to have human involvement at this stage so that logical decisions can be made based on experience to avoid the estimated capacity being too conservative or over-predicted.

1.1.1. Photogrammetry and Digital State Models (DSM)

Photogrammetry involves using photographic images to extract the required information. A Digital State Model (DSM) is defined as a 3D model of an object combined with any electromagnetic radiation. The electromagnetic radiation can be either visible light, Infrared rays, or X-rays, etc. A DSM contains a 3D point cloud generated from optical images, overlaid with 2D images as a texture to the surface mesh. A 2D infrared map of the object can also be overlaid on top of a 3D point cloud instead of using 2D images (Salomon, 2016a).

1.2. Scope of the Research

The current research established a step-by-step framework to achieve the aforementioned objectives. 3D point clouds were generated from photographs of full-scale prestressed concrete bridge girders damaged due to corrosion. The photographs were captured manually using a handheld camera following a certain procedure described in Section 3.1.1.1. Deep Learning was used to train a Neural Network to identify crack-like features on 2D images of the concrete girders. A combination of open-source datasets that were labeled at semantic level (pixel level) was used to train the network. More information on this is provided in Section 3.1.2. After the crack-like features were detected and highlighted on the images used for reconstruction of the 3D point cloud, these modified images were applied as a texture to the surface mesh generated on the 3D point cloud of the bridge girder. The textured mesh showing the highlighted cracks is henceforth referred to as a digital state model of the girder. Spalling on the point cloud of a beam was detected using an approach described in Section 2.2. The cracks and spalls together constituted the points corresponding to damage on the 3D point cloud. This step created a 3D damage map of the girder with the cracks and spalls highlighted. Any damage missed from the semi-automated detection process or other types of damage, if required could be manually identified by examining the textured mesh. The 3D point cloud of the girder was scaled to real-world dimensions, facilitating measurement of distance between any two desired points.

The points corresponding to damage were clustered to classify and label them into different clusters (groups). After this step, an attempt was made to estimate some of the dimensions of the crack and spall clusters. The damage clusters (points corresponding to spall and crack regions) affecting the strength of the girder at a cross-section, were projected on to the cross-section of the point cloud, at which strength was to be calculated. This cross-section (termed as a ‘slice’ henceforth) of the point cloud was then superimposed with a cross-section drawing of the bridge girder to identify the strands affected due to the corrosion damage. Each strand affected due to the projected corrosion damage was reduced in cross-sectional area as per the recommendations of Naito et al. (2011) listed in Section 2.8. After accounting for the corrosion damage, a reduced total cross-section area of prestressing strands and the new centroid of the reduced strands were calculated. Residual flexural capacity of the girder at the section was estimated using a strain compatibility approach. In the current research, the developed framework was implemented on four full-scale prestressed concrete bridge girders obtained from actual decommissioned bridges in Virginia which were 41 to 53 years old.

Moment capacities of a box beam and an I-beam were estimated at various cross-sections along the lengths, following the proposed framework and making use of the visual tools developed. Results from the destructive tests conducted on the same two girders by Al Rufaydah (2020) were used to validate the moment capacity calculations at the sections of failure. Comparisons were made between the estimated ultimate flexural capacities and the actual capacities from the tests at the sections of failure.

1.3. Extended Vision

Ultimately, the framework proposed in this research can be extended into using Unmanned Aerial Vehicles (UAV) in bridge inspections. Cameras mounted on UAVs can be used to take images of the bridge girders, for reconstruction of 3D point clouds. This part requires a more in-depth exploration of the field of UAVs. For the purposes of this research, the images were captured manually instead of using a UAV.

1.4. Benefits of the Proposed Framework

The benefits of the proposed framework are as follows.

- Automated methods of damage detection reduce the need for physical contact in close-in bridge inspections. Physical measurements made during the inspections using calipers, measuring tapes, etc. can be avoided to a large extent. These measurements can be made manually on the digital state models using an appropriate 3D point cloud and mesh processing tool (Salomon, 2016a).
- Detecting damage at 3D level gives a perception about the depth dimension of the spalls and cracks which cannot be obtained from 2D images (Erkal & Hajjar, 2014).
- The damage detected can be represented on the digital state model (DSM), giving an idea about its location, scale, and importance from a residual-strength estimation perspective (Salomon, 2016a).
- This will help bridge inspectors to make an informed decision on what measures are to be taken. They can determine if live load tests are necessary. This can also lead to evaluating if there is a need for load restrictions or load posting on the bridges.

1.5. Other Methods for Automated Condition Assessment

Other methods to achieve an automated Condition Assessment can make use of the following technologies: Light Detection and Ranging (LiDAR), Thermal Infrared (IR) Photography, Street View Style Photography, Digital Image Correlation (DIC), etc. (Salomon, 2016a). 3D point clouds can be generated using LiDAR where an object is scanned using a laser. The equipment has to be fixed at a location to scan the object. Multiple scans may be required for generating the point cloud of the whole object. The initial cost in this technique is higher than generating 3D point clouds from optical images.

Delamination in concrete structures can be detected from Thermal Infrared images. This technology measures material radiant temperature. Damage affects heat transfer between the delaminated surface and the concrete surface underneath. The delaminated surface and sound concrete inside would have different radiant temperatures and hence, would be detected differently. Digital Image Correlation (DIC) correlates images taken at different times by tracking some points in the images. Changes between pictures taken at different times can be detected using this technology. DIC is currently used in displacement measurements. This is applicable to measurements that change more rapidly unlike corrosion damage in structures.

1.6. Organization of the Thesis

This thesis is organized into five chapters. A literature review is performed in Chapter - 2 reporting relevant background information. This includes descriptions of published literature on generating 3D point clouds from 2D images, creating textured mesh models, Computer Vision based damage detection, Convolutional Neural Networks, damage quantification, and estimation of residual capacity of the corrosion damaged bridge girders. Chapter - 3 describes in detail, the step by step methodology followed and work done to complete the framework which is outlined in Chapter-1. Chapter-4 presents results of each step in the framework and discusses about the factors affecting the performance of each step or trials that went into arriving at manual parameters determined using trial and error, in some of the algorithms used. This section shows the results of residual strength estimation of a box beam and an I-Beam at various sections along their lengths. Comparisons of the estimated moment capacities with the actual capacities from the tests conducted by Al Rufaydah (2020) on these beams are shown and discussions about these comparisons are presented to evaluate the performance of the moment capacity estimations using different approaches. Chapter-5 presents conclusions reached from this research and future work that should be done to make this framework better, more automated, and tailored for implementation in Artificial Intelligence aided bridge inspections. Two appendices are added at the end explaining the damage affecting various sections from these beams and the reduction in area of the affected strands to account for the corrosion damage.

CHAPTER 2 REVIEW OF LITERATURE

In many parts of the United States, deicing chemicals are applied on road surfaces during the winter months. These deicing salts are a source of chlorides which penetrate into the deck and the girders through the bridge joints. Ingress of elevated levels of chlorides into the concrete causes prestressing steel and reinforcing steel to corrode. As the prestressing strands continue to corrode and increase in size due to the depositing of the corrosion products, a radial compression force is imposed on the surrounding concrete. This causes longitudinal and horizontal cracking in concrete (Naito et al., 2011). Naito et al. (2011) reported that the failures of prestressed box-beam bridges that they reviewed can be attributed to corrosion of prestressing strands that were exposed and severely corroded due to the spalling of cover on the bottom flange of the beams. Thus, corrosion damage on bridge girders affects their residual capacity and poses a threat to public safety.

Visual Inspections are an important means to monitor the condition of infrastructure (condition assessment). Limitations of manual visual inspections can be addressed by automating them using computer vision based methods (Cha, Choi, Suh, Mahmoudkhani, & Büyüköztürk, 2018). Automatic Damage detection in a structure following a disaster is of interest to many researchers because this facilitates a robot to identify structural cracks inside a building before humans need to enter, ensuring their safety. Consequently, these applications require artificial intelligence aided condition assessment.

Computer vision is a technique that tries to replicate the capabilities of human vision and enable computers to identify and process objects in images and videos. Computer vision based methods can be broadly of two types: rule-based and machine learning-based methods (X. Yang et al., 2018). Rule based approaches use pre-defined features such as intensity-based thresholding or edge detectors. Researchers such as Yeum & Dyke (2015), Dorafshan, Thomas, & Maguire (2018) used image processing techniques for damage detection on images. Supervised Machine Learning based methods use labeled data to train a model that has the ability to learn and identify crack-like features in an image. These methods include Support Vector Machines, Deep Learning etc. Literature that used these methods for crack detection is reviewed in Section 2.4.1.3. A comprehensive review of the research that used computer-vision based methods for damage detection is presented in Torok, Golparvar-Fard, & Kochersberger (2014) and Yang et al. (2018).

Most literature deals with detecting damage on 2D images of structural or architectural elements. Most of the existing computer vision based methods developed for detecting damage on concrete surfaces use labeled images of walls, pavements, bridge decks etc. The images of the bridge girders in this research had a lot of variability in terms of shadows, surrounding environment etc. Hence there is a need to create a trained model that has the capability of dealing with this variability. Also, there is not much literature that deals with identifying multiple types of damage on the images of concrete bridges at pixel level. There is a need to develop a model that has capability not only to detect multiple types of damage such as cracks, spalls, exposed and severed strands etc. but also has the capability to handle the variability of the real-world images. There is a need for a comprehensive damage detection tool for prestressed concrete bridges. This research addresses some of the problems mentioned above. Neural Network was used in this research to detect cracks on the images of concrete bridge girders.

The terms ‘3D point cloud’ and a ‘textured mesh model’ are introduced in Section 2.1. The current research also took advantage of taking crack detection to the 3D level. As mentioned in section 1.4, detecting damage at the 3D level gives information about the depth perspective of the damage and its location, scale, and importance with respect to the length of the girder. It also gives a visual damage map, where the damage is highlighted on a 3D textured mesh model. Handling the damage at the 3D level also makes it easy to measure the required dimensions and quantify the damage making use of the coordinates of the 3D data points.

Currently, there is very little research that attempts to estimate the residual strength of structural elements (in their existing condition) from a visual inspection. The current research attempted to put these two pieces together to propose a framework for specifically prestressed concrete bridges that starts with making a 3D point cloud of the girder, automates the damage detection process, and arrives at residual strength. One of the major aspects of this research which makes it unique was being able to validate the proposed framework using full-scale damaged girders from actual bridges as mentioned in Section 1.2.

The literature reviewed for each step of the framework is described below in the order of steps followed in the proposed framework.

2.1. Reconstruction of 3D Textured Mesh Models from Images

A 3D point cloud represents an object as a collection of points on its surface in three dimensions. The development in digital cameras facilitated their extensive usage in day to day life. 3D point clouds of objects can be reconstructed from 2D images (captured using a camera), following an algorithm called Structure from Motion (SfM). The algorithm estimates the relative locations and orientations of the cameras used for taking the images and tries to locate the pixels present in multiple 2D images in the 3D space. Firstly, similar features between images are matched with each other. To measure the similarity between images, feature extractors based on local pixel intensity gradients are used. This step is called feature matching and extraction (Salomon, 2016a). As the information required for estimating the 3D coordinates of the points is extracted from 2D images, this algorithm involves Photogrammetry. The whole process can be implemented using a software. Consecutive images captured should have enough overlap that the photogrammetry software can use for point cloud reconstruction. This technique is less expensive when compared to the generation of a point cloud using LiDAR or laser scanning (Salomon, 2016a).

2.1.1. Structure from Motion

This algorithm finds the 3D coordinates of all the matched pixels between images and the relative camera locations and orientations from a subset of the matching (overlapping) pixels from image to image. A nonlinear optimization problem is solved to minimize the error in the estimation of the 3D coordinates. Salomon (2016a) provides a detailed explanation of the working of this algorithm. The camera's pose is needed to calculate the 3D coordinates of points and vice versa. The steps that follow are described in the subsequent paragraphs.

Firstly, a sparse point cloud is generated from SfM as explained above. This process produces 'tie-points'(matched points between images) using feature matching. Then, the points matched between images in the first step are used to find new matches and their 3D coordinates. This step generates a dense point cloud. This process involves dense matching and involves the creation of depth maps. Depth maps are a kind of greyscale (single channel) images that provide information about the distance of the 3D points from a given camera location. Points most distant to the camera point are given a color value different from the points that are nearest (I-Art 3D, 2019). Depth maps are generated between two overlapping images and the dense point cloud is generated using the depth maps.

2.1.2. Textured Mesh Models

In the next step, Poisson mesh creation is used to create a triangulated mesh model from a set of 3D points from the point cloud of the object. This is called Poisson Surface Reconstruction (PSR) (Salomon, 2016a). A texture, in this research, consists of images applied to the surface mesh on the point cloud to transfer the detail, color and realism present in the images to the 3D model. The process of applying a texture to the surface mesh, called Texture Mapping, creates a textured mesh using the images that were previously used to reconstruct the 3D point cloud. Based on the location of the previously matched pixels whose locations in the 3D coordinates are known, new pixels find their way to their correct positions on the 3D mesh. This step is important because it enables capturing the details and features that are smaller than the distance between the points in the dense point cloud. The features of interest can be highlighted in a different color on the 2D images and they can be transferred to the 3D point cloud using the texture mapping step (Salomon, 2016a).

2.1.3. Tools for 3D Point Cloud Reconstruction

A commercially available software called Agisoft Metashape (Agisoft Metashape Professional (Version 1.6.2) (Software), 2020) facilitates the reconstruction of a sparse point cloud of an object from 2D images, dense point cloud reconstruction, meshing the dense point cloud, and texture mapping. The optimal parameters to be used for these steps on Agisoft Metashape were documented by Salomon (2016b), Helsel & Salomon (2015) and Orsa, Eatherton, & Moen (2011). Instructions for capturing images and for arriving at appropriate camera parameters are provided in Agisoft Metashape manuals, Helsel & Salomon (2015) and Orsa et al. (2011). Agisoft Helpdesk (2019) is a good source of instructions for the recommended camera settings, image capture locations around the object, etc. Good lighting is required for image capture. They suggest avoiding blind-zones (regions of the object captured in less than two images) since only portions visible from at least two cameras will be included in the 3D point cloud of the object. They also recommend that a minimum overlap of 60% should be maintained between two consecutive pictures. Blurry images are to be avoided and full object need not be in each image frame. If some parts of the object are missing in one image, they can be captured on other images. The object of interest should take up the maximum area in each image (Agisoft Helpdesk, 2019).

Helsel & Salomon (2015) conducted trials on Agisoft Metashape (previously named Agisoft Photoscan) to determine the optimal camera parameters for a good point cloud using a Nikon D7100 camera with a 20 millimeter lens. Camera parameters include shutter speed, f-stop, and ISO light sensitivity. Helsel & Salomon (2015) suggested that optimal camera parameters, that would be different for different lighting conditions, be determined by trial. According to them, optimal parameters would make the vertical bar in the eyepiece of the camera, displaying the light balance, read zero. They suggested using shutter speed in the range of 1/20 to 1/100 seconds. Agisoft Helpdesk (2019) suggested using minimum possible ISO value and higher f-stop values (in the range of f/8 - f/11). Shutter speed should not be slow enough to cause a blur in an image. Camera flash should not be used for image capture in the 'Manual' mode. Agisoft Helpdesk (2019) suggested employing a camera with at least 5 Megapixel resolution. Salomon (2016a) suggested performing a camera calibration, which is a process of finding distortion parameters and camera parameters such as focal length, format size, etc. The process of camera calibration is described in Salomon (2016a). Agisoft Metashape also has a capability to estimate camera calibration parameters automatically as per Agisoft Helpdesk (2019). Image preprocessing (such as cropping, resizing, and rotating) is to be avoided because it affects the autocalibration procedure during the point cloud generation.

Markers (also called Targets) can be printed from the Tools menu in the Agisoft Metashape software window using default settings. Sample markers are shown in Figure 2.1. The software has the capability to locate these markers in the images, automatically. Trials by previous researchers at Virginia Tech showed that using markers helped the alignment process. These markers can also be used to establish a coordinate system for the point cloud if the real-world distances between the markers had been measured. New virtual markers or targets can also be defined at the points on a point cloud whose real-world coordinates are known (in a valid coordinate system), to georeference the point cloud. Georeferencing means transforming the local coordinate system of the point cloud into a global coordinate system (for example, latitude, longitude, and elevation).

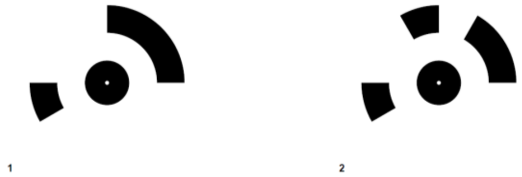


Figure 2.1. Physical Markers Used to Help the Process of Image Alignment in Agisoft Metashape Software (Source: Agisoft Metashape Professional Version 1.6.2, 2020)

2.2. Use of UAVs in Infrastructure Inspections

Unmanned Aerial Vehicles (UAVs) like drones are used in combination with a payload (equipment carried on a UAV) and a ground control system for infrastructure inspections (Dorafshan & Maguire, 2018). Dorafshan & Maguire (2018) conducted an extensive review of the usage of UAVs for inspection purposes in the United States. Typically, visual cameras are mounted on the UAVs to closely capture images or video of the bridge components. Corrosion damage is detected on the images either manually or by using computer vision based methods (real-time or post image collection). In some cases, thermal infrared cameras are also used to enable the detection of delamination. Other types of sensors used include LiDAR which can be used to reconstruct 3D surface mesh models. A UAV for an inspection can either be operated in a semi-automatic manner (where a pilot guides, controls, and navigates a UAV to follow a path) or in an automated manner. Control, stabilization, navigation, loss of GPS signals during flight (especially, under bridges) are some of the challenges in using UAVs for inspections. Some stabilization routines and Ground Positioning System (GPS) signals are used to maintain the position of a UAV. GPS signals help to determine the location of the UAVs and are used for their control and navigation along with other systems (Dorafshan & Maguire, 2018). From their review, Dorafshan & Maguire (2018) mentioned that autonomously operated UAVs mapping of the surroundings to navigate, localization (estimating its position), and path planning. Previous research involving images captured using UAVs for 3D mesh model reconstruction of buildings, bridge decks, pavements, etc. was reviewed by Dorafshan & Maguire (2018). Hallermann & Morgenthal (2014) used 38 overlapped images captured using a GPS-based autonomous UAV to reconstruct a 3D point cloud of a 590 ft long portion of a retaining wall. It was a common consensus among researchers that considerable research has to be done for implementing 3D model reconstruction from images or videos collected from full-scale bridges owing to the challenges in UAV operation, varying lighting conditions, and varying illumination of surfaces, wind disturbances, etc. They concluded that using UAVs in bridge inspections makes them safer and cheaper. Dorafshan, Thomas, & Maguire (2018b) used a UAV to capture images of a steel bridge in Idaho with fracture critical members for inspection. The inspection of the same bridge was also done using manual hands-on inspection. A review of this inspection reported in Dorafshan & Maguire (2018) indicated that the total cost of the hands-on inspection was $391 per hour (including the cost of UBIT), which took four hours to complete whereas a UAV based inspection using images took 4.5 hours to complete and cost $200 per hour, ending up costing more than the hands-on inspection. When GPS signals were lost under the bridge, the inspection had to rely on other systems and could not be completed successfully. Hence, more research has to be done to address the shortcomings of UAV based inspections and reduce the time taken to achieve cost-saving. The cost of a UAV was reported to be in the range of $500 to several thousand dollars. Pilot related costs were reported to be $600 to $1200 per day. Dorafshan, Maguire, Hoffer, & Coopmans (2017) suggested using a minimum surface illumination of 200 Lux for capturing good images using Nikon COOLPIX L830 and DJI Mavic (camera on a DJI Mavic model UAV) cameras to be able to identify cracks on the steel bridge with fracture critical members in Idaho mentioned above. Problems in UAV based inspections and future work required are elaborated in Chapter-5 of Dorafshan & Maguire (2018). Regulations put forth by Federal Aviation Administration (FAA) are to be followed for UAV based bridge inspections.

Salomon et al. (2017) used a UAV to capture images and videos of a bridge in Virginia to build 3D point clouds and textured mesh models of the bridge deck, components of the abutments and the girders close to support locations, which were later used to measure the required dimensions with an accuracy of 0.1 in. The UAV was translated incrementally in horizontal and vertical directions to scan the components of the bridge. They attempted to study the effects of local wind disturbances on UAV flight and sensing performance. They measured nominal wind speeds at the location of the bridge and localized wind speeds using wind sensors to study the effects of wind disturbances. They reported that wind disturbances affected the UAV’s proximity to the bridge and the flight speeds causing vibrations, blur in images, and errors in camera readings. GoPro camera (video recording capability with 120 frames per second with a resolution of 1280x7420 pixels) was used for capturing the video and using cameras with better resolutions was recommended for building more accurate textured mesh models. They established that 3D point clouds and textured mesh models are a great way to help visual inspections and enable extracting qualitative and quantitative measurements, eliminating the need for physical contact measurements using instruments such as crack microscopes, calipers, and wire potentiometers etc.

2.3. Damage Detection on 3D point cloud

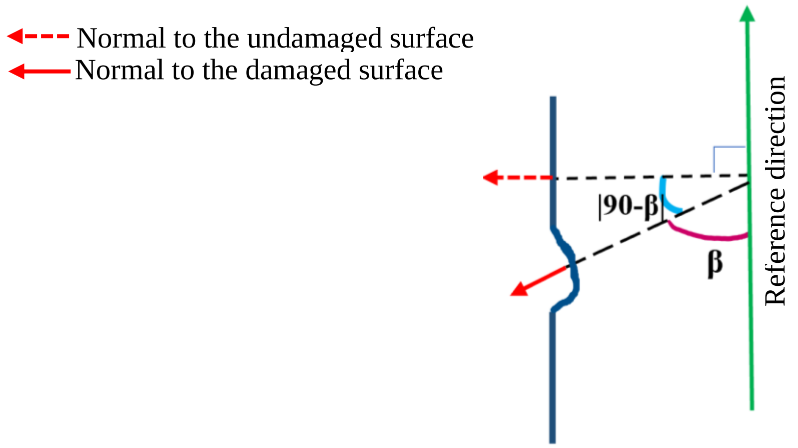
2.3.1. Surface Normals Based Approach for Damage Detection

Torok, Fard, & Kochersberger (2012) have proposed a method to detect the damage on 3D point clouds of infrastructure using normals to the surface mesh. According to Torok et al. (2014), 2D level damage detection (on images) does not provide depth information and hence, the severity of the cracks cannot be clearly understood. They pointed out that 3D level damage detection, on the other hand, provides depth information in addition to what an image-based damage detection would give.

In the surface normals approach, first a 3D point cloud of the structural object is generated using Structure from Motion method. Poisson mesh is created on the point cloud, and it is rotated so that the Y-axis is along the longitudinal axis of the object. A normal is drawn to each face of the surface mesh. Here, the total number of surface normals is equal to the number of faces in the mesh. Then, the orientation of a normal to the surface mesh, relative to the axial direction is used to identify damage on the 3D point cloud. If the normal is within a threshold angle (θ_{th}) of perpendicular to the reference direction, the element is considered undamaged (not cracked), but if it is far from perpendicular to the reference direction, the element is considered to be damaged (cracked). This threshold angle (θ_{th}) is chosen by trial and error. Among 10°, 15°and 20°, the 10° threshold angle is reported to find the largest number of cracks without too many false positives (Torok et al., 2012). This approach is demonstrated in Figure 2.2.

Erkal & Hajjar (2014) mentioned two ways of computing the reference direction with which all the point normals were to be compared. One way is to compare the point normals with the skeleton of the point cloud. That means if the point cloud is oriented longitudinally along the y-axis, then the y-axis or a unit normal in y-direction (0, 1, 0) is the reference direction. Another way to get the reference direction is to estimate the normal to an undamaged portion of the beam.

This is done by locating the points with the least curvature for a neighborhood of points. Average normal of three points with the least curvature is used as the reference normal or reference direction (Erkal & Hajjar, 2014).


Normal to the undamaged surface
Normal to the damaged surface
|90-β|
β
Reference direction

If |90-β| < = Threshold Angle (θ_{th}), surface is classified as undamaged
If |90-β| > Threshold Angle (θ_{th}), surface is classified as damaged

Figure 2.2. Normals to the Damaged and Undamaged Surfaces (Torok et al., 2014)

2.3.1.1. Alternate Approach for Generating the Normals Associated with Each point

Rusu, Marton, Blodow, Dolha, & Beetz (2008) proposed a different way of obtaining the surface normals. Instead of drawing a normal to each face of the surface mesh, they proposed estimating normal corresponding to a neighborhood of ‘k’ number of points around each point in a point cloud. A least-squares fit was used for plane fitting to those ‘k’ points and a normal to this plane was taken as the normal corresponding to the point of interest. The surface normal was determined from the eigenvectors and eigenvalues of a covariance matrix created from the nearest neighbors of the point of interest. These components were obtained by using Principle Component Analysis (PCA) on the covariance matrix, described in the Section 2.3.1.1.1. The eigenvector corresponding to the smallest eigenvalue was the approximation of the normal vector (n_x, n_y, n_z) to the plane fit to the ‘k’ points including the point of interest (Rusu et al., 2008). Then, this was repeated for all the points in the 3D point cloud. This method gave as many normals as there were points in the 3D point cloud.

A least-squares plane fitting estimation which was described in Shakarji (1998), was used to determine the normal $\vec{n}$ of the plane. As per this method, the sum of squares of perpendicular distances from each of the points to the plane had to be minimized. The plane was described by

P_x, a point on the plane and $\vec{n}$, normal to the plane. P_i is the ith data point. Perpendicular distance (d_i) from the point P_i to the plane to be fit is $d_i = (P_i - P_x).\vec{n}$. P_x and $\vec{n}$ were calculated to give $d_i = 0$ by taking the centroid of the neighboring points to lie on the plane fit. A unit surface normal $\vec{n}$ was estimated by finding eigenvalues and eigenvectors of a covariance matrix created from the data points. These components were obtained by using Principle Component Analysis (PCA). The eigenvector which corresponds to the smallest eigenvalue is the approximation of the normal to the plane, $\overrightarrow{+n}$ or $\overrightarrow{-n}$ and hence the normal to the point of interest (Rusu et al., 2008).

2.3.1.1.1. Principal Component Analysis (PCA)

Principal component analysis, in simple words, is coordinate transformation. The axes of the data are rotated such that the new X' axis is in the direction of maximum variation and the direction perpendicular to that (Y' axis) will have the least variation. Similarly, this can be extended to three-dimensional data. The transformed set of axes are related to the eigenvalues and the eigenvectors of the covariance matrix. Each eigenvector obtained is a unit vector in the direction of one of the transformed co-ordinate axes. Each eigenvalue gives a measure of the variation of the data points along the direction of the corresponding eigenvector (Pauly, Gross, & Kobbelt, 2002). The maximum eigenvalue corresponds to the axis along the direction of the highest variation (Wallisch, 2014). Normal to the data points is taken to be along the direction of the lowest variation. PCA minimizes the perpendicular distances from the data to the fitted model.

The normals to the neighborhood points calculated using this approach could be oriented in $\overrightarrow{+n}$ or $\overrightarrow{-n}$ direction and were ambiguous (Rusu et al., 2008). In this research, the orientation of the point normal in the positive or negative directions would not affect the results. For example, if the normal of an undamaged surface (1, 0, 0) was wrongly calculated as oriented in the opposite direction (-1, 0, 0), it would still be perpendicular to the reference directions, (0, 1, 0) and (0, 0, 1). So, it was assumed that the ambiguity in the positive or negative orientation of the normals does not affect the result.

2.3.1.2. Limitations of the Normals Based Approach for Damage Detection

The surface of concrete members is rough and often has small holes due to the surface finish. This method captures all the surface roughness along with cracks and spalls because all these locations have surface mesh normals that are not exactly perpendicular to the reference direction. This undesired noise contributes to false positives in the damage detected on the 3D point cloud and, hence it has to be dealt with.

From this method, the minimum detectable damage size is limited by the resolution of the point cloud (Erkal & Hajjar, 2014). If defects with sizes close to the resolution of the point cloud are to be detected, Erkal & Hajjar (2014) suggested using color or intensity information associated with the point cloud from the mapped texture. They determined a threshold to determine if that region is damaged or not. But in this research, this method was found out to be ineffective as different regions on a point cloud are illuminated differently in the real world, unlike in a controlled laboratory setting. The process used in Erkal & Hajjar (2014) in a flowchart is shown in Figure 2.3.

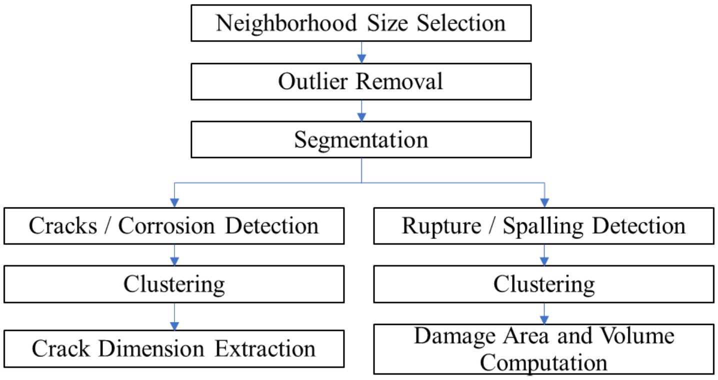

Neighborhood Size Selection
Outlier Removal
Segmentation
Cracks / Corrosion Detection
Rupture / Spalling Detection
Clustering
Clustering
Crack Dimension Extraction
Damage Area and Volume Computation

Figure 2.3. Surface Normal-Based Damage Detection Proposed by Erkal & Hajjar (2014)

Salomon (2016a) pointed out that the Normals approach cannot be automated as an independent damage detection method because it involves a very high level of noise, as the surface of the concrete is rough and has surface holes that are captured by the surface normals based approach. This is also observed in our trials. However, it does give valuable information pertaining to the damage of the surface. Also, using 3D point cloud damage detection offers an easier way to quantify the damage detected. So, in the current research, the results of this method were retained and used for damage quantification. Before proceeding further to residual strength estimation, an additional step is required to filter out the background noise generated by this approach. In the current research, 2D level damage detection on images that were used to create the 3D point cloud using a Convolutional Neural Network was explored as a possible step to reduce this background noise (Salomon, 2016a).

2.4. Damage Detection on 2D Images

Manual methods of crack detection in infrastructure inspections are time-consuming and labor-intensive. Advanced computer vision based techniques could be used to automate the process of crack detection at image level which would save many resources (X. Yang et al., 2018). 2D images can be read and written as matrices. Greyscale images are single channel images that can be represented with one matrix. Color images have three values for red, blue, and green color component values for each pixel. These images are represented using matrices with three dimensions.

2.4.1. Computer Vision Based Methods of Damage Detection

There are different techniques used in literature to detect cracks and other types of damage on pictures. Generally, the literature consists of Computer Vision and Machine learning based techniques to detect damage on asphalt pavements which was proposed by Zhang et al. (2019), Yang et al. (2018). Crack detection on bridge decks was proposed by Dorafshan et al. (2018) and crack detection on concrete walls was shown in Yang et al. (2018), etc. This kind of research is also intended to detect damage in post-earthquake scenarios.

Most of the review conducted showed the use of basically two types of Computer Vision based algorithms for the detection of damage on images of the infrastructure. These are Image Processing Techniques and Deep Learning based algorithms.

2.4.1.1. Image Processing Techniques (IPT)

Image Processing Techniques such as edge detection have been used to identify damage on 2D image along with some filters that facilitate noise removal. Image Processing techniques usually include edge detection, edge image enhancement, and segmentation or binarization (Dorafshan et al., 2018a). Dorafshan et al. (2018) used edge detection techniques such as Roberts, Prewitt, Sobel, Laplacian of Gaussian, Butterworth, and Gaussian filters. They compared the performance of these filters and concluded that Laplacian of Gaussian (LoG) filter had the best accuracy of 98% and true positive rate of 79% and true negative rate of 99%. These performance metrics are introduced in Section 2.4.1.3. The results of edge detectors have a considerable background noise, that arises from light and distortion of the images (Z. Liu, Cao, Wang, & Wang, 2019). IPTs require pre and postprocessing which can be time-consuming (Cha et al., 2018). Dorafshan et al. (2018) divided each image in their image dataset into patches (or sub-images) and used AlexNet, which is a Convolutional Neural Network to give a label of Crack or Non-Crack on each patch. They also proposed a hybrid detector, where the sub-images labeled as cracked by AlexNet are only subjected to the LoG filter to arrive at a pixel level detection of a crack. This reduced the noise that was earlier caused when only LoG filter was used for crack detection. They concluded that the hybrid method performed better than the traditional edge detectors for concrete crack detection. Convolutional Neural Networks are explored in more detail in section 2.4.1.3.

2.4.1.2. Bag of Visual Words with an SVM classifier

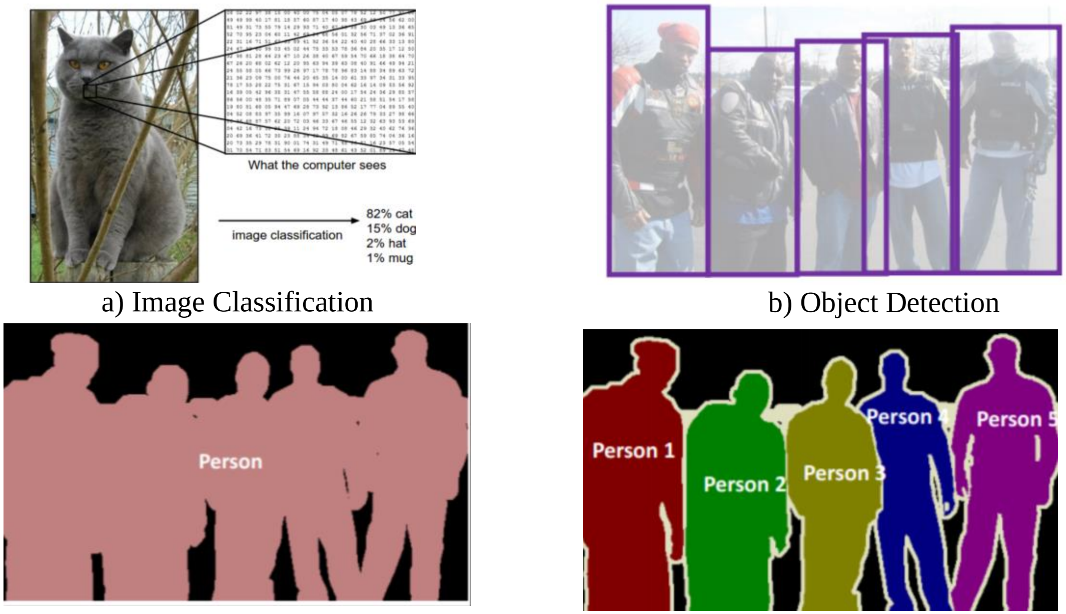
Salomon, (2016a) proposed a new framework to detect damage on 2D images. His approach starts with dividing the image into patches called Superpixels based on intensity values. Crack detection is done in two stages. In the first stage concrete vs. non-concrete detection is done and in the second stage concrete vs. cracked concrete is detected. This approach is an object-level detection, as it only puts a label on a Superpixel as concrete or cracked concrete. Further, a simple intensity-based thresholding is used to get to a pixel level crack detection inside a Superpixel labeled as cracked concrete. For detection in these two stages, Salomon (2016a) used an approach called Bag of Visual Words for feature extraction. An example of this approach is an image containing a bike would be labeled as it contains patches of wheel, frame, pedal, etc. (Salomon, 2016a). A visual word is a feature of the image such as wheel, frame, pedal etc. This approach involves the generation of a visual vocabulary which is a histogram of color visual words and texton visual words. A texton is a response of an image to a filter. Classification is done by using Support Vector Machine (SVM) classifier. At first, the algorithm learns the visual vocabulary of superpixels that are determined as cracked concrete from the labeled images provided by the user. When a new image is fed, the feature histograms of each superpixel are calculated and passed through the SVM classifier. Based on the learned knowledge, it predicts if each superpixel belongs to concrete or cracked concrete.

If a large amount of data is given, this model suffers from low accuracy. Other approaches such as convolutional neural networks show better classification results than this approach. Even then, this approach is popular due to its simplicity and computational efficiency (Richard & Gall, 2017). This approach was however not used in the current research as a CNN would be a better approach that has the capability to handle the variability present in the current dataset of concrete bridges.

2.4.1.3. Convolutional Neural Networks (CNN)

Convolutional Neural Network (CNN) is a Computer Vision based technique used for visual recognition. A CNN is made of interconnecting layers that function as a network of neurons in the human brain. They are made of input layers, hidden layers, and output layers. Neural Networks receive input and transform it through a series of hidden layers. CNNs are used to achieve the following types of tasks as described in (Shelhamer, Long, & Darrell, 2017) which are also demonstrated in Figure 2.4.

- **Image Classification:** Mapping an image to a class; This is classification of the object in an image. The label given is the class name of the object in the image.
- **Object Detection:** Drawing a bounding box around the desired object in an image. The label contains the class, position, and size of each object.
- **Semantic Segmentation:** Classification of each pixel within an image. Ground truth image contains a label for each pixel.
- **Instance Segmentation:** Pixel level classification and classification of each instance of a class separately



a) Image Classification

b) Object Detection

c) Semantic Segmentation

d) Instance Segmentation

Figure 2.4.Types of Predictions from Convolutional Neural Networks

Source for Images: (https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47 and http://cs231n.github.io/classification/ ("Class Notes for CS231n: Convolutional Neural Networks for Visual Recognition," n.d.))

In a Convolutional Neural Network, filters are used to detect features of interest in an image. A filter is a matrix of weights, that is updated in every run (epoch) during the training process. The goal of the training process is to learn the values of the weights until the updated set of weights shows the presence of the desired features in the output image called a prediction. A CNN is trained with a set of labeled images. The training images are supplied with a ground truth label which is an ideal result expected from the model. Ground truth label images contain the features of interest (for example, cracks on a concrete surface) manually highlighted by drawing a boundary around them filled with a specific color for training the network. Ground truth labels for detecting one class are usually binary, which means the background is represented in one color and the features of interest (for example, cracks) are shown in a different color. Figure 2.4c is an example of a ground truth label for semantic segmentation in which the class to be detected (person) is highlighted in a color that is different from the color of the background. The network is initiated with random weights and the class of the image is predicted and compared with the ground truth label. In an object detection CNN, a 'Score function' maps the image to class scores, and a Loss function gives a quantification of the overlap or agreement between the prediction and the ground truth labels. If the prediction from the network closely matches the ground truth label, the loss value has to be low. On the other hand, the loss value should be high if the prediction of the network is not correct. Convolutional Neural Networks will use a complex mapping to map images or image pixels to class scores. The weights should be set in such a way that the computed scores match the ground truth labels across the whole training set ("Class Notes for CS231n: Convolutional Neural Networks for Visual Recognition," n.d.).

The filters are moved over the image part by part to see if the features of interest are available in each part. In the training process, a Convolution operation is performed on each part of the image to come up with a number which gives information of the presence or absence of a feature of interest. Each pixel of the image is convolved with the set of weights (filter or kernel) to give a filtered image. In this operation, to get the value at the center pixel of part of the image in consideration, corresponding elements in the filter and the image are multiplied and added. Convolution operation is depicted by $I_f = I * f$, where I_f is the filtered image, I is the image and 'f' is the filter. The convolution operation at a pixel gives a large real number if the feature of interest is present in the pixel. For the element-wise multiplication to be possible, the filter should have the same number of channels as the input image. The output from the convolution operator between the image and the filter is added with a bias term that influences the output scores and passed through a non-linear activation function ("Class Notes for CS231n: Convolutional Neural Networks for Visual Recognition," n.d.).

Common evaluation metrics used in Semantic Segmentation or a pixel-level detection are Intersection over Union (IoU), Precision (P), Recall (R), and F1. Intersection over Union, also known as the Jaccard Index, quantifies overlap between the prediction of the network and the ground truth label. Cracked pixels are taken as positive instances. Comparing the Ground Truth labels with the predictions from the networks, pixels are divided into four types. Total Number of pixels in an image are classified as Number of True positives (TP), Number of False Positives (FP), Number of True Negatives (TN), and Number of False Negatives (FN). Precision (P) and Recall (R) are defined as follows by (Z. Liu et al., 2019):

$$P = \frac{TP}{TP + FP} \tag{2.1}$$

$$R = \frac{TP}{TP + FN} \tag{2.2}$$

F1 is defined based on the harmonic mean of precision and recall.

$$F1 = \frac{2 \times P \times R}{P + R} \tag{2.3}$$

Other evaluation metrics include accuracy, True Positive Rate and False Positive Rate which are defined by (Salomon, 2016a) as follows.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.4}$$

$$True\ Positive\ Rate = \frac{TP}{TP + FN} \tag{2.5}$$

$$False\ Positive\ Rate = \frac{FP}{TN + FP} \tag{2.6}$$

Intersection over Union (IoU) or the Jaccard index is defined by Taha & Hanbury (2015) as the number of common pixels between the prediction and the ground truth label divided by the union of the pixels present in the prediction and ground truth labels.

$$Jaccard\ Index\ or\ IoU = \frac{TP}{TP + FP + FN} \tag{2.7}$$

2.4.1.3.1. CNNs for Damage Detection on 2D Images

Convolutional Neural Networks (CNN), have the potential to overcome the limitations of image processing techniques. CNNs have a great level of learning capabilities (Leal da Silva & Lucena, 2018). Damage on 2D images of concrete surfaces is usually detected in different ways using deep learning. The first one is an image classification approach where a label of crack or non-crack is assigned to each image. Dorafshan et al. (2018) used a CNN called AlexNet-based in fully trained mode, transfer learning mode, and no training mode to predict if there is damage on the sub-images from a bigger image of a bridge deck. Their approach only put labels on the sub-images as cracked or uncracked. One can also use a sliding window based approach instead of splitting larger images into patches or sub-images.

The second approach is object level damage detection where damage is detected, and bounding boxes are drawn by the network around the damage detected. Cha et al. (2018) developed a quasi-real-time simultaneous detection of multiple types of damages namely – cracks in concrete, steel corrosion with two levels (medium and high), bolt corrosion, and steel delamination. They used a modified Faster Region-based Convolutional Neural Network (Faster R-CNN) architecture to identify the aforementioned damage types. The training dataset contained about 2,300 images labeled for these types of damage. The size of the training images was 500 × 375 pixels. They reported a mean of the average precision (AP) for the five damage types as 87.8%. It has very fast test speeds and so, was used to a real-time damage detection on videos.

If a sliding window based approach is used for crack detection on images, the accuracy of the method depends on how small the window is. But if the patches get too small, this creates an additional difficulty detecting the damage because of their small size, and accuracy drops (Z. Liu et al., 2019). So, they suggested Semantic Segmentation for crack detection.

2.4.1.3.1.1. Use of Semantic Segmentation based CNNs for Crack Detection

Another approach is Semantic Segmentation which detects cracks in an image at pixel level. Many researchers such as Yang et al. (2018), Zhang et al. (2019), Dung & Anh (2019), Bang, Park, Kim, & Kim (2019), etc. used Semantic Segmentation for cracks detection on 2D images of concrete structures.

2.4.1.3.1.2. Fully Convolutional Network (FCN) for Crack Detection

FCN has a base network that is used in the convolutional layers to perform object level crack detection. FCN converts an object level detection to pixel level detection i.e, from a class label to a semantic segmentation based prediction. This is also called Dense Prediction (X. Yang et al., 2018). A model such as VGG16, AlexNet, and ResNet, etc. can be used as the base network. In FCN, the fully connected (FC) layer in a base CNN such as AlexNet is replaced with a convolutional layer. This change is to achieve weight sparseness, weight sharing, and reducing the number of parameters in the neural network (Z. Liu et al., 2019). Up-sampling part which uses transposed convolution inverses the process of down-sampling. The encoder consists of convolutional layers of the base network for extracting crack features, and the decoder consists of deconvolutional layers for localizing the cracks in an input image (Bang et al., 2019). Upscaling facilitates enlarging the output from down-sampling to original size. The convolutional layers or down-sampling detects the cracks on the surface and the information in the up-scaling part or the deconvolutional layers have the information of the location of the crack (X. Yang et al., 2018). One of the types of Segmentation architecture for an FCN is a U-Net.

U-Net is an encoder-decoder type of network based on FCN. U-Net architecture has two almost mirrored CNNs to form the shape of the letter ‘U’. An output from a U-Net is of the same size as the input. The common application for these networks is in biomedical research. (Ronneberger, Fischer, & Brox, 2015)

Yang et al. (2018) used a Fully convolutional network (FCN) for semantic level crack detection. They used about 800 images of cracked surfaces and ground truth labels of size 224x244 for the training process. They used VGG19 for down-sampling and to perform the crack detection task. They recommended a learning rate of 10^{-4}. They reported that the accuracy, precision, recall, and F1 score of their approach are 97.96%, 81.73%, 78.97%, and 79.95% respectively. A contribution of this research was the crack skeleton based measurement of crack lengths and crack widths on 2D images.

Liu et al. (2019) used a U-Net network based on FCN. The success of U-Net for segmentation in biomedical images motivated them to use U-Net for crack detection. They also mentioned that it gave more accurate results when compared to a CNN. The training images in their dataset consisted of only 84 images of concrete walls with cracks and corresponding pixel level annotated labels. The size of the images and the labels in the training dataset was 512 x 512 pixels. Training was done for 80 epochs, which is on the higher side. They reported that the precision, recall, and F1 score of their approach are 90%, 91%, and 90%, respectively. They concluded that crack detection using U-Net had high accuracy and wide adaptability to the situations that did not appear in the training and verification process. They compared their results with those of Yang et al. (2018) and concluded that their method has higher accuracy. The test dataset for these comparisons was not the same.

2.4.1.3.1.3. Challenges in Using CNNs for Crack Detection on Concrete Bridges

The existing computer vison based crack detection techniques have low generality and are inefficient. There is a need to develop an approach that can identify and measure diverse cracks at the pixel level (X. Yang et al., 2018). In the case of Deep Learning methods, most of the open-source datasets that are available, lack variability, and are ideally captured such that they are uniformly lit and do not include any surroundings in the background. Such networks would be unable to identify diverse cracks in the dataset pertaining to the current research. Images in the current dataset of damaged prestressed concrete bridge girders include surroundings (either outside or a laboratory setting), shadows, and variable lighting conditions. Another challenge is that there are no labeled datasets available open-source that specifically contain images from prestressed concrete bridges. So, labeled datasets of concrete bridge decks, walls, pavements, etc. were used. Challenges in crack detection on a 2D image of a concrete surface include lighting and shadow changes (Z. Liu et al., 2019). Hence there is a need to collect and merge multiple datasets to bring about enough variability so that cracks are efficiently detected in a new image, with a lot of above-mentioned variabilities, that the network is not previously trained on.

2.5. 3D Damage Maps

A textured surface mesh model on a 3D point cloud contains the details and information present in the images of the object. Salomon (2016a) mentioned that the integration of the damage detected on 2D images with the digital state model creates a novel way to present the damage. He replaced the original images with the new crack segmented images by texture mapping to the surface mesh created on the 3D point cloud. He pointed out that the DSMs could be used as damage maps containing highlighted segmented areas of damage would be useful for condition assessment of structures. The location and depth of the cracks and spalls could be measured using the DSMs. Kalfarisi, Wu, & Soh (2020) used deep learning techniques to detect cracks on images of a wall of a building and used photogrammetry to construct a 3D mesh model from the images. Consecutive images were captured with about 50% overlap between each other to enable the photogrammetry software to create the 3D mesh model. They used texture mapping to transfer the cracks detected on to the 3D mesh model of the wall which they called metaphorically as a skin cover. To enable measurement of crack dimensions, they proposed to use a scaling factor (length/pixels) to convert the number of pixels to a metric measure and color coded the cracks based on their calculated widths. They concluded that using 3D models for visualization helps inspections of infrastructure. Hence, this approach was adopted in this research to highlight the cracks on the textured mesh models of the girders and transfer the colors from the crack segmented images to 3D point clouds to facilitate using these cracks point clouds for dimension measurement (using the coordinates of the points) and for the further steps described in the following sections. After highlighting cracks and spalls on the point clouds, this damage can be divided into clusters so that each crack or spall would be in a different cluster for convenience in visualization and the process that follows.

2.5.1. Clustering

Clustering is, essentially an unsupervised machine learning algorithm where there is no labeled data to learn from. Clustering was an important step in the approach proposed by Erkal & Hajjar (2017) because it allowed the separation of all damaged points in the point cloud into independent groups. Each crack or a spall could be labeled as a single cluster. Length, width, area, and volume of each cluster could be calculated as explained in section 2.6.1. They followed a hierarchical clustering approach which generates clusters of data points. To eliminate very small clusters that contain noise (false positives) like the points corresponding to the surface roughness of concrete surfaces and not actual damage, a minimum cluster size could be specified to eliminate noise. This user-defined parameter was determined by trial and error based on the scope of the defect types.

In the current research, Density Based Spatial Clustering of Applications with Noise (DBSCAN) was explored. As per this approach, clusters are the areas of high point density separated from the areas of low density. The goal is to find core samples (groups of points) of high density and find clusters by joining these regions. This approach is good for data that contains clusters of similar density. There are two user-defined parameters, 'minimum samples' and 'eps'. Higher minimum samples or lower eps indicate higher density necessary to form a cluster (scikit-learn developers, 2019). The parameter 'eps' is a distance measure that has to be determined by trial and error. If there is more noise in the dataset, it is advised to use a higher value of 'minimum samples'.

2.5.2. Statistical Outlier Removal (SOR)

This is an algorithm for cleaning a 3D point cloud by removal of noise points (outliers) around the point cloud. It has two user-defined parameters, the number of nearest neighbors ('k') considered for each point and a multiplier to standard deviation 'α'. The algorithm starts with calculating 'k' nearest neighbors to each point in the dataset. For each point in the dataset, distance from the point to each of its 'k' neighbors is calculated. Then, the mean (μ) and standard deviation (σ) of these distances are computed for each point. The mean and standard deviation are used to define a threshold distance which would be compared to the distances calculated from each point to its nearest neighbors. This threshold distance value is $\mu + (\alpha * \sigma)$. For each point in the dataset, the neighboring points that lie at a distance greater than the threshold distance, are discarded as outliers (Rusu et al., 2008). Rusu et al. (2008) also suggested considering the neighboring points with distances calculated that are less than $\mu - (\alpha * \sigma)$ also as outliers. However, it is not clear why the neighboring points that are closer than a certain distance to the point of interest, would be outliers. Hence, only the neighboring points farther than the threshold distance mentioned above are considered as outliers in this research.

2.6. Damage Quantification

The next step after highlighting the damage on 3D point clouds and classifying the damage into different clusters was getting a quantitative measure of the damage identified. In the literature reviewed, after the damage was detected, it was quantified on either the 2D images or on the 3D point clouds. Damage Quantification on 2D Images was proposed in Jahanshahi, Masri, Padgett, & Sukhatme (2013) and Yang et al. (2018). After clustering the detected damaged points, damage quantification of the clusters in 3D point clouds was discussed in Erkal & Hajjar (2014) and Erkal & Hajjar (2017).

2.6.1. Damage Quantification in 3D Point Clouds

The quantities that were of interest in a 3D point cloud with damage placed into different clusters were area and volume of the clusters. Also, the dimensions of the cracks such as crack widths, crack lengths were important (Erkal & Hajjar, 2014).

Crack Dimension extraction was described in Erkal & Hajjar (2014) as follows. First, the clusters were projected on to a plane that was fit to the undamaged surface of the object. This approach works on the projected 2D points of each cluster. A rectangle was fit around each of the 2D projected clusters by using the difference between the minimum and maximum values along a particular axis. Then, the rectangle fit to each cluster was rotated until an optimum orientation, which minimizes the area value bounded within the rectangle was achieved. Length (l_i) and width (w_i) of the rectangle fit around each of the clusters were noted. For the projected clusters that were labeled as a spall, area, and volume were calculated.

For the projected clusters labeled as cracks, crack lengths and widths were estimated. To estimate the length of a crack, the length of the bounding box around the projected crack cluster was reported as the crack length. If the crack was curved, this approach only gives the distance between the start and the end points of the projected cluster as the crack length. To estimate the crack widths, they proposed the following approach. They call this dimension the crack thickness. Each rectangle fit around a projected crack was divided into several smaller rectangles of the length of t_{max} along the length of the bigger bounding rectangle. Then, linear regression was performed to the points that fall in each of the smaller rectangles. The residual values from the regression line fit were used to estimate the crack width. The values of residual pairs or the distances from the line fit to the points on either side of the line were taken as the average crack width of each of the smaller rectangles. The largest value width value among all the smaller boxes was then reported as the computed crack width. The accuracies of these measurements were given as a percentage error for the computed length values ranges between 0 to 10% and the percentage error for computed crack thickness ranges from 5% to 34%. However, it was also mentioned that these error values can be due to the outliers in each cluster and they can be related to the quality of the point cloud.

2.6.1.1. Area and Volume Estimation for the Clusters Labeled as Spalls

Erkal & Hajjar (2017) proposed a mesh grid based method for calculating the volume of each cluster in the bottom deck portion of a bridge. First, the points of each cluster were projected on to the plane fit to the undamaged surface of the bottom deck. The projected points of the cluster were used to create a 2D meshgrid. Then, a mesh grid was formed over the defect points in the actual cluster. A surface was fit over the generated meshgrid of the actual damage points by using their Z-coordinates as elevations. Z values at the mesh grid points on the fit surface were interpolated. The volume of each defect, four points that represent the corners of each square located on the generated mesh grid for projected points, and the corresponding four points on the fitted surface were used. Each eight-point set made a cuboid, and the volume of the whole defect was the sum of all cuboids. The cuboids with height less than 0.001 in. were excluded from the spall volume calculation. For the area calculations, only four points that represent the corners of each square of the mesh grid were used. The area of all the squares on the grid was added to give an approximation of the damaged area.

2.7. Region Growing to Fit Planes to Undamaged Surfaces of the Beams

As per Rabbani, van den Heuvel, & Vosselman (2006), region growing is a segmentation algorithm to group points that fall on smooth surfaces in a 3D point cloud based on the point normals and their residuals. It groups the points that fall on smooth surface patches together. For the purposes of this research, this algorithm can be used to fit planes to the undamaged surfaces of the beams. This iterative approach starts with a seed point to initiate the surface and grows by choosing other points that satisfy certain criteria. Point normal to each of the points is estimated by plane fitting to a neighborhood of size ‘k’ around each point. Residual value from each point to the plane fit is recorded and compared to a residual threshold to prevent points of high residuals from being seed points. Seed points are compared to its neighbors to determine the angle between the normal of the seed point (n_s) to the normal of each neighboring point (n_p). If the absolute value of the dot product of the normals is greater than cosine of a threshold angle (θ_{th}), called the smoothness threshold, then the neighboring point in question can be added to the current region (Rabbani et al., 2006). This condition is written as:

$$\left|\left|n_p . n_s\right|\right| > cos(\theta_{th}) \tag{2.8}$$

This continues until no new points are added to the region. Then, the points with residuals less than the threshold (r_{th}), become potential seed points. The algorithm continues until no valid seed points are available. It was reported that if r_{th} is too small, it leads to obtaining many segments.

2.8. Residual Strength Estimation from Visually Detected Damage

Naito conducted a detailed study of failures in prestressed box-beam bridges that had occurred due to excessive corrosion of the prestressing steel and spalling of the concrete cover on the bottom flanges of the beams. They conducted visual inspections and destructive evaluation on seven decommissioned box beams and attempted to link the surface damage observed on these beams with the residual capacity. The observed damage consisted of longitudinal cracks, severe spalling on the bottom flanges, and exposed prestressing strands. They classified the corrosion damage observed on these strands into six levels of corrosion and corresponding Damage Indices (DI). These different conditions of strand damage and their corresponding Damage Indices (DI) are shown in Figure 2.5, which is taken from Naito et al. (2011). They found that the level of damage on the strands was associated with the longitudinal cracking observed. They mapped the locations of the longitudinal cracks with the corrosion levels on the strands located in various layers either directly above the longitudinal cracks or adjacent to them. From destructive evaluation, they evaluated the probabilities of corrosion associated with the locations of the strands with respect to the longitudinal cracking. On the basis of observed results, they used a probabilistic approach to estimate the strength of the box beams in their existing condition by correlating the locations of longitudinal cracks, the level of corrosion in the strands with the reduction in strength of the strands. Naito et al. (2006) conducted tension tests on individual wires separated from prestressing strands until failure. They conducted five to six tests per level of corrosion. A light level of corrosion (Damage Index=1) did not reduce the strength of a strand wire. On the other hand, strength reductions of 20.1% and 28.6% on wires were observed due to pitting (DI = 2) and heavy pitting (DI = 3) levels of damage, respectively when compared to the peak measured strength in the tensile strength tests conducted on five strands per condition type (Naito, Sause, Hodgson, Pessiki, & Desai, 2006). Naito et al. (2011), presented average wire strengths from the tests of Naito et al. (2006) for each level of corrosion as summarized in Table 2.1.

Table 2.1. Wire Strengths for Different Levels of Corrosion from Naito et al. (2006) and Naito et al. (2011)

Wire Condition	**Average Wire Strength, ksi**	**Relative Average Strength with respect to that of Light Corrosion, %**
Light Corrosion	288	100
Pitting	230	79.9
Heavy Pitting	205.6	71.4

Naito et al. (2011) put forth the following recommendations to arrive at a conservative estimate of the residual capacity of a corrosion damaged box beam. As per these recommendations, for the estimation of strength at a section, longitudinal cracking, spalling, and exposed strands observed within a region of two development lengths (with the section of interest at the middle) have to be assumed to occur at the section of interest. It is recommended to consider only 75% of the cross-section area of strands in the strength calculations if they lie in different levels, directly in-line with a longitudinal crack or if the strands closest to the surface are adjacent to a crack within a distance of 3 in. from the crack. Strands that are exposed due to spalling and have wire loss or fracture are to be discounted completely from the calculation. If the beam has longitudinal cracking, the strands that are neither reduced to 0% nor 75%, are to be reduced to 95% of their cross-sectional area in the strength calculations. Naito et al. (2006) suggested considering the pattern of the crack while determining the strand reductions. If the crack pattern on the bottom of the beam is such that, it shifts transversely from one strand to another, more strands lying on the crack inside the two development lengths region should be discounted (Naito et al., 2006). Furthermore, if the design information is unavailable, they recommended conservative estimates for the lengths of inspection windows to be used to look for damage, in place of the aforementioned two development lengths region. The section at which the strength is to be calculated is taken to be in the middle of the inspection window. They calculated moment capacities of the seven box beams using their recommendations and compared these values with the in-situ nominal capacities of the beams calculated using the observations from the forensic investigation of the strands. They determined the worst-case cross-sectional strand damage and arrived at strand strength reductions based on experimental data to be considered for in-situ moment strength calculations. They concluded that the moment capacities calculated using their recommendations were less conservative than the capacities calculated using the method used by the Pennsylvania Department of Transportation. In the current research, the term, ‘Inspection Window’ (IW) was used to refer to a distance of two development lengths of the strands with the section at which the strength is to be calculated, at the midpoint.

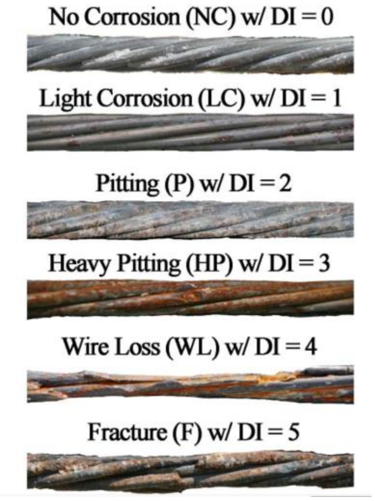

No Corrosion (NC) w/ DI = 0
Light Corrosion (LC) w/ DI = 1
Pitting (P) w/ DI = 2
Heavy Pitting (HP) w/ DI = 3
Wire Loss (WL) w/ DI = 4
Fracture (F) w/ DI = 5

Figure 2.5. Strands Damage Conditions proposed by Naito et al. (2011) (Source: Naito et al. (2011))

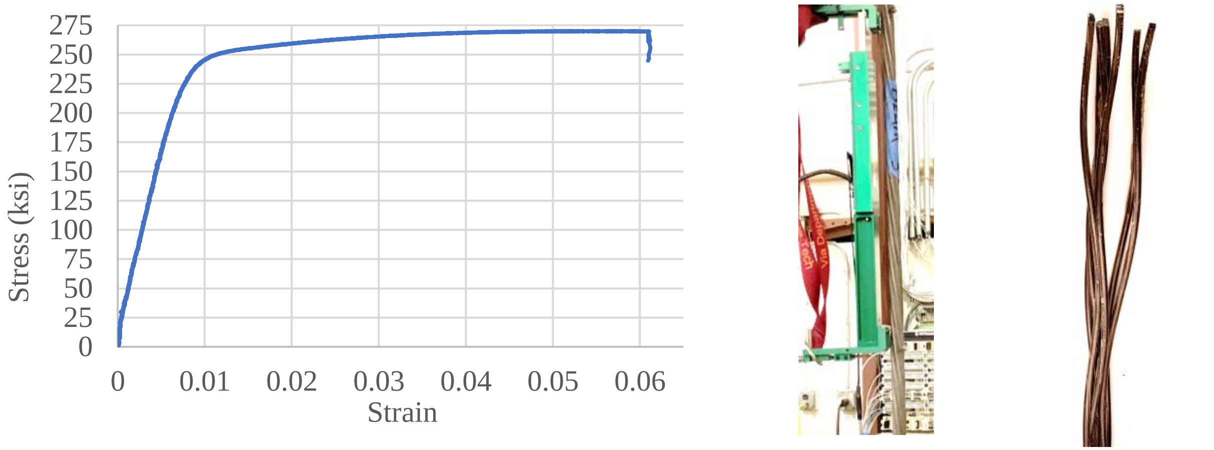
2.9. Tension Tests Conducted on Strand Samples from Corrosion-Damaged Prestressed Concrete Bridge Girders by Alfailakawi (2019)

Alfailakawi (2019) conducted destructive tests on three prestressed concrete I-Beams and three prestressed concrete box beams extracted from decommissioned bridges in Virginia, Lesner Bridge and Aden Road Bridge, respectively, to determine their ultimate flexural capacity. The beams were inspected for corrosion damage prior to testing to assess their condition, and damage maps of the beams were developed. Resistivity and half-cell potential measurements were also performed on the beams before testing. Residual capacities of the girders were estimated using the recommendations of Naito et al. (2011) in combination with strain compatibility approach and

AASHTO LRFD method. These calculated flexural capacities of the beams were compared to those from the tests. Modified recommendations to Naito's approach were proposed for calculating the residual capacity of corrosion damaged I-Beams. The modified recommendations proposed by Alfailakawi (2019) gave strengths that are about 0.97 times to 1.06 times the strengths calculated with the original recommendations of Naito et al. (2011) when used with strain compatibility approach. After destructive tests were done, samples of strands were extracted from the beams and tested to evaluate their tensile capacities. Tests were conducted using a 300 kip capacity Universal Testing Machine (UTM) and a 9 in. gage length clip-on extensometer on the strands. All these strands were stress-relieved Gr. 270 strands of 7/16 in. diameter. These strand samples were obtained from relatively less damaged sections of the beams. The results of these tensile tests are summarized in Table 2.2. The conditions of these strands are reviewed to relate them with the observed level of corrosion. The experimental stress-strain curve and photographs for the sample extracted from I-Beam 3 during testing and after testing are shown in Figure 2.6a, Figure 2.6b, and Figure 2.6c, respectively. The sample extracted from I-Beam 3 was reported to be in a good condition which can be observed in Figure 2.6b. As per the levels of strand corrosion damage put forth by Naito et al. (2011), this condition can be assumed to be with 'No Corrosion' which explains the strand reaching an ultimate strain of 6.122%. The experimental stress-strain curve of the strand sample from I-Beam3 shown in Figure 2.6a was used in this research as mentioned in Section 3.1.3.7.5 for the behavior of strands to calculate residual strength of a beam using strain compatibility approach.

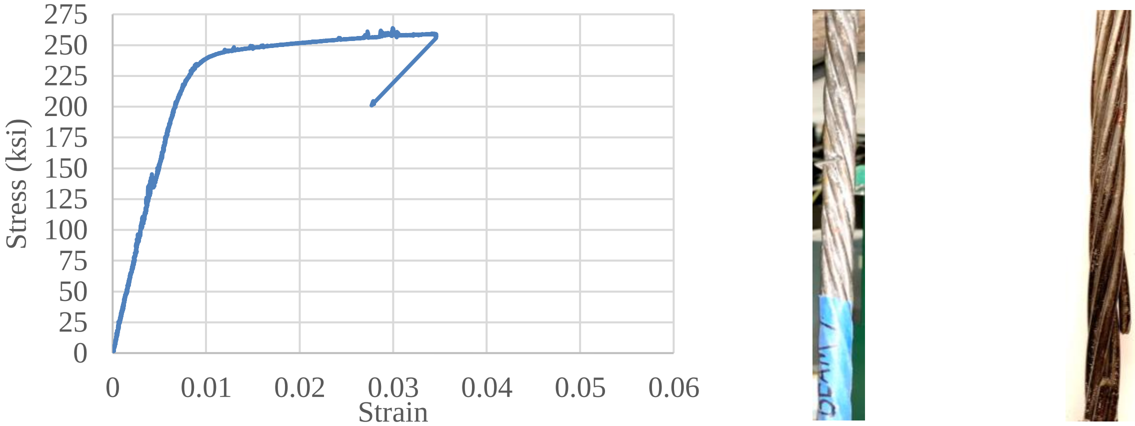
Table 2.2. Results from Tensile Tests on Strand Samples Extracted from Lesner and Aden Road Bridge Beams conducted by Alfailakawi (2019)

Girder from which the strand sample was extracted	**Elastic Modulus, ksi**	**Yield Strength, ksi**	**Ultimate Strength, ksi**	**Ultimate Strain, %**
I-Beam 3	34,000	252	272	6.122
I-Beam 7	31,000	243.5	261.5	3.462
Box Beam 1	33,000	257	269	1.675
Box Beam 2	30,000	N.A	224	0.844
I-Beam 6	27,500	N.A	188.5	0.713



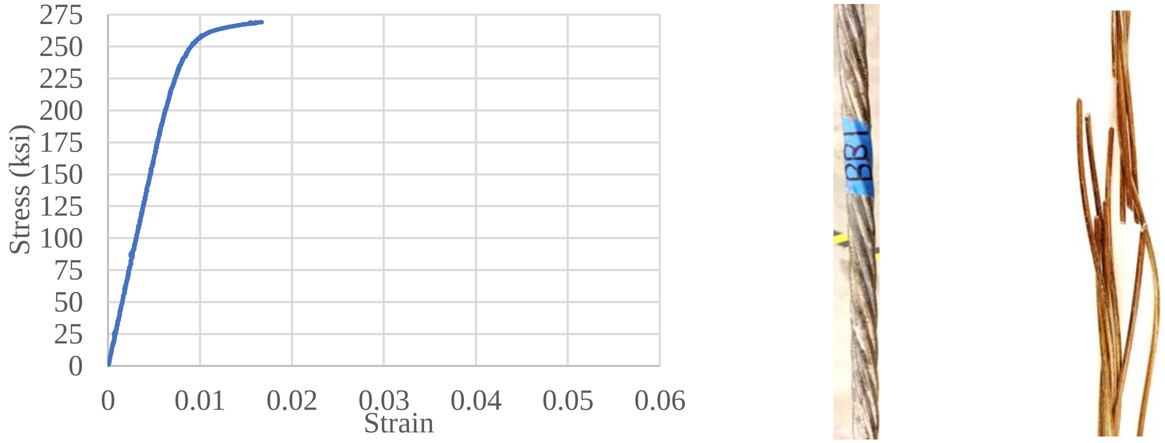
a) Experimental Stress-Strain Relationship b) During Testing c) After Testing

Figure 2.6. Experimental Stress-Strain Curve and Condition During and After Tensile Testing for Strand Sample from I-Beam 3



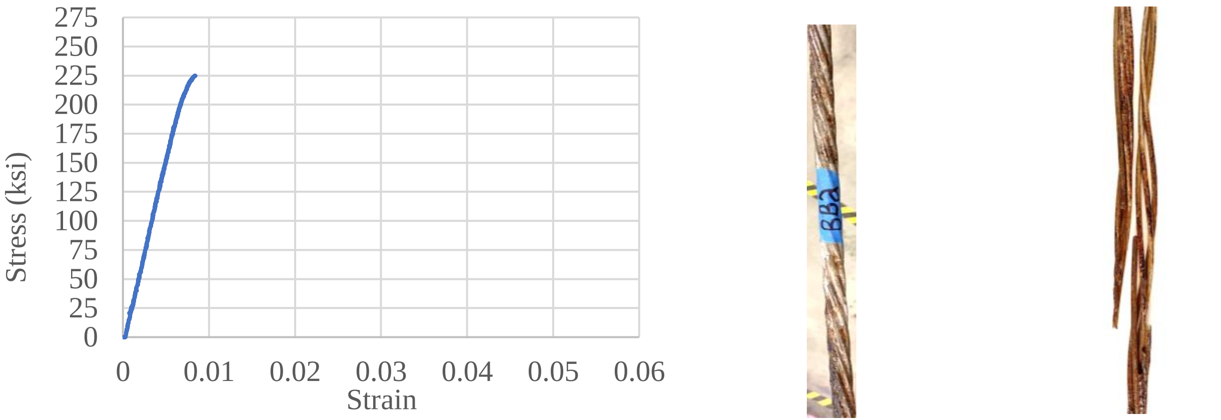
a) Experimental Stress-Strain Relationship b) Before Testing c) After Testing

Figure 2.7. Experimental Stress-Strain Curve and Condition During and After Tensile Testing for Strand Sample from I-Beam 7



a) Experimental Stress-Strain Relationship b) Before Testing c) After Testing

Figure 2.8. Experimental Stress-Strain Curve and Condition Before and After Tensile Testing for Strand Sample from Box-Beam 1

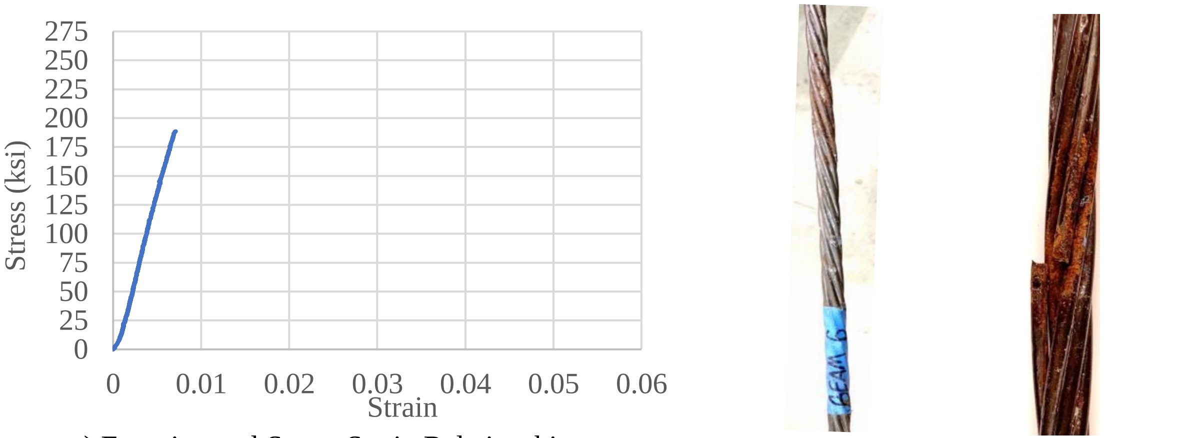


a) Experimental Stress-Strain Relationship

b) Before Testing

c) After Testing

Figure 2.9. Experimental Stress-Strain Curve and Condition Before and After Tensile Testing for Strand Sample from Box-Beam 2



a) Experimental Stress-Strain Relationship

b) Before Testing

c) After Testing

Figure 2.10. Experimental Stress-Strain Curve and Condition Before and After Tensile Testing for Strand Sample from I-Beam 6

The strand sample extracted from I-Beam 7, which can be seen in Figure 2.7b was reported to have light corrosion damage on its surface. It reached an ultimate strain of 3.462% before failure and can be associated with a strand damage condition of ‘Light Corrosion’ on the scale proposed by Naito et al. (2011). The strand samples extracted from all three box beams that were tested and I-Beam 6 were reported to have heavy corrosion damage by Alfailakawi (2019). The strand from Box Beam-1 reached an ultimate strain of 1.675% before failure whereas both the strand samples from I-Beam 6 and Box Beam 2 that showed heavy corrosion damage, failed before they reached a longitudinal strain of 1%. From inspection of the photographs, the strand from Box Beam 1 showed corrosion more than Naito’s ‘Light Corrosion’ but there was no visible pitting on the part of the strand that was photographed. Strands from Box Beam 2 and I-Beam 6 showed ‘Heavy Pitting’ type of damage, upon inspection of photographs in Figure 2.9 and Figure 2.10 and failed at ultimate strains of 0.844 % and 0.713%, respectively. The experimental stress-strain curve for the strand sample from I-Beam 4 showed a non-linear response with a small slope and failed before it reached the prescribed yield strength of Gr. 270 SR strands. So, this test was discarded as unreliable in this research.

CHAPTER 3 METHODOLOGY

This chapter discusses the methodologies implemented in the current research. The framework proposed in this research, introduced in Section 1.2, attempts to estimate the residual capacities of the bridge girders after accounting for the corrosion damage detected using photogrammetry and artificial intelligence. Corrosion damage was highlighted on the 3D digital state models of the bridge girders and used in arriving at estimates of the flexural capacities of the girders. The current chapter describes the stepwise procedure followed in implementing the framework proposed. Each step is discussed in detail. Software, programming, and other visual tools used in each step are described and the procedures followed are demonstrated.

3.1. Steps in the Proposed Framework

This research aimed to estimate residual capacities of bridge girders from semi-automatically detected visual damage on the digital state models (DSM) of the girders. The framework is discussed step by step in detail, in the sub-sections of the current section. Three main steps in this framework are digital state model reconstruction of the bridge girders, development of 3D damage maps, and evaluation of residual ultimate flexural capacities of the girders. As described in Section 1.2, a digital state model of a bridge girder contains a surface mesh on a 3D point cloud, with texture built using images on which cracks were detected. Digital state model reconstruction starts with using images of a bridge girder captured with a handheld camera, to build a 3D point cloud of the girder. Subsequent steps include detection of crack-like features on the 2D images of the girder and using them to texture the surface mesh and detection of spalls on the dense point cloud. The 3D point cloud of the girder with corrosion damage (cracks and spalls) highlighted serves a 3D damage map. The next step includes using the damage map to identify the damage that affects the moment capacity at a section, arriving at a reduced area of strands after accounting for the corrosion damage, and calculating the strength of the girder at the section using strain compatibility approach. The framework proposed in this research is presented in Figure 3.1 in the form of a flow-chart. The framework was implemented to estimate moment capacities of two bridge girders, Aden Road Bridge Box Beam-3 and Lesner Bridge I-Beam 9. Observations from the tests discussed in Section 3.1.4 on these beams were used to validate the moment capacity calculations. Each of the steps from the flow-chart is explained in the following sub-sections.

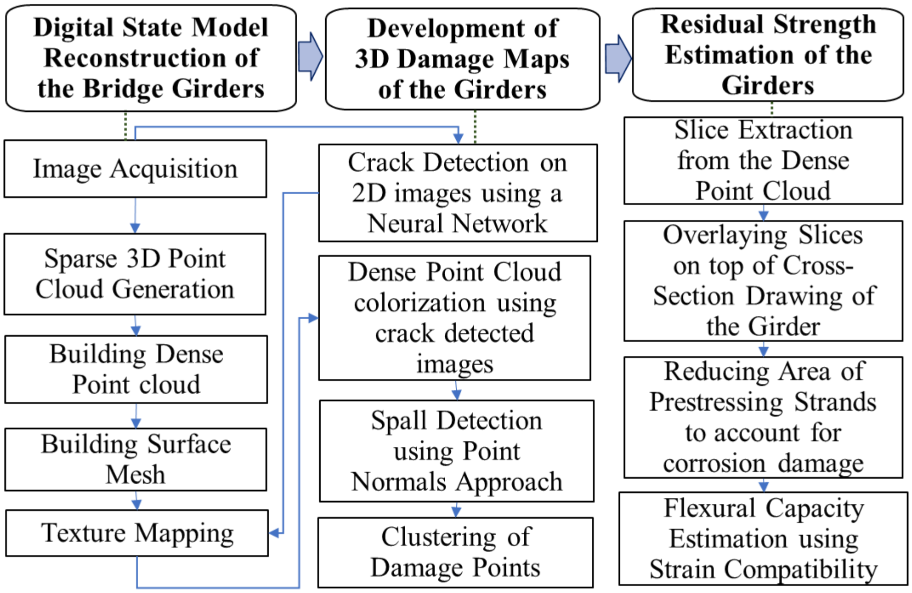

Digital State Model Reconstruction of the Bridge Girders
Development of 3D Damage Maps of the Girders
Residual Strength Estimation of the Girders
Image Acquisition
Sparse 3D Point Cloud Generation
Building Dense Point cloud
Building Surface Mesh
Texture Mapping
Crack Detection on 2D images using a Neural Network
Dense Point Cloud colorization using crack detected images
Spall Detection using Point Normals Approach
Clustering of Damage Points
Slice Extraction from the Dense Point Cloud
Overlaying Slices on top of Cross-Section Drawing of the Girder
Reducing Area of Prestressing Strands to account for corrosion damage
Flexural Capacity Estimation using Strain Compatibility

Figure 3.1. Flowchart of the Framework Proposed for Arriving at an Estimate of Residual Strength of a Bridge Girder using Photogrammetry-aided Damage Detection

The developed framework was implemented on four full-scale prestressed concrete bridge girders obtained from decommissioned bridges, Aden Road Bridge and Lesner Bridge in Virginia. Aden Road bridge was built in 1979 and was demolished in 2013. Lesner Bridge was built in 1967 and was demolished in 2017. The bridge girders used in this research are mentioned in Table 3.1. Moment capacities of Box Beam-3 from Aden Road Bridge and Lesner Bridge I-Beam 9 were evaluated at various cross-sections along the lengths of these girders, making use of the visual tools developed and following the recommendations of Naito et al. (2011). Tests conducted on these two girders by Al Rufaydah (2020) were used to validate the moment capacity calculations at the sections of failure. Comparisons were made between the capacity calculated at the sections of failure in the tests and the total moment demands from the tests. Discussions on the comparisons, efficacy of the approach, and the extent of conservatism in the moment capacities calculated following the recommendations of Naito et al. (2011) are presented in this report. In this research, Python was used for all programming needs.

Table 3.1. Full-Scale Bridge Girders from Decommissioned Bridges Used in this Research

Name of the Bridge	Location of the Bridge	Age of the bridge girders, years	Type of Girder Cross-Section	Names of girders used in this research	Purpose of the girder in the current research	Length of Girder, ft
Aden Road Bridge	Virginia Beach, VA	41	Box Beams	Box Beam-3	Estimation of moment capacity at various sections and comparison with tests	55
Lesner Bridge	Near Quantico, VA	53	I-beams	I-Beam 9		49.75
				I-Beam 1	Building 3D damage maps	50
				I-Beam 5		

3.1.1. Digital State Model (DSM) Reconstruction of the Bridge Girders

A Photogrammetry software called Agisoft Metashape was used for reconstruction of the DSM as described in Section 2.1.3. The software adopts an algorithm called Structure from Motion which is described in Section 2.1.1 for building 3D Point clouds of the bridge girders from 2D images. The 3D model reconstruction process consists of four steps. Firstly, a sparse point cloud is generated by identifying and matching similar features between images using Structure from motion approach. Then, dense point cloud is generated, and surface meshing is done. Texture mapping adds texture to the surface mesh from the 2D images.

The images of the bridge girders had to be captured using a specific procedure for the software to be able to build the point clouds. Image capture procedure, camera specifications, and parameters and other aspects of this step were found either by following the documentation available on the website of Agisoft Metashape software, the guidance of their support team or from Salomon (2016b), Helsel & Salomon (2015) or Orsa, Eatherton, & Moen (2011). The procedure followed for taking the images and aspects of image capturing that affected the quality of the point clouds are discussed in Section 3.1.1.1.

3.1.1.1. Image Acquisition

Color Images were captured using a single handheld Nikon D7100 camera with an AF Nikkor 20mm f/2.8D lens, walking around the bridge girders, and taking images at several locations along this path. The bridge girders used in this research were 50 ft to 55 ft long and 3ft to 4.5 ft deep. First, trials were conducted on smaller objects such as a broken portion of a concrete beam from one of the previous tests conducted at Virginia Tech which had a length of about 8 ft. It was used for trials with the Photogrammetry software and damage detection. Images of the girders mentioned in Table 3.1 were captured in two types of settings – indoors and outdoors. Instructions provided in Helsel & Salomon (2015) and Orsa et al. (2011) were used followed for capturing images and for arriving at appropriate camera parameters as described in Section 3.1.1.1.3. Guidelines from Agisoft Metashape manuals were also a great source to learn about the best way of capturing images. The following are some of the parameters which affected the quality of the point clouds.

3.1.1.1.1. Lighting

The surroundings of the girders had to be well lit while taking the images. It was observed that using almost the same background lighting for the images captured from each side of the object, helped to get better results. Different sides of the girders were usually found to be illuminated differently, both when capturing the images outside and inside, due to the surroundings of the girders. For this reason, when working inside the laboratory setting, LED lights were used to provide artificial lighting to reduce the effect of shadows and relatively darker regions around the girders. Care was taken so that the LED lights did not obstruct the field of view of the camera and that the portion of the girder exposed to lighting was uniformly lit. The lights were moved around the beams as the process of image capture continued. Bi-color (yellow and white light) LED lights used while taking images of the box beam inside the laboratory were of model NEEWER NL480 (Power: 28W, Lumen: 3360Lux/m) and are shown in Figure 3.2. Two lights were used to illuminate the side surface of the beam and two lights for the bottom surface. For some of the beams, images were captured in an outdoor setting. In these cases, it was ensured that there was enough light (sunny day). If the times of the day varied too much during the image capture process, the quality of the point cloud was affected by different regions having different illumination. So, it was made sure that the times of the day did not vary too much during the process. No artificial lighting was used during the image capture process in outdoor settings. The bottom surfaces of the beams were dimly lit. Due to this, it was important to choose the camera parameters that provided good illumination on the bottom surfaces. Another major difficulty while capturing images outside is the glare from the sunlight. Care had to be taken that the images from each location did not have too much glare due to the sun. When images are captured using a UAV, lighting is required for the dimly lit parts of the bridge. More work is to be done to determine the optimal lighting to capture sharp and well-lit images. As mentioned in Section 2.2, Dorafshan et al. (2017) used a minimum surface illumination of 200 Lux for capturing good images of a steel bridge to identify the cracks. They did not reconstruct a 3D mesh model from the images and used the images directly to identify the cracks manually.

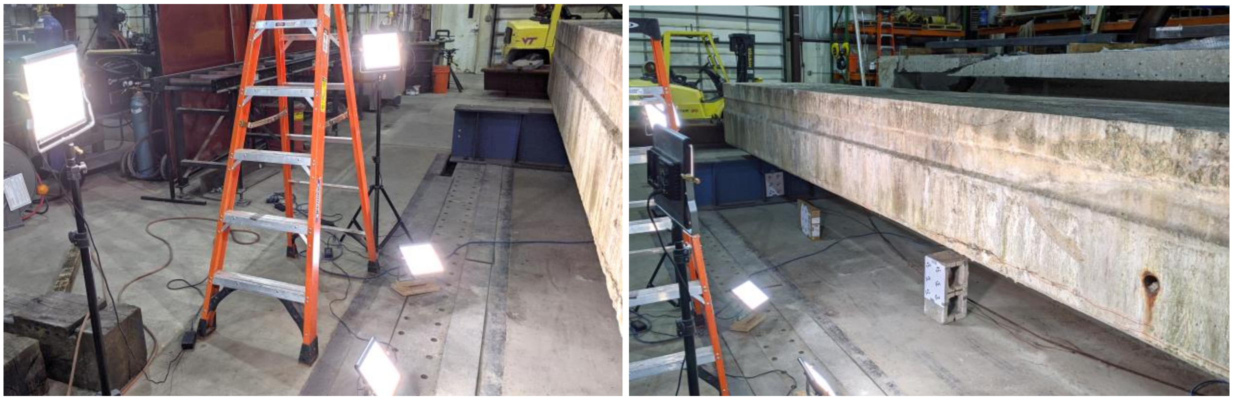
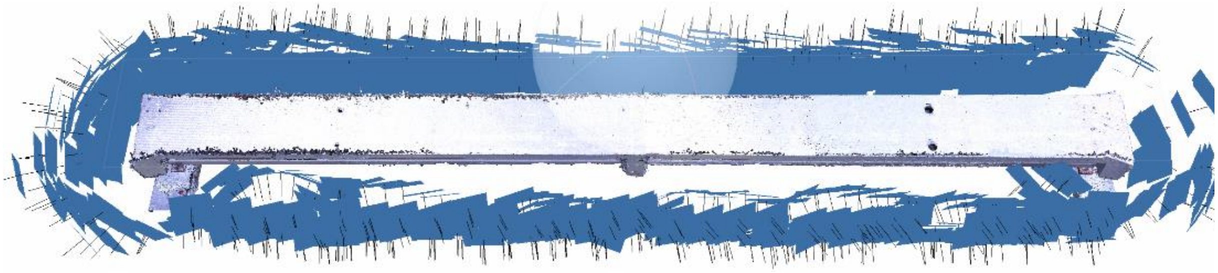


Figure 3.2. LED Lights Used to Capture Images of the Girders in an Indoor Setting

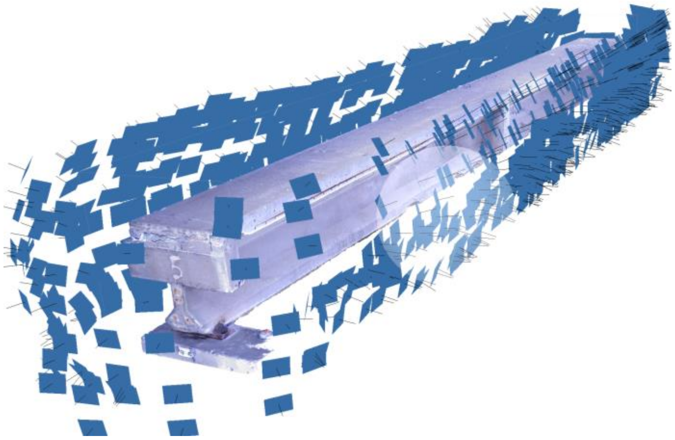
3.1.1.1.2. Image Capture Locations around the Girders

In this process, images were captured with a handheld camera at various locations by walking around the beam so that the entire beam is covered. The total number of images taken was dependent on the availability of space around the beam. The goal was to cover the entire beam incrementally. At each location, only some portion of the beam was in the frame of the camera, and images were taken at multiple levels (from top to the bottom of the girder) using a ladder, to cover the whole cross-section of the beam including the bottom surface. The bottom of the beam was captured only from a side angle with the camera held at a low height from the ground, not going all the way underneath the beam. All the image capture locations of the handheld camera for I-Beam 5 are shown in Figure 3.3. In each picture, only a part of the cross-section of the beam was visible. This can be seen in Figure 3.4 where a sequence of images taken for I-Beam 5 is shown. Care was taken such that any two consecutive images and images from two consecutive locations had about 70-75% overlap. The overlap between two images at the same location (capturing different levels along the depth) can be called side overlap and the overlap between images from each location along the length of the beam is called forward overlap. For aerial photography, Agisoft recommended a 60% of side overlap and 80% of forward overlap. This overlap requirement has to be satisfied if the image capture is done with a UAV. It was ensured such that each point on the beam was captured in at least two pictures so that the software can locate this point in three dimensions. Regions that could not be seen in more than one image, called blind-zones were avoided. Swift movement from one location to another or in between two consecutive pictures, caused the camera to be unstable which caused the pictures to be blurry. Blurry images had to be avoided.

It was ensured that the portion of the beam being photographed, covered the maximum area on each frame while taking the images. The entire cross-section of the beam was always not in the frame. Hence, for example, if the deck of the beam and only a portion of the top flange was in the first image frame, the next image was taken such that it covered a portion of the top flange and the region down maintaining an overlap of 70-75% with the previous image. A sequence of images captured from consecutive locations starting at one of the ends of I-Beam 5 is shown in Figure 3.4. Some of the intermediate images between the images shown in Figure 3.4 are not included here because it is recommended not to use excessive overlap. Although the actual images set had more intermediate images, the overlap was not large enough to cause problems in the point cloud of I-Beam 5. In other trials conducted, an excessive overlap of about 90% between consecutive images was found to be problematic, as it caused black colored noise on the point cloud. An overlap below 60% between two consecutive images is not recommended in the documentation. Optimal image overlap would result in more stable alignment and reconstruction results. Markers were placed at about 5 ft to 6 ft spacing underneath the beams on both sides. These markers can be seen underneath the box beam in Figure 3.2. In this research, markers were not used to establish a coordinate system for the point cloud as the real-world distances between the markers were not measured.



a) All Image Capture Camera Locations around I-Beam 5



b) Demonstration of Image Capture Camera Locations along the length and depth of a beam

Figure 3.3. Image Capture Locations for Lesner Bridge I-Beam 5 shown on Agisoft Metashape Software

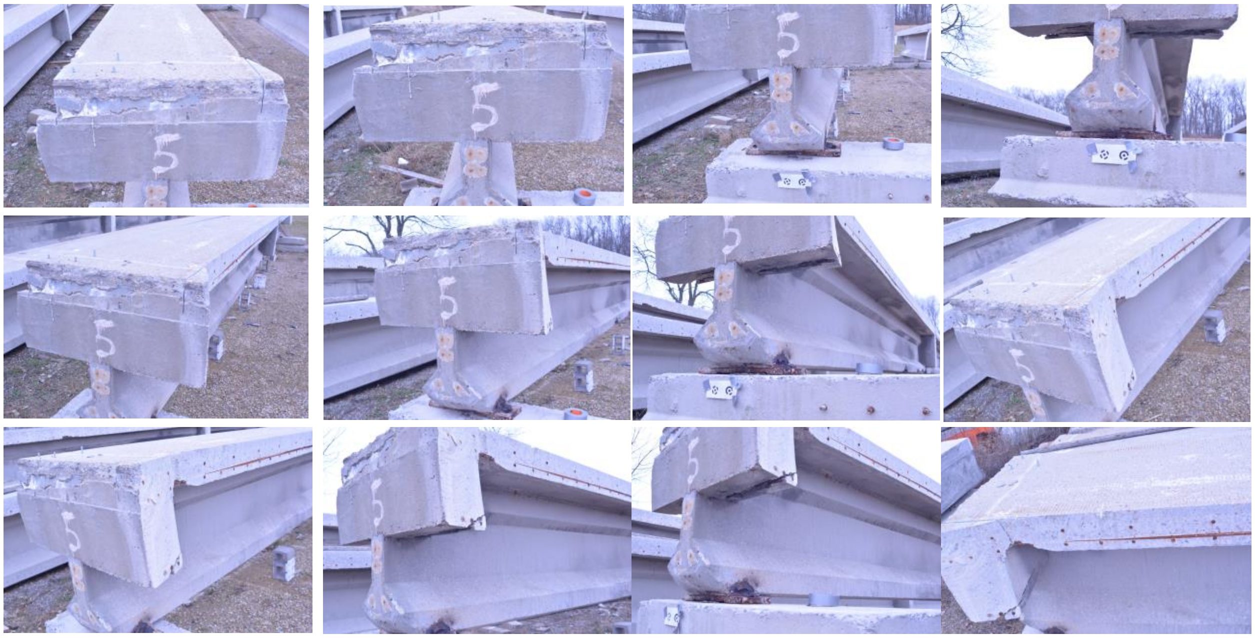


Figure 3.4. A Sequence of Images from Consecutive Locations Captured for Point Cloud Reconstruction of I-Beam 5

3.1.1.1.3. Camera Parameters

Optimal camera parameters which include f-stop, shutter speed and ISO would make the vertical bar in the eyepiece of the camera, displaying the light balance, read zero. Shutter speed in the range of 1/40 to 1/100 seconds, minimum possible ISO, and higher f-stop values (f/8 - f/11) were used. Shutter speed was varied until zero is read on the bar shown in the eyepiece. In indoor image capture trials where the light was too low, higher ISO values (about 400) were used. The parameters were selected such that the images taken from different sides of the beam were well lit. The depth of field was sometimes limited by access available around the beam. The depth of field had to be maintained almost the same when taking images from two different sides of the beam. A farther depth of field was found to be better. When taking pictures in an outdoor setting, about 7 ft depth of field from the beam was maintained. However, inside the lab because of the space constraints, the images were captured from a distance of about 5 ft away from the beam. This distance was mostly maintained during the image capture process. The images of the bottom surface of the beams were captured from locations nearer than 4 ft from the beam. The resolution of images used for all cases was 1.3x (4800 x 3200) which gave an image size of 4800 x 3200 pixels. An effort was made to ensure the camera parameters chosen did not change throughout the image acquisition process. This was sometimes difficult to maintain especially when the images were taken outdoors. Different sides of the beam had different illumination from the sun. Parameters were chosen such that the images captured from either side of the beam were not too dark. Shallow focal depth sometimes caused the images to be blurry, which was not recommended. It was ensured that the images captured were sharp and not blurry. Images were captured using the 'Manual' mode of the camera without a flash, using either manual focus or Autofocus. It was observed that the Agisoft Metashape software was successful in generating the point clouds of the beams even if Autofocus mode was used to capture the images.

Although suggested by (Salomon, 2016a), Camera Calibration, which is a process of finding distortion parameters and camera parameters such as focal length, format size, etc. was not done before capturing images of the girders because Agisoft Metashape estimates camera calibration parameters automatically (Agisoft Helpdesk, 2019). Image preprocessing using processes such as cropping, resizing, and rotating was avoided. When the images captured were too dark, brightness and contrast correction was done on the images. After the initial alignment step, the dark images were replaced with the brightness corrected images and the dense point cloud was reconstructed. The procedure followed for uploading these images to software called Agisoft Metashape and further processing of the images to build a 3D point cloud are discussed in the subsequent sections.

3.1.1.2. Alignment of Images and Sparse Point Cloud Generation

Images captured using the procedure explained above were fed to Agisoft Metashape to build a 3D point cloud. The images can be uploaded as one group or can be split into multiple groups (if there are too many images). Each group is called a ‘chunk’ in Agisoft Metashape. If only one chunk is used, all the images are processed simultaneously. This could cause the computer to run out of memory if it does not have enough computational capacity. Instead, if the images are uploaded in different chunks, it processes each chunk separately, and the chunks are aligned and merged later. To split the images of a girder into multiple chunks, pictures from each side of the beam could be uploaded in separate chunks, if required, and could be merged later. However, in this research, only one chunk was used, as it was observed that, using multiple chunks caused problems in merging. The computational capacity of the computer used in this research and the tasks that it was able to process are shown in Section 3.2. Steps to arrive at the digital state model of a girder included alignment of the images, dense point cloud generation, building surface mesh, and texture mapping. Initially, instructions for using the software put forth by Salomon (2016b) were followed for these steps. Later, the parameters that worked for the current images were found using trial and error. The procedure and the parameters used for each step in this process are explained below.

The first step in building the 3D digital state model of the girders using Agisoft Metashape is, aligning the photographs and generating a sparse point cloud as described. Parameters used for ‘Align Photos’ step on Agisoft Metashape are shown in Figure 3.5a. Time taken for this step is demonstrated in Chapter-4 Section 4.1. Sparse point cloud generated for I-Beam 5 shown in Figure 3.6a contains a large number of points from the surroundings around the object of interest. These surroundings were not needed and had to be removed to reduce the time taken for the next step, dense point cloud reconstruction. This is done by moving and resizing the ‘region’ of the object on Agisoft Metashape, which meant specifying a bounding box around the object in the sparse cloud. The software built a dense cloud for the points lying only inside the bounding box, and this was found to reduce the time taken for dense point cloud step drastically. Bounding box specified around the sparse point cloud of I-Beam 5 is shown in Figure 3.6b.

3.1.1.3. Building Dense Point Cloud

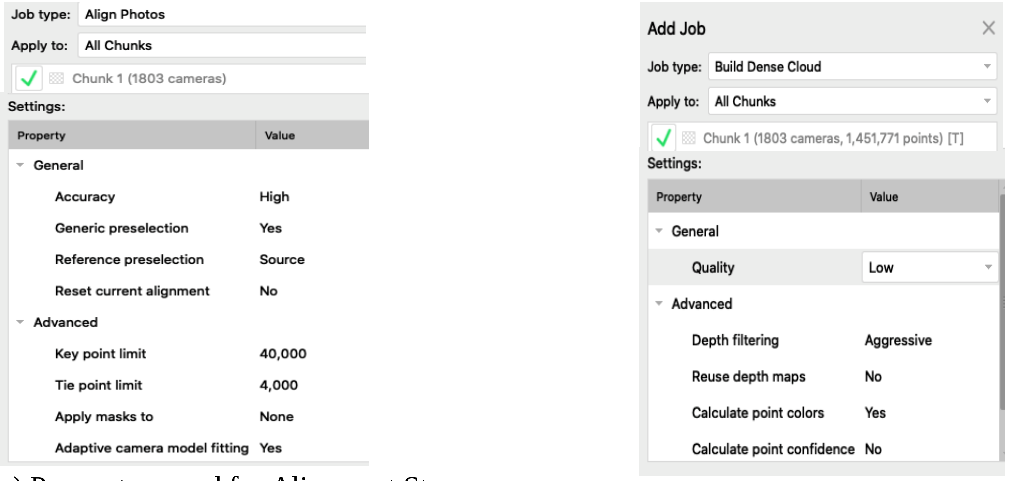
In the next step, a dense point cloud was reconstructed from the sparse point cloud. Agisoft Metashape offers five different levels of a quality parameter for this step. An ‘ultra-high’ quality makes use of the original images for generating the dense point cloud and all the other lower qualities use images downscaled by different factors. A downscaling factor of two for ‘high’ quality dense cloud makes the resolution of the dense cloud to be two times lower than the original images. For the purpose of this research, to reduce the time taken for dense cloud reconstruction and the time taken for processing in the further steps, a ‘low’ quality parameter was used. It reduced the image resolution by eight times. Parameters used for dense cloud reconstruction in this research are shown in Figure 3.5b. A low quality parameter generated a dense cloud of much smaller size (a total number of points that is about 15 times smaller) when compared to the high quality parameter. However, for the purposes of this research, a low quality parameter was found to give satisfactory results with the dense point cloud of each girder consisting of 15 million to 28 million points. The dense point cloud of I-Beam 5, that has about 15 million points is shown in Figure 3.6c. Further details about the time taken for this step depending on the quality parameter are shown in Section 4.1. Instead of using such a high downscaling factor by using the low quality parameter, a camera with a lower resolution could have been used or fewer images could have been captured for making the point cloud, by providing less overlap (not less than 60%) between consecutive images or images captured from consecutive locations.

3.1.1.4. Building Surface Mesh

In this step, a surface mesh was generated on the dense point cloud. The parameters used for this step are shown in Figure 3.5c. This step joins some of the vertices in the dense point cloud with triangular mesh and creates faces. The face count parameter was set as ‘high’.

3.1.1.5. Texture Mapping

After a surface mesh is generated on the dense point cloud, texture is applied to it using images of the bridge girder with cracks highlighted. Crack-like features were detected on the images which were earlier used for generating the dense point cloud of the girder, using artificial intelligence. A U-Net type network was trained on semantic-level labeled images of cracked surfaces as explained in Section 2.4.1.3.1.2. The process of training the network and using the trained model to detect crack-like features on the images of the bridge girders are explained in Section 3.1.2.1. The original images (called cameras in Agisoft Metashape), were replaced with the new crack-detected images. The parameters used for the Texture mapping step are shown in Figure 3.5d. Two main parameters used in this step were texture size and texture count. Trials were conducted to determine optimal values of these parameters that give a texture that was not blurry. The results of these trials are summarized in Section 4.1.1.2. A texture count of 1 and a texture size of 12288 were found to give a satisfactory result. These parameters corresponded to the size and number of texture atlas pages generated. For example, a texture size of 12288 consisted of an atlas page (image) of size 12288 x 12288 pixels with several smaller images of portions of the girders combined to form the bigger image. When the textured mesh was exported from Agisoft Metashape in the further steps, a texture size of 12288 generated only one texture image of 12288 x 12288 pixels. These models access the information on the atlas pages using texture coordinates. The textured model, which is referred to as a digital state model (DSM) in this report, for I-Beam 5 is shown in Figure 3.6d. After this step, the dense point cloud and the textured model of the girder were exported from Agisoft Metashape as explained in Section 3.1.2.2.



Job type: Align Photos
Apply to: All Chunks
✓ Chunk 1 (1803 cameras)
Settings:

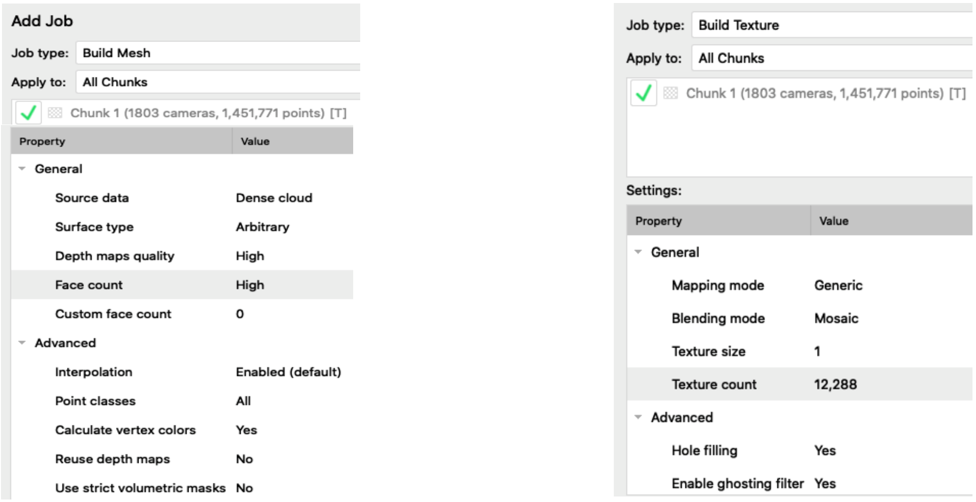
Property	Value
General	
Accuracy	High
Generic preselection	Yes
Reference preselection	Source
Reset current alignment	No
Advanced	
Key point limit	40,000
Tie point limit	4,000
Apply masks to	None
Adaptive camera model fitting	Yes

a) Parameters used for Alignment Step

Add Job
Job type: Build Dense Cloud
Apply to: All Chunks
✓ Chunk 1 (1803 cameras, 1,451,771 points) [T]
Settings:

Property	Value
General	
Quality	Low
Advanced	
Depth filtering	Aggressive
Reuse depth maps	No
Calculate point colors	Yes
Calculate point confidence	No

b) Parameters used for Building Dense Point Cloud Step



Add Job

Job type: Build Mesh

Apply to: All Chunks

✓ Chunk 1 (1803 cameras, 1,451,771 points) [T]

Property	Value
General	
Source data	Dense cloud
Surface type	Arbitrary
Depth maps quality	High
Face count	High
Custom face count	0
Advanced	
Interpolation	Enabled (default)
Point classes	All
Calculate vertex colors	Yes
Reuse depth maps	No
Use strict volumetric masks	No

Job type: Build Texture

Apply to: All Chunks

✓ Chunk 1 (1803 cameras, 1,451,771 points) [T]

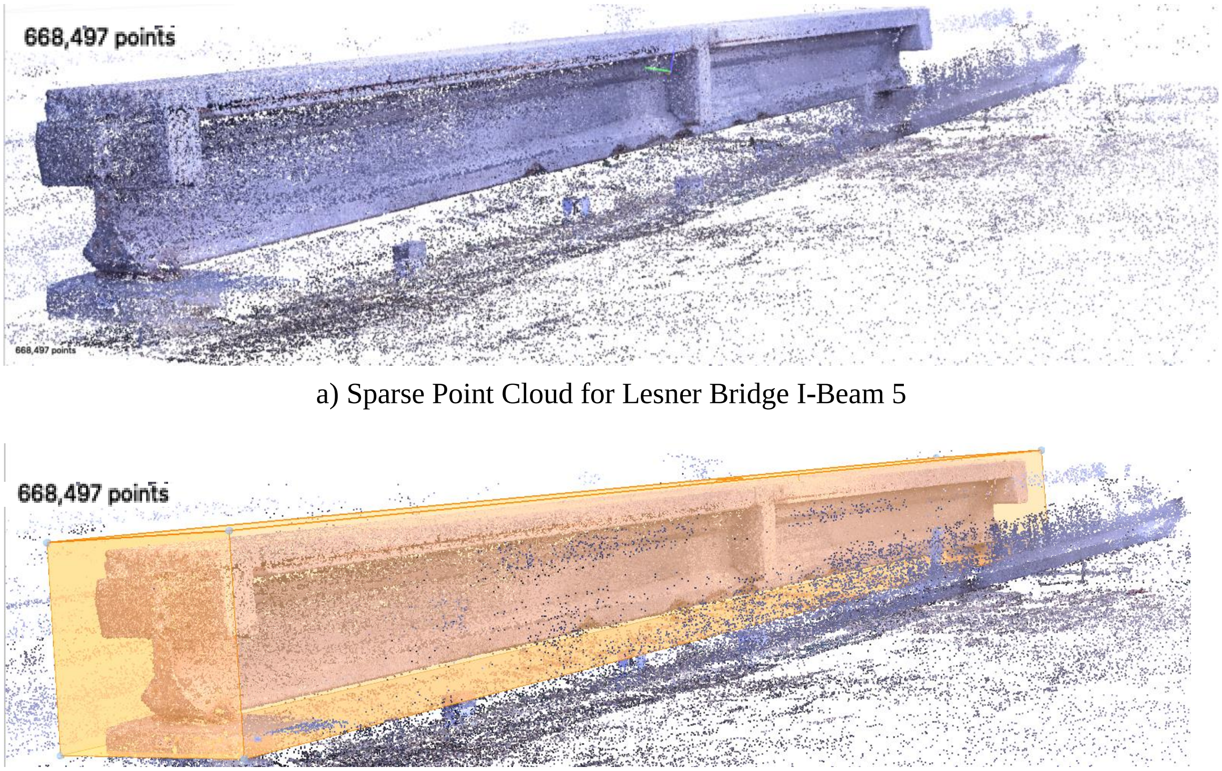
Settings:

Property	Value
General	
Mapping mode	Generic
Blending mode	Mosaic
Texture size	1
Texture count	12,288
Advanced	
Hole filling	Yes
Enable ghosting filter	Yes

c) Parameters used for Building Mesh Step

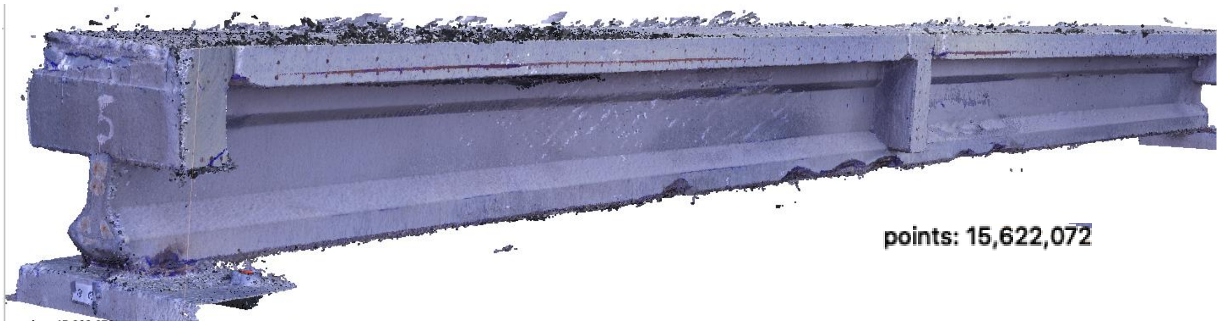
d) Parameters used for Texture Mapping Step

Figure 3.5. Parameters Used in Agisoft Metashape for Each Step for Digital State Model Reconstruction of a Bridge Girder

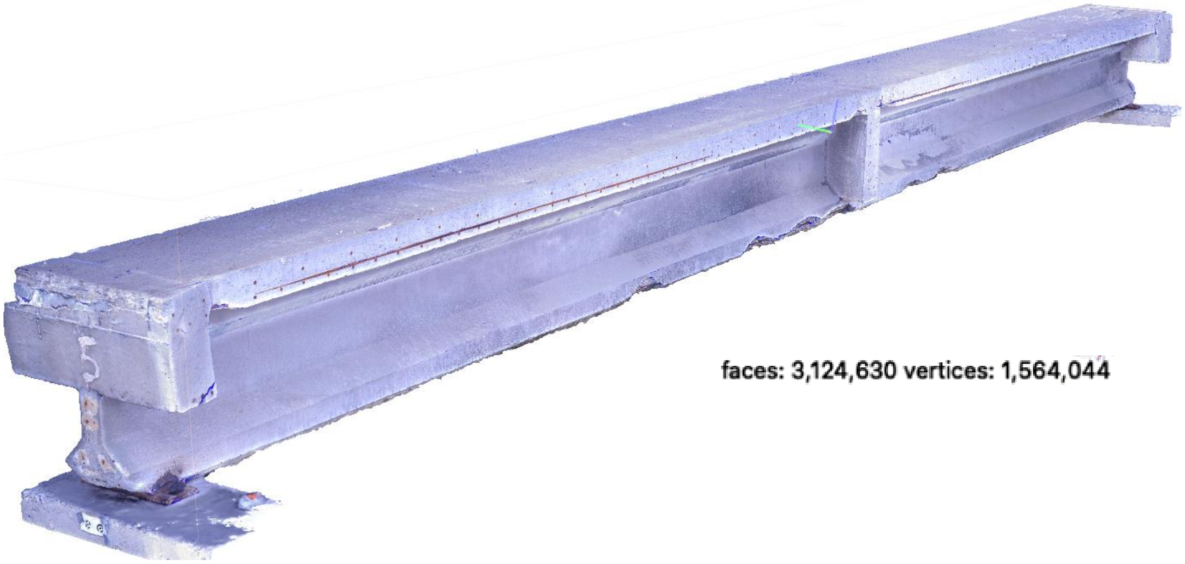


a) Sparse Point Cloud for Lesner Bridge I-Beam 5

b) Specifying the Region for Sparse Point Cloud for I-Beam 5 by Enclosing in a Bounding Box before Building Dense Point Cloud



c) Dense Point Cloud for Lesner Bridge I-Beam 5 generated using a ‘Low Quality’ Parameter



d) Digital State Model (DSM) for I-Beam 5 texture mapped using Texture Count of 1 and Texture Size of 12288

Figure 3.6. Sparse Point Cloud, Dense Point Cloud and Textured Mesh Model of Lesner Bridge I-Beam 5

Before the texture mapping step, cracks were detected on the images of concrete girders, This process is discussed in Section 3.1.2.1, as it is a part of the detection of the corrosion damage on the bridge girders and generating the damage maps. Transferring the cracks detected on 2D images on to the 3D model by texturing and detection of spalls on the 3D dense point clouds generated 3D damage maps of the girders with the corrosion damage (cracks and spalls) highlighted.

3.1.2. Development of 3D Damage Maps of the Girders

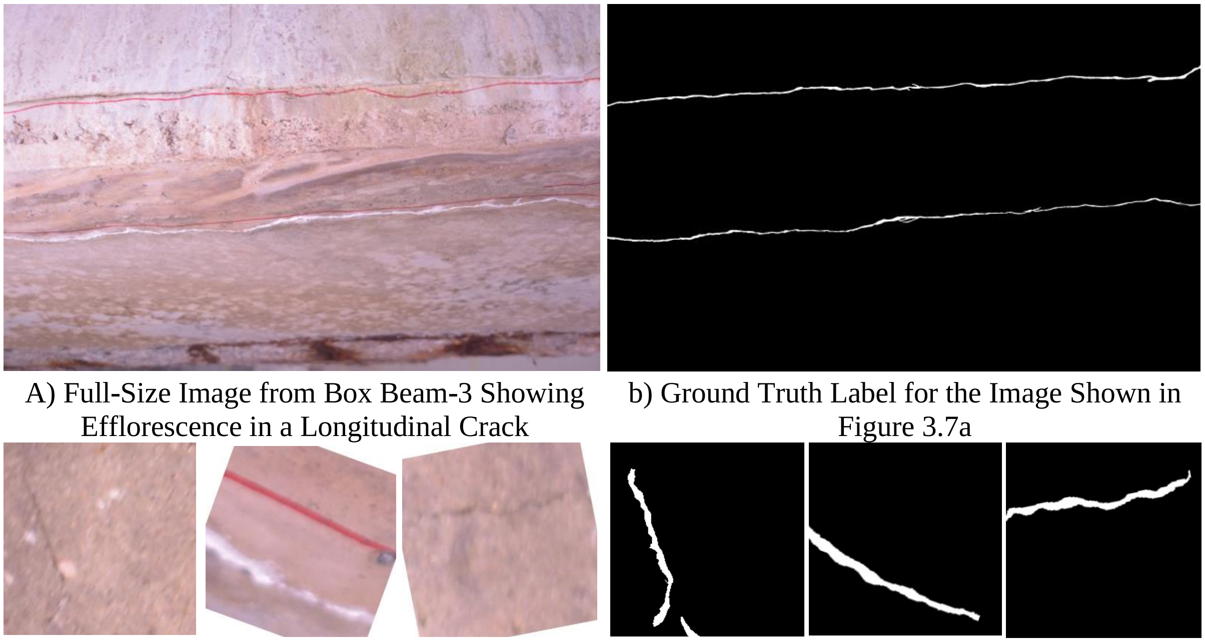
3.1.2.1. Damage Detection on 2D Images

In this research, cracks were detected on 2D images by a Neural Network trained using deep learning. A pixel level detection (semantic segmentation described in Section 2.4.1.3) was desired in order to highlight all the pixels corresponding to crack-like features on the images of the concrete girders and transfer this detected damage to the 3D point clouds as texture to the previously created surface meshes. As described in Section 2.4.1.3.1.2, a U-Net type network was trained on a dataset of labeled images of cracked and uncracked surfaces.

3.1.2.1.1. Dataset

The images on which crack detection had to be done, comprised of large variability owing to the surroundings of the girders, lighting, shadows, etc. The training dataset had to include considerable amount of variability to get a fairly good performance. The current research used a combined dataset that consisted of sub-images (small image patches extracted from larger images) and corresponding ground truth labels of cracked and uncracked concrete surfaces. The combined dataset was assembled from multiple open source datasets available online, prepared by various researchers. The dataset included images mostly of concrete and asphalt pavement surfaces. Other images included surfaces of concrete bridge decks and wall surfaces with and without cracks. The variety of the surfaces, lighting conditions, and colors in these images brought about some robustness to the model. The original datasets that had varying image sizes were collected from L. Zhang, Yang, Zhang, & Zhu (2016), F. Yang et al. (2019), Eisenbach et al. (2017), Shi, Cui, Qi, Meng, & Chen (2016), Zou, Cao, Li, Mao, & Wang (2012), Y. Liu, Yao, Lu, Xie, & Li (2019), L. Yang et al. (2017), Ç. F. Özgenel (2019), F. Özgenel & Gönenç Sorguç (2018), X. Yang et al. (2018), Maguire, Dorafshan, & Thomas (2018) and Dorafshan et al. (2018). All the images collected from these various datasets were resized to a size of 256 x 256 pixels. The images from these datasets had commonly observed cracks that showed distinct darker colors relative to the surrounding uncracked concrete. As a contrast, efflorescence deposited on the longitudinal cracks on Box beam-3 used in this research altered the appearance of the cracks to a white color which was quite different from the cracks on the assembled dataset. This can be seen on the longitudinal cracks on the bottom and side surfaces of the Box Beam-3 shown in Figure 3.7a and Figure 3.7c.

Other cracks on the box beam were narrower or did not have a distinct dark-colored opening. To include this type of cracks in the damage detected, six images of size 4800 x 3200 pixels, from the box beam images set were added to the combined dataset. Ground truth labels (at pixel-level) for these six images were prepared using a tool called GIMP (The GIMP Development Team, n.d.), by drawing around the boundaries of cracks on the images. The labels were greyscale images that had cracks in white color and the uncracked concrete in black color. An image showing longitudinal cracking on the Box Beam-3 and the corresponding ground truth labels are shown in Figure 3.7a and Figure 3.7b, respectively. The images and the corresponding labels were split into sub-images of size 256 x 256 pixels. Data augmentation is done on the sub-images by applying transformations such as flipping vertically, flipping horizontally, and rotating by an arbitrary angle. This was done to increase the size of the training and validation datasets. The images rotated or flipped for data augmentation are shown in Figure 3.7c and their ground truth labels are shown in Figure 3.7d The sub-images from the box beam images were added to the combined dataset, making the size of the entire dataset to have about 22,000 sub-images of size 256 x 256 pixels and their corresponding ground truth labels.



A) Full-Size Image from Box Beam-3 Showing Efflorescence in a Longitudinal Crack

b) Ground Truth Label for the Image Shown in Figure 3.7a

c) Box Beam -3 Sub-Images (256 x 256 pixels) Generated for Data Augmentation

d) Ground Truth Labels for the Sub-Images Shown in Figure 3.7c

Figure 3.7. Sample Images, Sub-Images and Ground Truth Labels from the Box Beam-3 Images Added to the Assembled Dataset

3.1.2.1.2. Training and Testing

The entire dataset was divided into a training dataset with 85% of the total images and validation dataset with 15% of total images. The training dataset was used to train a Fully Connected Network (FCN) with a U-Net architecture described in Section 2.4.1.3.1.2. VGG16 model was chosen as a base for the U-Net. The network was fully trained from scratch on the training dataset. All weights were determined by iterations using on the training dataset, and the validation dataset was used to evaluate this model using unseen images during the training process. As explained in Section 2.4.1.3, the goal of training is to minimize the loss function. The loss function was calculated for both training and validation datasets during each epoch. The network was trained for 10 epochs, and the model was saved at the end of each epoch. Based on the variation of the training and validation losses with the number of epochs, it was ensured that the model at this number of epochs was still not subject to overfitting (model starting to memorize the training dataset and performing poorly on new images). The model (saved at the end of each epoch) that gave minimum loss for the validation dataset was chosen as the appropriate model for testing. Other parameters used during this process were initial learning rate of 0.001, momentum of 0.9, print frequency of 20, weight decay of 0.0001, batch size of 4, and number of workers of 4. The trained model was used on testing the images set of each girder. This phase was called the testing phase.

A sliding window approach was used for testing larger (4800 x 3200 pixels) images. The program written in Python had Nvidia GPU (CUDA) dependency for both training and testing to speed up the processes. The computation capabilities of the machines used for training and testing processes are shown in Section 3.2. A stride ratio of 1 was used to feed the input images patch wise (patch size of 256 x 256 pixels) to the testing program. Two parameters in the testing phase were stride ratio for feeding the images and a threshold parameter to cut off crack response. First, the trained model was used to get predictions for each patch, and a probability map of cracks was constructed for each whole image. On the normalized probability map based on the maximum value on the map, the pixels with intensity less than the threshold parameter cut off as non-cracks. A threshold value of about 0.3 to 0.4 was found to give fairly good crack detection without giving too many false positives. Other than the six labeled images of Box Beam-3, other images from the Box Beam-3 set and images from the other beams were not used for training the network. The model showed satisfactory results for crack detection on the new images it was not trained on before. The qualitative performance of the model for crack detection on the images of the bridge girders is discussed in Section 4.2.1.

3.1.2.2. Exporting the Dense Point Cloud and the Textured Model from Agisoft Metashape

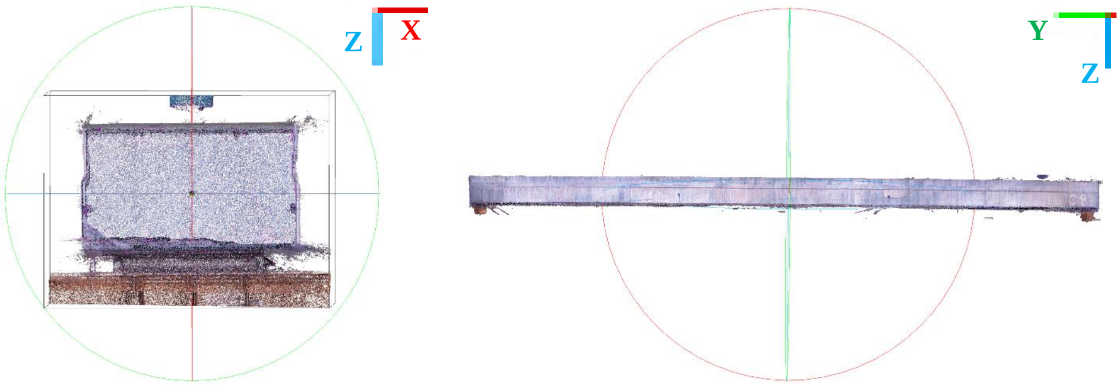
After the crack-like features were highlighted on the 2D images of the concrete girders, they were used to texture the surface meshes previously generated from dense point clouds as explained in Section 3.1.1.5. At this step in the process, digital state models of the girders included highlighted cracks. After this step, colors on the crack detected images were transferred to the dense point cloud, which made cracks highlighted in blue color on the dense point cloud as well. The dense point cloud and the textured model were both exported from Agisoft Metashape. The dense point cloud of the girder was exported in a '.ply' format and the textured model was exported using a '.obj' format. The next step deals with the detection of spall damage on the 3D dense point cloud (the exported '.ply' file) of the bridge girder. In the '.ply' dense point cloud file, colors associated with points in the dense cloud are also saved in the form of R, G, B values along with the X, Y, Z coordinates of the points.

3.1.2.3. Rotating and Scaling the Dense Point Cloud and the Textured Model to the Real-World Dimensions

In the previous step, the dense point cloud and the textured mesh were exported from Agisoft Metashape. The following steps were carried out using a 3D point cloud and mesh processing software, available open-source, called CloudCompare. As mentioned in Section 1.2, one of the goals of the current research was to establish a visual tool to assess damage associated with each section of a bridge girder at which strength has to be estimated. Using a digital state model as a visual tool, facilitates the realization of the objectives of visual inspection on bridges, without the need for physical access to the actual bridges which can be difficult and resource intensive. CloudCompare is a promising tool for this purpose which offers features such as aligning the point cloud with the X, Y, Z axes, scaling it to real-world dimensions, making distance measurements on the point cloud between any two points, etc. Other features used in this research are described in the corresponding sections of the thesis. In this step, the dense point clouds and the textured mesh models of the girders were aligned with the coordinate axes and scaled to real-world dimensions to enable measurement of required dimensions.

3.1.2.3.1. Aligning the Dense Point Cloud with the Co-ordinate Axes

The coordinate system for the dense point clouds was the default coordinate system set by Agisoft Metashape. This point cloud was not aligned with any of the coordinate axes. The dense cloud exported in '.ply' format was opened in CloudCompare and aligning with the coordinate axes was done manually. In this step, online documentation available for using the software was used to learn about the Graphic User Interface (GUI) of the software. The window of CloudCompare's GUI where mesh processing is done is called a 3D view. CloudCompare offers three different ways of applying a transformation to rotate a point cloud. In the GUI of the software, the X, Y, Z axes directions are shown at the bottom right of the window in red, green, and blue colors. In this step, alignment was done using an axis and angle approach. The axis about which the point cloud had to be rotated was specified along with the angle by which it had to be rotated. Similarly, there was an option to specify the translation if required. The point cloud was rotated about all three axes, one by one using trial and error approach by eyeballing until the longitudinal direction of the point cloud is aligned with the Y-axis, the depth direction of the cross-section is aligned with the Z-axis and width direction is aligned with the X-axis. The rotation symbol (displayed a 3D cross-hair) on the screen helped in eyeballing the orientation of the point cloud. The dense point cloud of Box Beam-3 aligned with the coordinate axes is shown in Figure 3.8. For convenience, the labels of the axes are shown in this figure. The positive Z-axis was chosen to be pointing downwards.



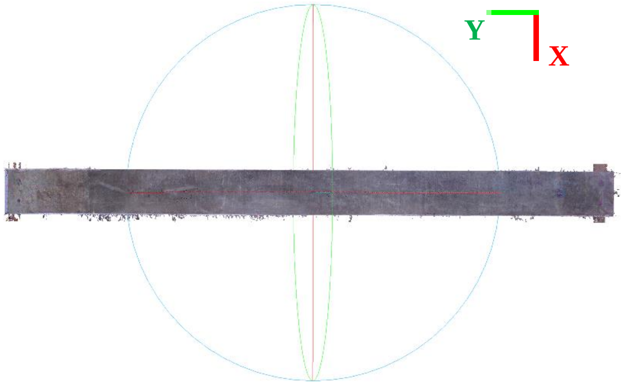


Figure 3.8. Aligning the Dense Point Cloud of Box Beam-3 with the Coordinate Axes

3.1.2.3.2. Scaling the Digital State Model and Dense Cloud to Real World Dimensions

In this step, the textured mesh and dense point cloud were scaled to real-world dimensions using a scaling factor. The scaling factor was determined using known dimensions on the girders. Two actual dimensions (for example, the width of the beam at a specific location) of the beam were measured using a tape measure. The corresponding dimensions on the point cloud were measured using the distance measurement tool on CloudCompare by picking the starting and the ending points. Scaling factors were calculated by dividing the actual dimensions by the corresponding dimensions measured on the point cloud at two different locations and, the average of these factors was used to scale the point cloud. CloudCompare has a feature to record all the transformations and scaling applied on the point cloud, and export them as a '.mat.txt' file. As mentioned in Section 3.1.2.2, the textured model of the girder was exported in '.obj' format. This exported file was used to apply the same transformations on the textured model ('.obj' file). Next, the dense point cloud was cleaned to remove the outliers. The use of dense point cloud and textured mesh are introduced in Section 3.1.2.5.

3.1.2.4. Cleaning the Dense Point Cloud

In this step, noise or outliers around a dense cloud of a girder were eliminated to obtain a cleaner point cloud using Statistical Outlier Removal (SOR) as described in Section 2.5.2. This step had two manually defined parameters, the number of nearest neighbors considered for each point and a multiplier to standard deviation as explained in Section 2.5.2. The optimal values to get a clean point cloud were determined using trial and error. CloudCompare has a SOR tool, that was used in this research.

3.1.2.5. Use of Dense Point Clouds and Textured Mesh Models of the Bridge Girders in this Research

After cleaning the dense point cloud, the vertex with minimum X, Y, Z coordinates was modified to (0, 0, 0) so that the coordinates of all vertices in the dense cloud were positive. This is referred to as zeroing the point cloud. Dense point clouds and textured mesh models of the bridge girders were used in the process of estimating the flexural capacities of the bridge girders. The dense clouds were used for extracting the damage clusters as explained in Section 3.1.2.7 and for extracting cross-section slices of the point cloud as explained in Section 3.1.3.1. The textured mesh models or digital state models of the girders were used, as explained in Section 3.1.3.6, as a visual tool for determining the corrosion damage to be considered to determine the moment capacity at a section. The idea was that, after scaling, cleaning, and zeroing, the dense point cloud and the textured mesh lay exactly on top of each other so that, any damage that went undetected in the damage detection could be visible on the textured mesh. This step added a layer of verification so that corrosion damage detected could be validated with the observed damage on the textured mesh. The textured mesh, in simple words, served as an image of the full-scale girder in three dimensions. For this reason, the textured mesh was translated so that it falls exactly on top of the zeroed dense point cloud of the girder. The next section deals with the detection of spall damage on the 3D dense point cloud (the exported '.ply' file) of the bridge girder.

3.1.2.6. Damage Detection on 3D Dense Point Cloud

Point normals based approach for detecting damage on 3D point clouds of concrete members described in Section 2.3.1 was used for detecting spalls on the 3D point clouds of the girders. A normal associated with a point on the dense cloud is estimated using plane fitting to 'k' number of neighboring points to the point of interest. The normal drawn to this plane fit was taken as the normal that corresponds to the point of interest. Principal Component Analysis explained in Section 2.3.1.1.1 was used to estimate the normal associated with each point. For each point, the data points (X, Y, Z coordinates) of 'k' neighboring points (including the point) were modified to give a zero mean. Covariance matrix of the adjusted vertices was calculated. Eigenvalues and eigenvectors were calculated for the covariance matrix and sorted based on the values of the eigenvalues. In this approach, the eigenvectors corresponding to the least eigenvalue represent the normal of the plane fit to the data points. Thus, by repeating this calculation for all the points in the point cloud, normals corresponding to all the points were calculated.

Angle made by each normal with a reference direction had to be calculated and used to detect the damage. In this research, three reference directions along X, Y, Z axes were considered. The angles made by a point normal with X, Y, Z axes can be represented as β_x, β_y and β_z. Absolute values of difference of these angles β_x, β_y and β_z and 90 degrees ($|90- \beta_x|$, $|90- \beta_y|$ and $|90- \beta_z|$) were calculated and compared with a threshold angle (θ_{th}), which was manually defined. If $|90- \beta_x| > \theta_{th}$ and $|90- \beta_y| > \theta_{th}$ and $|90- \beta_z| > \theta_{th}$, then the point was considered to be a part of the damage. It should be noted that these conditions involving each of the X, Y, Z axes were joined together by an 'and' operator. More information on this is presented in Section 4.2.3 where the results of the algorithm are discussed. If the threshold angle (θ_{th}) was too small, most of the concrete surface roughness also got captured in the damage, which was undesirable as they were not parts of the spalls. To filter out some of this background noise, different values for threshold were tried to reach an appropriate value that gave less noise.

Another trial and error parameter in this approach was the value of 'k', the number of nearest neighboring vertices around a point used to fit a plane, and to compute the normal. A lower value of 'k' captured local surface roughness of concrete along with the cracks and spalls. As higher values of 'k' were used, the damage that had a larger depth was detected. Since, crack-like details were already captured from the 2D level damage detection on images, in this step only spalls were desired to be detected. For achieving this, instead of trying low values (less than 10) for 'k' as observed in the literature, higher values were tried and an appropriate value of a number of nearest neighbors, 'k' was found such that fewer portions of local roughness and cracks were captured in the damage detected. The damage detected by choosing this high value of 'k' only consisted of portions of the spalls. In each of the spalled regions, different parts made different angles with the reference directions. So, by choosing a value of the threshold and by choosing a value of 'k', some of these spalled regions were undetected in this approach. An effort was made to choose a value of 'k' to achieve a balance between these two aspects, reducing background noise while giving considerably better spall detection. This step highlighted portions of the spalls on the 3D point cloud. The effects of these two parameters on the damage detected are demonstrated in Section 4.2.3. This algorithm was programmed in Python. The points where the normals violate the thresholding limit were highlighted in red color and were exported as a separate point cloud (consisting of only red points).

The size of the original dense point cloud used was in the range of 15 million to 28 million vertices. Processing these large number of vertices for the detection of spalls, was computationally very expensive. The time taken to run this program increased exponentially as ‘k’ value was increased. To address this issue, the dense cloud was downsized to about 4 million points to reduce the computation time. The computational capacity of the computer used for this process was 32GB of memory with dual GPUs as described in Section 3.2. The procedure or the algorithm used for downsizing is described in Section 3.1.2.6.1.

Spall detection could also be done using a Neural network on 2D images of the concrete girders and they could be transferred to a 3D level similar to the crack detection described in Section 3.1.2.1. But, by using a thresholding type of spall detection adopted in this research, the need to collect a labeled dataset for spall detection and training a model of spall detection were avoided.

3.1.2.6.1. Downsizing the Dense Cloud for Spall Detection

The dense point cloud was downsized (or downsampled) before spall detection, using a voxel grid-based approach. The user-defined parameter in this approach was called ‘voxel size’. A subdivision of the space over the dense point cloud into a 3D grid (called voxel grid) using fixed-width boxes (called voxels), was generated, the fixed width of the boxes was manually specified as the ‘voxel size’ parameter. The points that lay inside each voxel were replaced with their centroid. Thus, the total number of points in the dense cloud was reduced (Open3D Point Cloud Tutorial, 2020). This decreased computational speed for 3D level damage detection, drastically. A voxel size of 0.2 was found to be effective. The effect of the manually defined parameter, voxel size, on the resulting reduced point cloud is reported in Section 4.2.3.3.

3.1.2.7. Cleaning the Detected Damage and Clustering

As discussed in Section 3.1.2.2, cracks detected on the textured mesh were transferred on to the dense cloud as blue colored points. In this step, only the blue colored points are filtered out from the dense cloud using thresholding and saved as a separate point cloud file in ‘.ply’ format.

The crack points on the dense cloud were in different shades of blue color and the R, G, B values of these points were not exactly (0, 0, 255), and hence they had to be separated using thresholding on color (R, G, B) values of the points. The points with color values of 'R' component less than any value in the range of 114 to 153, 'G' component less than a value in the range of 114 to 153, 'B' component greater than any value in the range of 153 to 166, were filtered out into a new point cloud file as points on cracks. The exact values of R, G, B used for thresholding were picked from the ranges mentioned above and were determined using trials. These threshold values were determined using trial and error. Also, as described in Section 3.1.2.6, the damage detected at the 3D level was already exported in a separate point cloud (also in '.ply' format). Thus, all points belonging to cracks and spalls were exported as separate point cloud files. These individual cracks and spalls point clouds were subjected to a clustering algorithm to divide the whole damage detected into separate groups called clusters. Before this step, the detected damage consisted of some noise that had to be cleaned. Furthermore, the output of 3D level damage detection contained regions where the normals of the points did not satisfy the thresholding criteria. These regions included some surface roughness on the top and sides of the girders and surface holes that did not constitute corrosion damage on the bridge girders. Hence, cleaning these crack and spall point clouds was required to isolate the corrosion damage from all the damage and noise detected in these point clouds. This cleaning was done using SOR (described in Section 2.5.2) and using manual cleaning using CloudCompare. 'Segment tool' was used to draw bounding polygons on the cracks and spalls that constitute corrosion damage to separate them from the noise and other surface roughness. The output from this step included cleaned crack and spall damage point cloud files which were used for clustering of the damage.

3.1.2.7.1. Extracting Damage Clusters

For clustering on the cleaned crack and spall damage points, Density Based Spatial Clustering of Applications with Noise (DBSCAN) was used, as this is good for data with clusters of similar density. There are two user-defined parameters, 'minimum samples' and 'eps', which were determined using trial and error. Using a higher value of the 'minimum samples' parameter gave better noise filtering. However, as the cleaning was done manually, the 'minimum samples' parameter was limited to 5 (almost no noise filtering) in this step. The attempt was to isolate each crack or spall into a separate cluster. The results of different clustering parameters and the relevant discussion are presented in Section 4.2.4. Each damage cluster was saved in a separate point cloud file at the end of this step. These individual damage cluster point clouds are used for residual strength estimation as described in Section 3.1.3.6.

3.1.2.7.2. An Attempt at Cluster Dimension Extraction

Planes were fit to the undamaged bottom and side surfaces of the dense point clouds of the beams using region growing algorithm described in Section 2.7. The program for region growing was adapted from Smith (2020). Each damage cluster was projected on to the planes fit to the undamaged bottom and side surfaces. For an I-Beam, plane can be fit to the side surface of the bottom flange. Bounding minimum area rectangles were fit automatically around each of the projected clusters (on both the bottom and side planes). The bounding rectangles around the projected damage clusters on the side and bottom planes were used to give an idea about the location and the extent of this damage cluster. Approximations of the lengths of the crack clusters in two dimensions were determined from the length of the bounding rectangles on the bottom and side planes. Then, an attempt was made to approximately determine the length of the cracks in three dimensions. All the points corresponding to each crack cluster were sliced into a prescribed number of slices. Vertices that lay between two consecutive slices were replaced with the centroid of these points so that there would be just one point between any two consecutive slices. The distances between all the consecutive centroids were added to give an approximate length of the crack in three dimensions.

Approximations for the widths of the spall clusters were estimated as the widths of the bounding rectangles on the bottom plane and the approximate depths of spalls were determined as the width of the bounding rectangles on the side plane. The parameters included neighborhood size ‘k’ of 15, minimum points of 1000, a smoothness (angle) threshold of 3 degrees. For plane fitting to the bottom and side surfaces, only the vertices that lay in the extreme 2.5% to 4.5% of the corresponding dimension were considered. The results of these estimations of damage cluster dimensions are shown in Section 4.2.4.1.

3.1.3. Residual Strength Estimation of the Girders

At this point in the procedure, the dense point clouds of the bridge girders had cracks and spalls highlighted on them. Also, the textured mesh models had cracks detected and highlighted on them. Next, cross-sections of the dense point clouds were extracted at desired locations along the length, where residual capacities of the girders were desired to be evaluated. These extracted cross-sections are saved as separate image files ('.png' format) and referred to as 'slices' in the rest of the report. As the as-built drawings of the bridge were not available, these slices were later overlaid on top of design CAD drawings of the cross-sections of bridge girders to create visual aids that facilitate identification of the prestressing strands that are adjacent to longitudinal cracks and the strands that lie in spalled regions. This is discussed in Section 3.1.3.4. The procedure followed for slice extraction from the dense point clouds is explained in this section.

3.1.3.1. Extracting slices from a 3D Point Cloud

The locations on the beam at which the slices are taken are referred to as 'slice-sections'. As described in Section 2.8, to calculate the residual strength at a section, recommendations of Naito et al. (2011) suggest considering the damage existing within a region of one development length of the strands on each side of the section. Therefore, the slices not only captured the damage present at the section but also included projected damage within a region of one development length of the bottommost layer of strands on each side of the slice-section.

3.1.3.2. Slice Spacing

Cross-sectional slices were extracted from the point cloud at locations where the flexural capacity was needed to be calculated. The first and last slices were taken a distance of 1 ft from the ends of the point cloud. For the slice extraction step, initially, a trial was conducted on the dense point cloud of the box beam with an intermediate slice spacing of 2.5 ft. The development length of the strands in the bottom layers of the bridge girders was calculated using ACI 318-14 Equation 25.4.8.1 to be about 5.5 ft. This development length of strands depends on the stress in the prestressing steel at the nominal strength of the member (f_{ps}) and on the effective prestress in the strands (f_{se}). ACI 318-14 (Equation 25.4.8.1) recommends calculating the development length (L_d) of pretensioned seven-wire strands as follows.

$$L_d = \left(\frac{f_{se}}{3000}\right) d_b + \left(\frac{f_{ps} - f_{se}}{1000}\right) d_b \tag{3.1}$$

Here d_b is the nominal diameter of the prestressing strands. Equation 5.9.4.3.2-1 in AASHTO LRFD Bridge Design Specifications, 2017 for the calculation of development length of strands, shown in Equation 3.2 also yields the same result as the ACI equation when κ in Equation 3.2 is taken as 1. κ is a multiplier based on the depth of the member. f_{ps} and f_{se} in this equation are used in ksi units.

$$L_d \geq \kappa \left(f_{ps} - \frac{2}{3} f_{se}\right) d_b \tag{3.2}$$

It was observed that using an intermediate slice spacing of 2.5 ft was too small, and many consecutive slices ended up having the same damage associated with them because all damage in the ‘two development lengths’ region was considered. For this reason, the intermediate slice spacing was modified to one-tenth of the total length of the point cloud. As the total length was about 50 ft for the Lesner Bridge I-beams, this intermediate slice spacing came out to be around 5 ft. Spacing between the slice-planes for Box Beam-3 is shown in Figure 3.9.

3.1.3.3. Algorithm for Slice Extraction

The algorithm for slice extraction was taken from Smith (2020). Firstly, a dense point cloud of a bridge girder was fed as an input to the algorithm. The coordinates of the vertices in the point cloud were sorted in the increasing order of their Y co-ordinates. The total length of the point cloud is determined by calculating the difference between the maximum and minimum Y coordinates (minimum Y coordinate was zero as the point cloud was previously zeroed). So, the bounding box around the point cloud needed to lay exactly on top of both the ends of the point cloud. For the box beam point cloud, a user-defined parameter ’n’ was given as 21 (number of intermediate slices). This parameter could be obtained based on our intermediate slice spacing, which was 2.5 ft for the box beam. For a total box beam point cloud length of 55ft and an intermediate slice spacing of 2.5ft, we obtained 23 slices. As the first and last slices were both external slices set to be extracted at a distance of 1 ft from their respective ends, there were 21 remaining internal slices.

The algorithm required a point cloud containing only the detected cracks and another point cloud of only spalls, and an estimate of development length of the bottommost layer of strands. Vertices belonging to the cracks and spalls that were located at a distance of one development length on each side of the slice-section were projected on to the slice. A total distance of two development lengths of the strands with a slice-section at the midpoint is henceforth referred to as an ‘Inspection Window’. More information on the purpose of the inspection window is explained in Section 3.1.3.6. The vertices that belong to cracks and spalls had blue color and red color respectively, on the slices. Next, vertices with Y-coordinates within a length of 0.05 ft before and after the slice of interest were extracted from the array of vertices. All these vertices (the vertices in the point cloud within a distance of 0.1ft around the slice, and the crack and spall points within a region of two development length regions projected on top of the slice) along with their original colors were written to an image file (‘.png’ format). One such image was written for each one of the slices. Also, a plane, called a slice-plane, was fit to the vertices at each slice location and written as a separate output point cloud for convenience in visualization. All the slice-planes could now be plotted on top of our original dense point cloud or textured mesh so that it would be easier to identify the damage clusters on the dense point cloud that lay close to each slice-plane. The slice-planes on the textured mesh of Box beam-3 are shown in Figure 3.9.

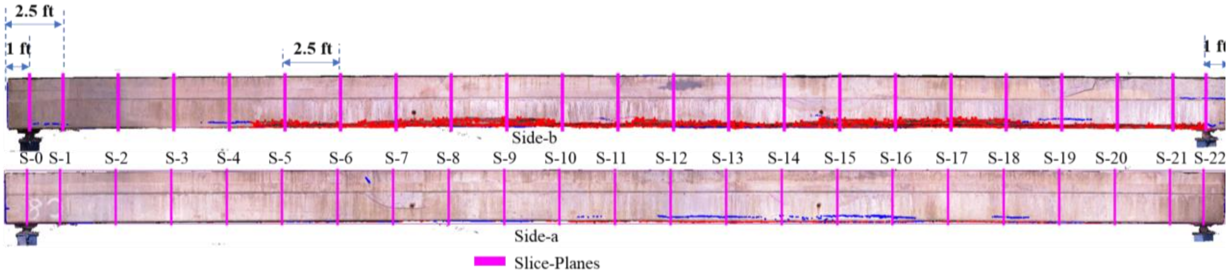
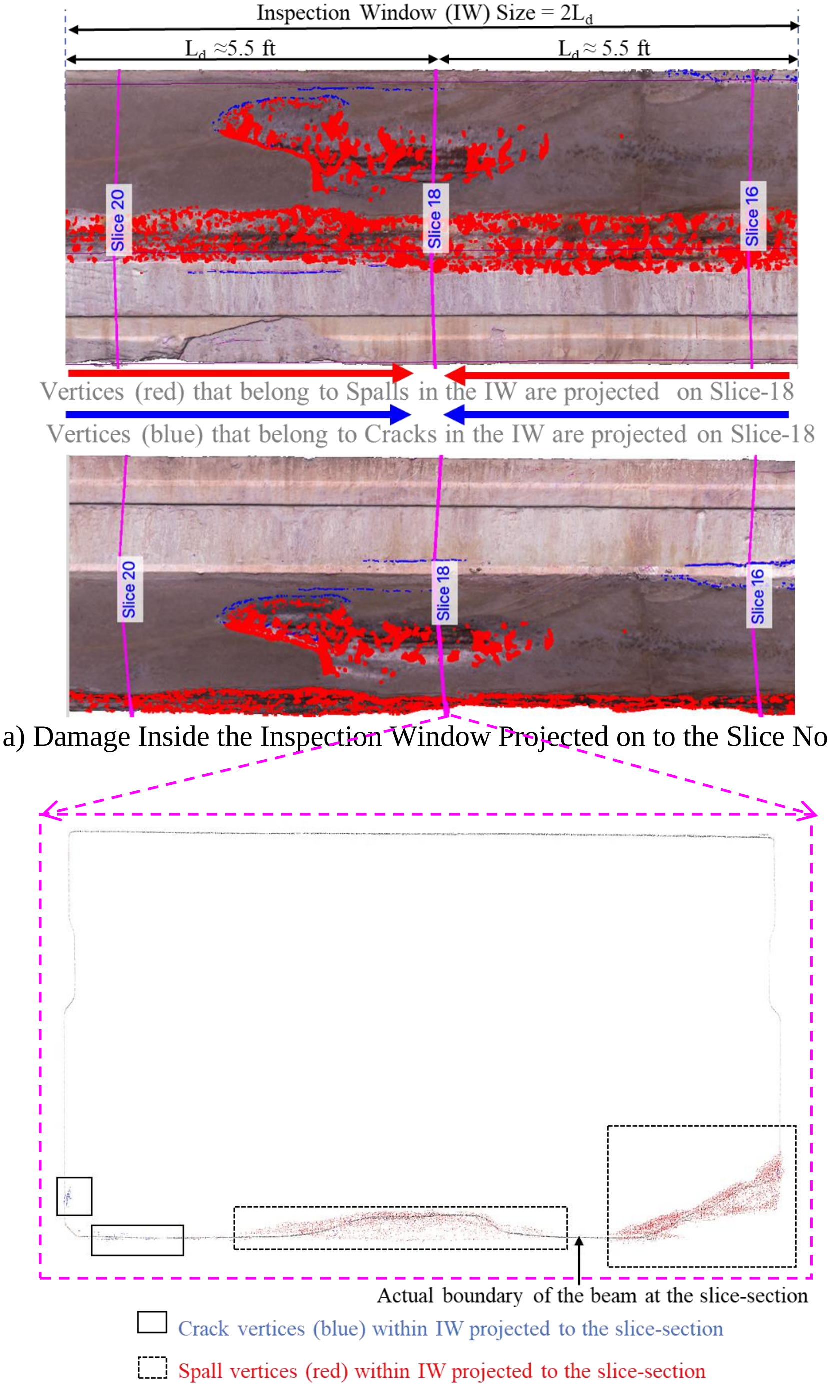

2.5 ft
1 ft
2.5 ft
1 ft
Side-b
S-0 S-1 S-2 S-3 S-4 S-5 S-6 S-7 S-8 S-9 S-10 S-11 S-12 S-13 S-14 S-15 S-16 S-17 S-18 S-19 S-20 S-21 S-22
Side-a
Slice-Planes

Figure 3.9. Slice-Planes on the Textured Mesh of Aden Road Bridge Box Beam-3

Furthermore, the individual damage clusters were used to determine the clusters that were located close to each of the slices. An output text file was written with details of the clusters that lie close to each of the slices within a distance of one development length on each side. The closest distance of a boundary of the damage cluster from the slice was also included in the output text file so that it would be easy for the user to identify these damage clusters around each slice under consideration. Next, the procedure for superimposing these slices on top of the design cross-sectional drawings of the bridge girders is described in Section 3.1.3.4. An example Slice No. 18

extracted from the box beam dense point is shown in Figure 3.10. Red and blue points constituting spalls and cracks, respectively projected on to the Slice-18 are demonstrated in Figure 3.10a.


Inspection Window (IW) Size = 2L$_d$
L$_d$ ≈5.5 ft
L$_d$ ≈ 5.5 ft
Slice 20
Slice 18
Slice 16
Vertices (red) that belong to Spalls in the IW are projected on Slice-18
Vertices (blue) that belong to Cracks in the IW are projected on Slice-18
Slice 20
Slice 18
Slice 16
Actual boundary of the beam at the slice-section
Crack vertices (blue) within IW projected to the slice-section
Spall vertices (red) within IW projected to the slice-section

a) Damage Inside the Inspection Window Projected on to the Slice No. 18

b) Slice No. 18 with Projected Red and Blue Points (Damage)

Figure 3.10. Slice No.18 from Box Beam-3 Showing Projected Damage

3.1.3.4. Overlay of Point Cloud Slices on Top of the Cross-Section Drawings

In this step, slices extracted from the point cloud (in '.pdf' format) as described in Section 3.1.3.3, were superimposed on top of the design cross-sectional drawings of the bridge girders using AutoCAD. As the as-built drawings of the bridge girders were not available, design drawings were used in this step. These slices had to be scaled to real-world dimensions of the girders before superimposing them. Firstly, an initial scaling factor was determined by dividing the actual depth of the beam (in real-world dimensions) with the corresponding dimension in the image of the slice embedded in the .pdf file. Later, this scaling factor was refined, and a rotation angle was determined by manually superimposing the slice and the cross-section drawing with each other so that the edges of the point cloud slice aligned with the edges of the cross-sectional drawing. Next, this predetermined scale factor and rotation angle were used to superimpose the remaining slices. In these overlays, spalls at the location of the slice were now clearly visible. Also, the points corresponding to the cracks and spalls projected on to the slice were visible in blue and red colors. In the next step, these points on the overlays were used to locate the cracks and spalls on the dense point cloud, to be considered in residual moment capacity calculations for the required slice-sections. All the damage to be considered around each slice-section was documented and the residual capacity was calculated. This process is described in Section 3.1.3.5. An example overlay of Slice No. 16 extracted from Box Beam-3 dense point cloud with the cross-section drawing is shown in Figure 3.11.

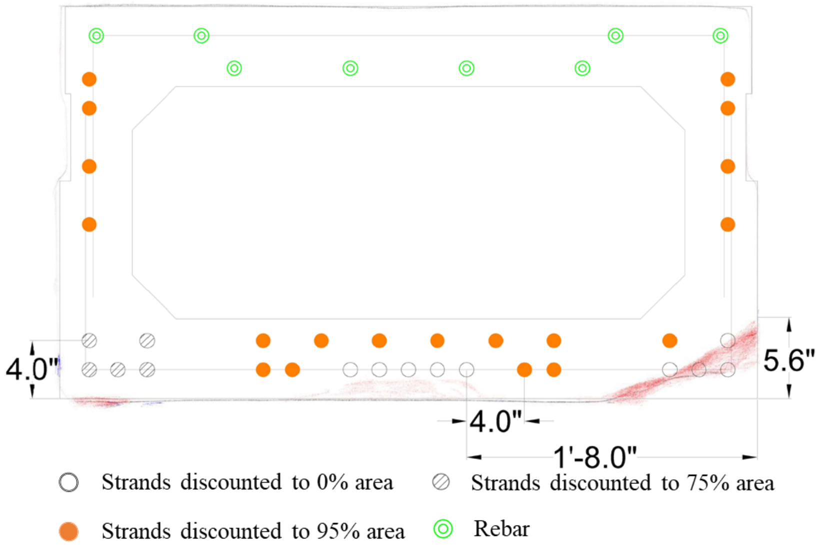

4.0"
5.6"
4.0"
1'-8.0"
○ Strands discounted to 0% area
⊘ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

Figure 3.11. Overlay of Slice No. 16 from Box Beam-3 Dense Point Cloud with the Cross-Section Drawing

3.1.3.5. Residual Capacity Estimation at the Slice-Sections

In this step, all the tools formulated until this point were used to identify the corrosion damage which is to be considered for estimating the residual capacity of the beams at the slice-sections. Reduced areas of strands at the slice-sections were arrived at, after accounting for corrosion damage as per the recommendations proposed by Naito et al. (2011), which are described in Section 2.8. Moment capacities at the slice-sections were calculated using a strain compatibility approach. The tools gathered until this point were the dense point cloud, the textured mesh, crack and spall damage clusters, slice-planes (planes fit at each slice-section), an automatically generated text file that listed all the damage clusters to be considered for strength evaluation at each slice-section and the overlay of slice and cross-sectional CAD drawing. The procedure followed for residual strength estimation of Aden Road Bridge Box Beam-3 and Lesner Bridge I-Beam 9, mentioned in Section 3.1.4, using all these tools is described in the following sections.

3.1.3.6. Corrosion Damage to be Accounted for in Strength Calculation at a Slice-Section

Recommendations proposed by Naito et al. (2011) suggest that corrosion damage within a distance equal to one development length of the strands on each side of the section has to be considered in strength calculation. A total distance of two development lengths of the strands with a slice-section at the midpoint is referred to as an ‘Inspection Window’. Damage present inside an inspection window has to be assumed to be present at the section where strength is to be calculated by arriving at a reduced area of strands.

Slice and CAD drawing overlay was the first source to determine the damage to be considered for the reduction of the cross-sectional area of the strands. Strands that were either covered by red points (spalled regions projected on to the slice from a distance within a length of two development lengths around the section) or the strands that were close to the blue points (crack vertices projected on to the slice) were reduced in the cross-sectional area either partially or fully in moment capacity calculation at the slice-section, as per the recommendations described in Section 2.8.

As mentioned in Section 3.1.2.3, CloudCompare, a 3D point cloud and mesh processing software that is available open-source (free of cost) was used as a visualization tool in this step.

Firstly, the textured mesh, the dense point cloud, the slice-planes, and the crack and spall point clouds were opened in CloudCompare. It has some features to facilitate feature measurement and better visualization. Distance measuring tool for measuring distances between any two points on a point cloud, increasing the point size (only for better visualization), option to display custom names for each point cloud or mesh that is open in the current window are some of the features were used in this step as a visual tool for identifying the corrosion damage affecting the prestressing strands. The slice-plane at which the moment capacity of the beam is to be calculated was highlighted with a larger point size for better visualization, and custom names were displayed on top of the point clouds in the 3D view (display window).

The strands affected due to the detected corrosion damage were identified from the overlay as explained at the beginning of this section. The textured mesh of the beam inside the inspection window was superimposed with the damage (crack and spall) clusters (in the CloudCompare window). Then, the window was examined to validate the observed damage on the overlay. The slice-plane at which strength had to be calculated was at the midpoint of this inspection window. In this step, additional guidance on identifying this damage was provided by the text file containing the list of clusters to be considered for the slice-section. Cracks that went undetected in the model because of the false negatives from the 2D level damage detection were identified on the textured model which, in simple words. At every slice-plane, the textured model was carefully examined to account for these undetected cracks. Distance to a strand from the surface of the beam was measured using the distance measurement tool to identify the strands exposed on the point cloud. Distances to the longitudinal cracks from the surfaces were also measured using this tool to identify the strands that lay close to the cracks. The implementation of this entire process for some cross-sections is illustrated in Section 4.3.

After these observations are noted, prestressing strands that are affected due to the corrosion damage were examined. The strands with a partially or completely exposed cross-section due to spalling inside the inspection window were completely discounted from the calculations. The strands in the bottom two layers, that were directly in-line with longitudinal cracks or adjacent to the cracks inside the window, lying within a distance of 3 in., were noted and reduced to 75% of their cross-sectional area in the strength calculations. Also, as the beams had longitudinal cracking, all the other strands in the cross-section, which were neither reduced to 0% nor 75%, were reduced to 95% of their cross-sectional area. After all these applicable reductions were made, the reduced areas of prestressing strands in each layer and the modified centroid of the reduced strands were calculated. These were used in the Strain Compatibility analysis to calculate the moment capacity at the slice-section.

3.1.3.7. Strain Compatibility Analysis for Moment Capacity Calculation

In a prestressed concrete beam, a flexural failure can occur either with crushing in concrete or with corroded strands rupturing before the concrete crushes. In strain compatibility analysis, one of these failure modes is assumed by prescribing a value of ultimate crushing strain in concrete (ε_{cu}) or by limiting the total rupture strain in prestressing steel (ε_{ps}). ε_{ps} is calculated as a sum of three component strains. Firstly, strain in prestressing steel (ε_1) caused by the effective prestress and the compressive strain in concrete at the level of prestressing steel (ε_2) if the only force applied was the prestressing force, are calculated. Depth to Neutral axis of the cross-section (c) from the extreme compression fiber of concrete is assumed. Then, the strain in concrete at level of prestressing steel when section develops its nominal capacity (ε_3) is calculated. Total strain in the prestressing steel ($\varepsilon_{ps} = \varepsilon_1 + \varepsilon_2 + \varepsilon_3$) is used to estimate the stress in prestressing steel at nominal strength of the member (f_{ps}) using the assumed constitutive stress-strain relationship for prestressing steel. Total Tensile force (T) and Compressive force (C) at the cross-section are calculated. The value of 'c' is iterated to ensure equilibrium of forces at the cross-section (until C = T). Flexural capacity is calculated using the appropriate moment arms.

In the current research, strain compatibility analysis was performed using different parameters for the materials: concrete and prestressing steel, in order to enable comparison among the results of the analysis using different parameters. These material parameters are listed in Table 3.2. Strain compatibility analysis was performed with the failure of the cross-section controlled by either crushing of concrete or by fracture of prestressing strands (denoted by either 'CON' for crushing in concrete or 'STL' for rupture of prestressing steel in the analysis listed in Table 3.2). This is achieved by prescribing a limiting strain that causes crushing in concrete or a fracture in the prestressing strands. In the analysis, two constitutive relationships were used for each of the materials. Constitutive relationships for prestressing steel are denoted in Table 3.2 as either 'THR' for theoretical stress-strain curve or 'EXPT' for experimental stress-strain curve. The experimental stress-strain curve for prestressing steel was adapted from a tension test conducted by Alfailakawi (2019), shown in Figure 2.6a. The theoretical stress-strain curve for prestressing steel mentioned in this table is shown in Figure 3.12. Models for concrete are denoted in Table 3.2 as either 'WHT' for Whitney stress block or 'HOG' for Hognestad Model. Hognestad Stress-Strain model for concrete is shown in Figure 3.13. Reduction in the cross-sectional area of prestressing steel mentioned in this table (denoted by letter 'R') was calculated either following the recommendations of Naito et al. (2011) or using modified recommendations as explained in the corresponding sections of the report.

In the analysis, the initial stress in the strands after transfer was taken from the original drawings of the girders. Effective prestress (f_{se}) in the strands was calculated after estimating the time-dependent losses in prestress using the AASHTO model. The centroid of prestressing strands in the original condition (without any corrosion damage) was considered to calculate the prestress losses. The effective prestressing force was calculated by multiplying the effective prestress with the reduced cross-sectional area of strands determined after by accounting for the corrosion damage as explained in Section 3.1.3.6. All dimensions of the beams were obtained from the original design cross-section drawings of the bridge girders. Compressive strength of concrete (f'_c) was obtained from tests conducted by Al Rufaydah (2020) on concrete core samples extracted from the beams whose results are listed in Section 3.1.4. All non-prestressed flexural reinforcement was accounted for in the calculation. The strand profile was straight, with no harping for both Lesner and Aden Road Bridge beams. For the slice-sections located inside a distance equal to the development length of the strands from the nearest end of the beam, force in the strands was calculated accordingly accounting for the location of the sections as the strands cannot develop stress equal to that at nominal strength of the member (f_{ps}) at these locations. The different material parameters used for strain compatibility analysis are explained in the following sub-sections.

Table 3.2. Different Material Parameters Used for Strain Compatibility Analysis in this Research

Material Parameters	**Assumed Failure Mode**	**Limiting Strain in the Material Mentioned in Column 2**	**Concrete Material Model**	**Stress-Strain Model for Prestressing Steel**	**Cross-Sectional Area of Strands Considered**
CON-WHT-THR-R	Crushing in concrete	0.003	Whitney Stress Block	Theoretical Stress-Strain Curve	Reduced

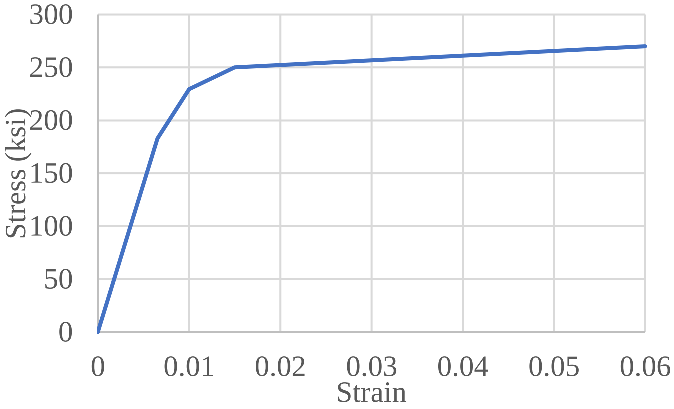
STL-HOG-THR-R	Rupture of Prestressing Steel	Variable	Hognestad Model	Theoretical Stress-Strain Curve	Reduced
STL- HOG-THR-ORG	Rupture of Prestressing Steel	Variable	Hognestad Model	Theoretical Stress-Strain Curve	Original or undamaged area of prestressing steel
CON-HOG-THR-R	Crushing in concrete	0.0038	Hognestad Model	Theoretical Stress-Strain Curve	Reduced
CON-HOG-EXPT-R	Crushing in concrete	0.0038	Hognestad Model	Experimental Stress-Strain Curve	Reduced

3.1.3.7.1. CON-WHT-THR-R Material Parameters

A spreadsheet was set up to calculate the moment capacity of the beam at each slice-section. As the original stress-strain relationship for the strands was not available, a theoretical stress-strain curve shown in Figure 3.12 was adopted to determine the stress in the prestressing strands at nominal flexural strength (f_{ps}) in the strain compatibility analysis. An ultimate strain in concrete (ε_{cu}) was assumed to be 0.003 and stress distribution in concrete was approximated using Whitney stress block. Whitney stress block assumes uniform compressive stress of 0.85f'c distributed over an equivalent compression zone. From this analysis, the residual capacity of the beams at each of the slice-sections was calculated after accounting for the existing corrosion damage as previously described.

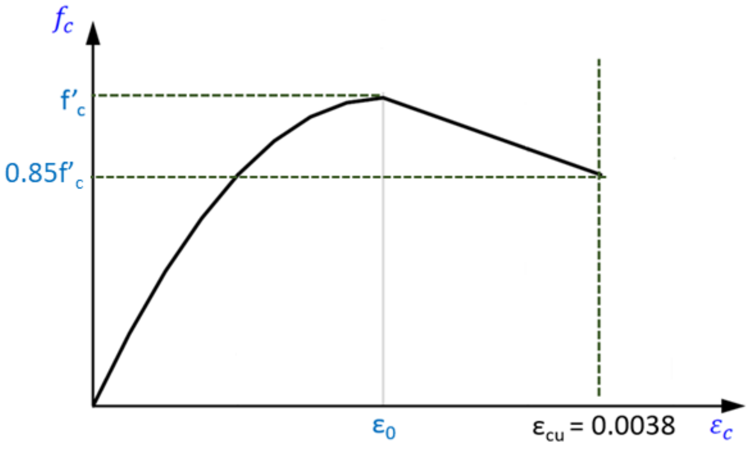
3.1.3.7.2. STL-HOG-THR-R Material Parameters

In the analysis described above, the failure is governed by the ultimate strain in the extreme compression fiber reaching 0.003. Here, a target total ultimate strain in the strands (ε_{ps}) in the bottommost layer was chosen to mimic a strand rupture failure occurring before the concrete crushes in compression. A trial value of ε_{ps} was chosen and the compatible components of strain in other layers of strands were computed. The corresponding value of ultimate strain in concrete was determined from the linear strain distribution along the depth of the cross-section of the beam. A reduced area of prestressing steel was used for calculation after accounting for corrosion damage as proposed by Naito et al. (2011). A theoretical stress-strain curve shown in Figure 3.12 was adopted for the behavior of prestressing strands. Hognestad model with an ascending parabolic region and descending straight line region as shown in Figure 3.13 was used for stress-strain behavior of concrete. Based on the ultimate strain in concrete, it was determined if it corresponds to the parabolic region or the linear region of the stress block. Total compressive force (C) in concrete was calculated using an assumed value of the depth to the Neutral Axis (c). Tension force (T) in the prestressing strands and forces in the non-prestressing bars were calculated. The calculation was repeated to get an updated value of ‘c’ that gave equal Compressive and Tensile forces in the section. For the beam that failed with rupture of the bottommost layer strands, this analysis was used to find a value of ε_{ps} that gives the same moment capacity as the demand at the section of failure during the test to find the ultimate strain in the strands at rupture.



Stress (ksi)	Strain
0	0
183	0.00654
229.5	0.01
250	0.015
270	0.06

Figure 3.12. Theoretical Stress-Strain Curve for Prestressing Strands used in Moment Capacity Calculation



Ascending Branch ($\varepsilon_c < \varepsilon_0$):

$$f_c = f_c' \left[\frac{2\varepsilon_c}{\varepsilon_0} - \left(\frac{\varepsilon_c}{\varepsilon_0} \right)^2 \right]$$

Descending Branch ($\varepsilon_c \geq \varepsilon_0$):
Straight line connecting the peak to 0.85f’c @ strain ε_{cu} = 0.0038

$$E_t = 1.8 \times 10^6 + 460(f'c)$$

$$\varepsilon_0 = \frac{2 * f'c}{E_t}$$

$$\mathrm{k} = \frac{\varepsilon_0}{\varepsilon_{cu}}$$

Figure 3.13. Hognestad Model for Concrete

3.1.3.7.3. STL-HOG-THR-ORG Material Parameters

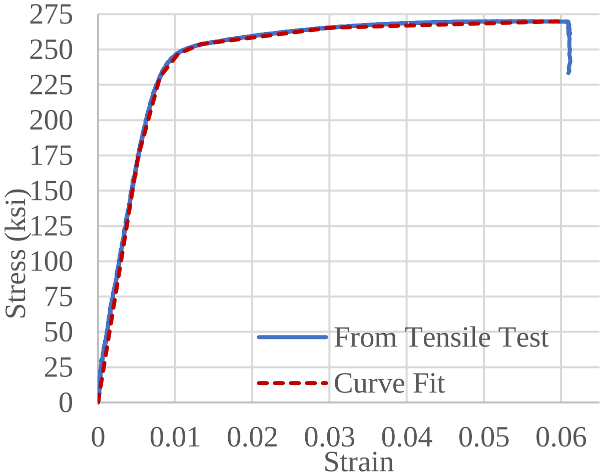
Instead of taking a reduced area of strands for strength calculations, the full undamaged cross-sectional area of strands was considered in the analysis. The idea behind this calculation was that, by limiting the ultimate strain, the stress in the strands at the ultimate capacity of the section was also constrained. Thus, the effect of corrosion damage was already accounted for. On top of this, if the strands were reduced in cross-sectional area to account for the corrosion damage present within the inspection window of two development lengths around the section, the effect of corrosion damage was being considered twice.

3.1.3.7.4. CON-HOG-THR-R Material Parameters

Hognestad stress-strain model (shown in Figure 3.13) was used for the stress block for concrete in compression, and for calculating the total compressive force in concrete. Moment capacity thus calculated was compared with that calculated using the procedure in Section 3.1.3.7.1.

3.1.3.7.5. CON-HOG-EXPT-R Material Parameters

Experimental stress-strain curve shown in Figure 2.6a from a tension test conducted by Alfailakawi (2019) on a strand sample extracted from Lesner Bridge I-Beam 3 was adopted for the behavior of prestressing strands. This experimental stress-strain curve is shown in Figure 3.14. Also, the Hognestad stress-strain model was used, for the stress block for concrete in compression, and for calculating the total compressive force in concrete.



Curve Fit for Experimental Data

Stress (ksi)	**Strain**
0.00	0
173.7	0.0052
230.8	0.0080
247	0.0104
253.7	0.0134
265.4	0.0301
270	0.06

Figure 3.14. Curve Fit to an Experimental Stress-Strain Curve for Prestressing Strand from Test by Alfailakawi (2019) on Lesner Bridge I-Beam 3

3.1.4. Destructive Strength Tests on Aden Road Bridge Box Beam-3 and Lesner Bridge I-Beam 9

This section discusses tests performed by Al Rufaydah (2020) on Aden Road Bridge Box Beam-3 and Lesner Bridge Beam-9. Cross-sections of these beams obtained from the original drawings of the beams are shown in Figure 3.15 and Figure 3.16. The total length of Box Beam-3 was 55 ft, and the length of the Lesner Bridge I-Beam 9 was 49.75 ft. Each end of these beams was tested in a three-point loading test with an intention to get a shear failure. Among the two ends of each beam, the end with less corrosion damage and the end with comparatively more corrosion damage were identified. In Test-1 for each beam, the end with less corrosion damage was tested, and the more damaged end was tested in Test-2 for each beam. The location of point load for each test was determined based on shear capacity calculations of the beams so that the resulting shear span was neither too short to cause a direct diagonal crushing failure, nor too long to result in a flexural failure. However, flexural failures occurred in two of these tests which are discussed in the following sub-sections. Test setup, procedure, and failure mode observed in each of these tests are described in Sections 3.1.4.1 and 3.1.4.2. A summary of the maximum moment demand at the section where the failure occurred and the type of failure observed in each test are shown in Table 3.3. Al Rufaydah (2020) conducted compressive strength tests on concrete cores extracted from these two beams. The average compressive strength of three cores for each case are presented in Table 3.4.

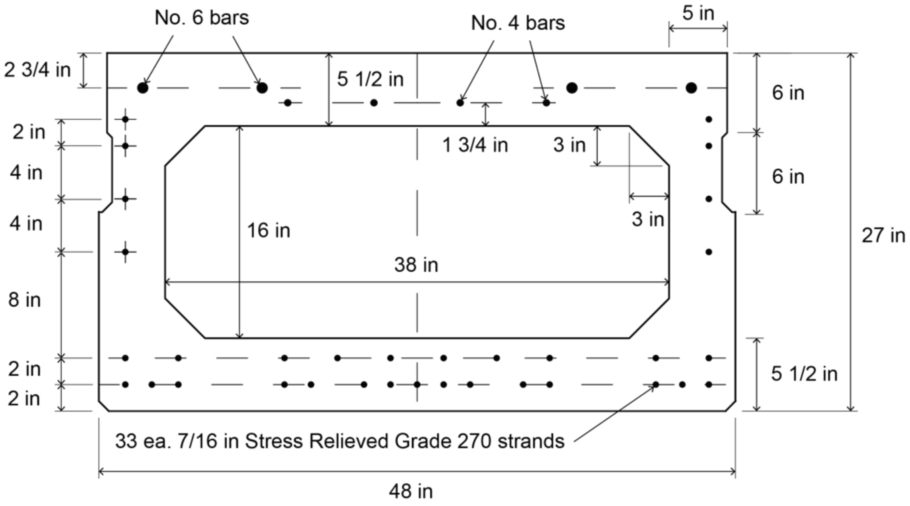


Figure 3.15. Cross-Section of Aden Road Bridge Box Beam-3 from Alfailakawi (2019)

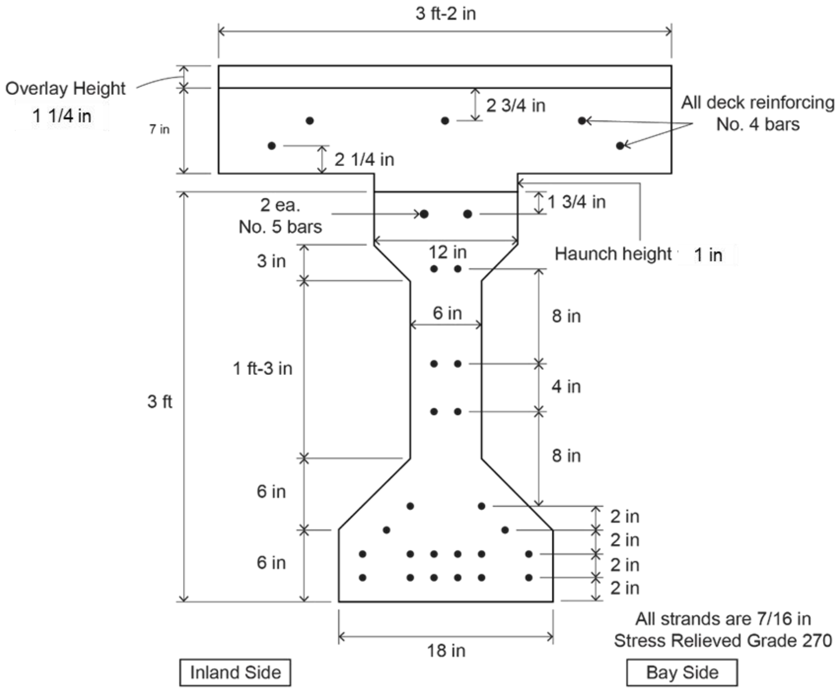


Figure 3.16. Cross-Section of Lesner Bridge I-Beam 9 from Alfailakawi (2019)

Table 3.3. Maximum Moment Demands and Modes of Failure in Tests on Box Beam-3 and I-Beam 9

Girder	**Test**	**Maximum Load, kip**	**Maximum Moment Demand at the failure sections, kip-ft**	**Failure Mode**
Aden Road Bridge Box Beam -3	Test -1	259.2	1029	Shear Failure
Aden Road Bridge Box Beam -3	Test -2	230.5	848.6	Rupture of strands in the bottommost layer
Lesner Bridge I-Beam 9	Test -1	264.8	1880	Flexural failure with concrete crushing in the deck
Lesner Bridge I-Beam 9	Test -2	382.4	1910	Shear Failure

Table 3.4. Average Compressive Strength of Concrete Cores Extracted from Box Beam-3 and I-Beam 9 (Al Rufaydah, 2020)

Name of the Beam	**Average Compressive Strength of Concrete in the Beam, psi**	**Average Compressive Strength of Concrete in the Deck, psi**
Aden Road Bridge Box Beam-3	6080	NA
Lesner Bridge I-Beam 9	4957	4090

3.1.4.1. Tests on Aden Road Bridge Box Beam-3

3.1.4.1.1. Test-1 on Aden Road Bridge Box Beam-3

The end of the beam with comparatively less corrosion damage was chosen for this test to determine a base-line shear capacity of the beam. The beam was placed on pin and roller supports giving it a span of 53.3 ft. The point load was applied at 4.83 ft from the undamaged end, using a 400-kip hydraulic actuator. The test setup is shown in Figure 3.17. A spreader beam of a span of 3.2 ft between the bearing plates was used to distribute the load to the two webs of the box beam. Bridge Diagnostics Inc. (BDI) strain transducers were attached to the concrete surface to measure the strain along the depth of the girder at the point of loading. The deflection in the beam at the location of the point load was measured using a Wire Potentiometer. The applied load was measured using a load cell, placed between the hydraulic ram and the crosshead of the loading frame in the actuator. The load cell, wire potentiometer, and the BDI strain transducers were connected to a wireless BDI data acquisition system. The load was increased in 10 kip or 5 kip increments. Towards the end, when load did not increase significantly, the loading protocol was changed to deflection control. The deflection was recorded for each increase in load and the Load-Deflection response was plotted, which is shown in Figure 3.18.

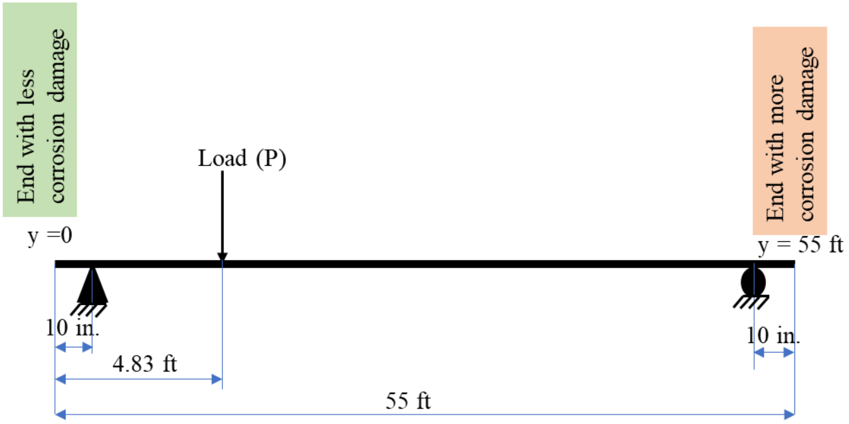

End with less corrosion damage
Load (P)
End with more corrosion damage
y =0
y = 55 ft
10 in.
4.83 ft
10 in.
55 ft

Figure 3.17. Test Setup for Test-1 on Aden Road Bridge Box Beam-3

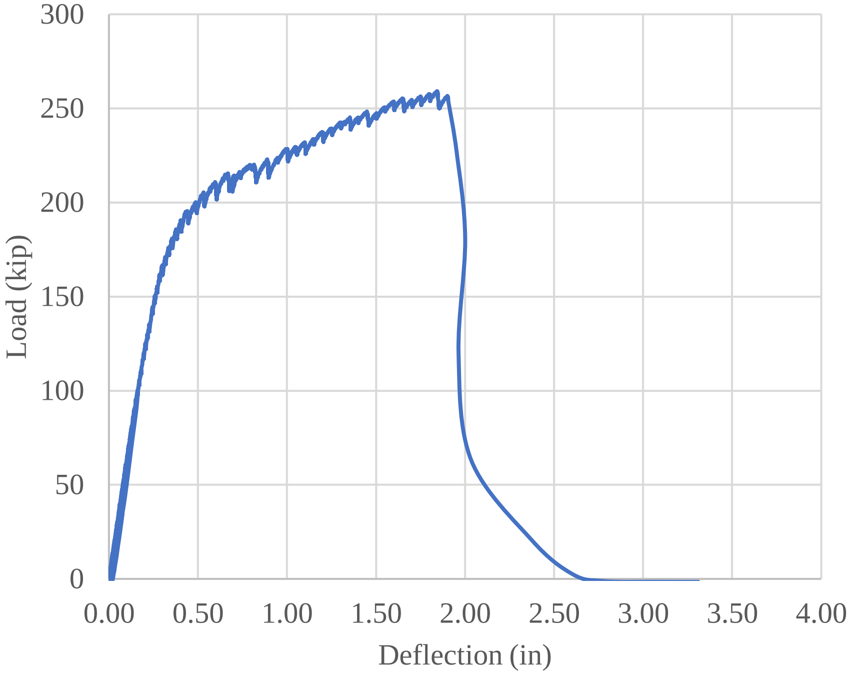


Figure 3.18. Load-Deflection Curve for Test-1 on Aden Road Bridge Box Beam-3

3.1.4.1.1.1. Failure Mode in Test-1

The beam endured a maximum load of 259.2 kip before it failed in a shear failure. The moment demand at the location of the point load, corresponding to this maximum load including self-weight moment was calculated to be 1029 kip-ft. The end of the beam after testing is shown in Figure 3.19.

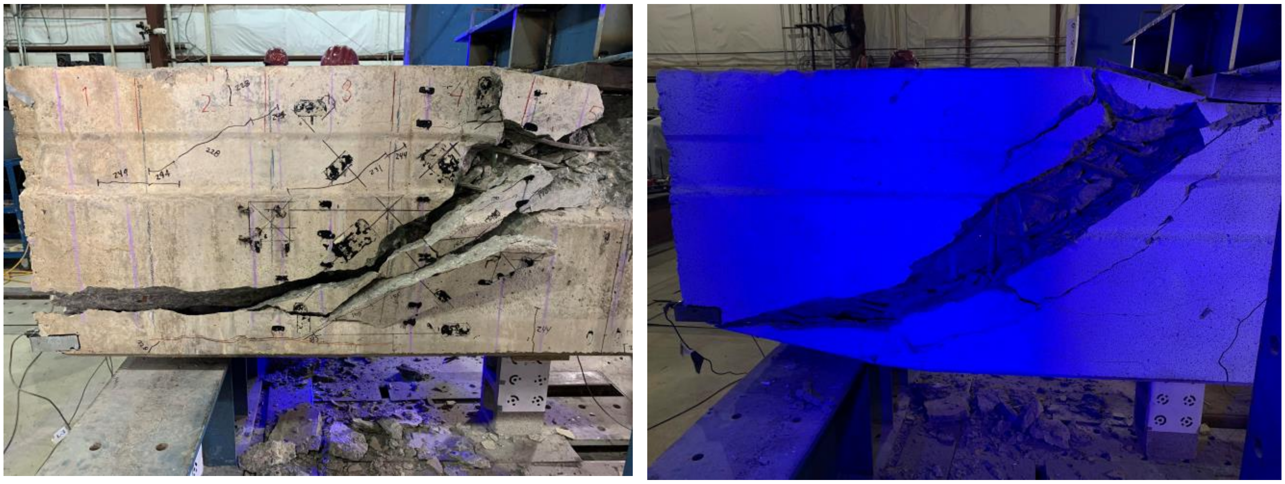


Figure 3.19. Shear Failure in Test-1 on Aden Road Bridge Box Beam-3

3.1.4.1.2. Test-2 on Aden Road Bridge Box Beam-3

In this test, one of the supports was moved past the end that was previously tested (in Test - 1) in shear (shown in Figure 9), creating an overhang region, so that this damage did not affect the testing end in Test-2. A roller support was provided at 9.42 ft from the end previously failed in shear, and the point load was applied at 50.42 ft from this end. The span of the beam in positive bending was 45 ft long. Spreader beam was used in this test also to distribute the load to the webs of the beam. The test setup is shown in Figure 3.20. Instrumentation and loading protocol used in this test were similar to those in the previous test. The Load-Deflection response observed during this test is shown in Figure 3.21.

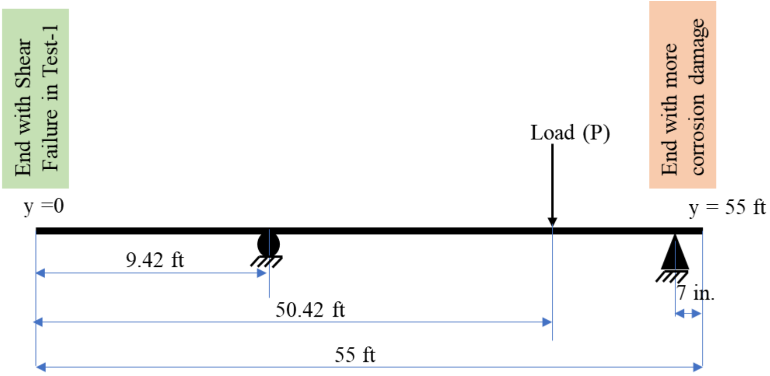

End with Shear Failure in Test-1
End with more corrosion damage
Load (P)
y =0
y = 55 ft
9.42 ft
50.42 ft
55 ft
7 in.

Figure 3.20. Test Setup in Test-2 on Aden Road Bridge Box Beam-3

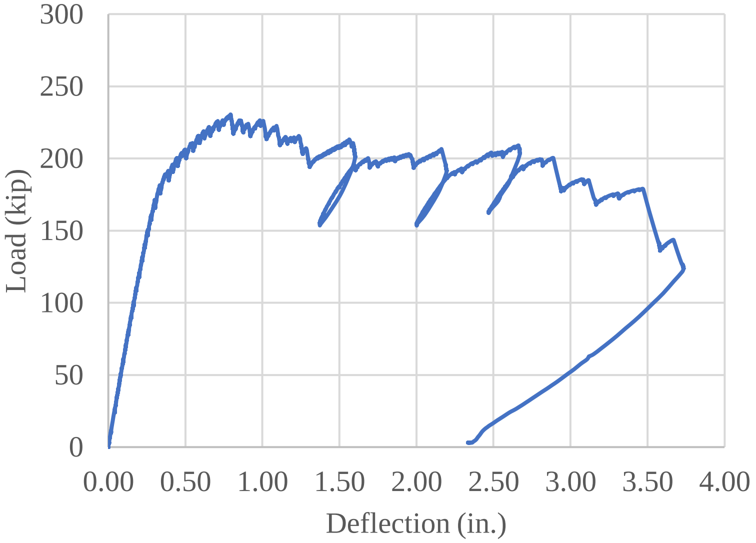


Figure 3.21. Load-Deflection Response in Test-2 on Aden Road Bridge Box Beam-3

3.1.4.1.2.1. Failure Mode in Test-2

Failure in this test did not occur at the section where the point load was applied. At the failure location, which was about 4.96 ft from the loading section, the prestressing strands started rupturing. The section at which strand rupture occurred, had a spalling in the middle with five exposed strands before the test, due to corrosion damage. This section was 45.46 ft away from the left end. This spalling and heavily corroded prestressing strands at this section altered the failure mode from a shear failure or a flexural failure at the section of loading to strand rupture at the spalled region. The Load-Deflection response showed strength and stiffness degradation as the strands fractured. The test was stopped at a point as the load continued to drop. The region at which the failure occurred is shown in Figure 3.22. The beam endured a maximum load of 230.5 kip before the load started dropping. The maximum moment demand at the failure section including self-weight moment was calculated to be 848.6 kip-ft. The maximum moment demand at the section where the point load was applied, including self-weight moment, was 896.3 kip-ft.

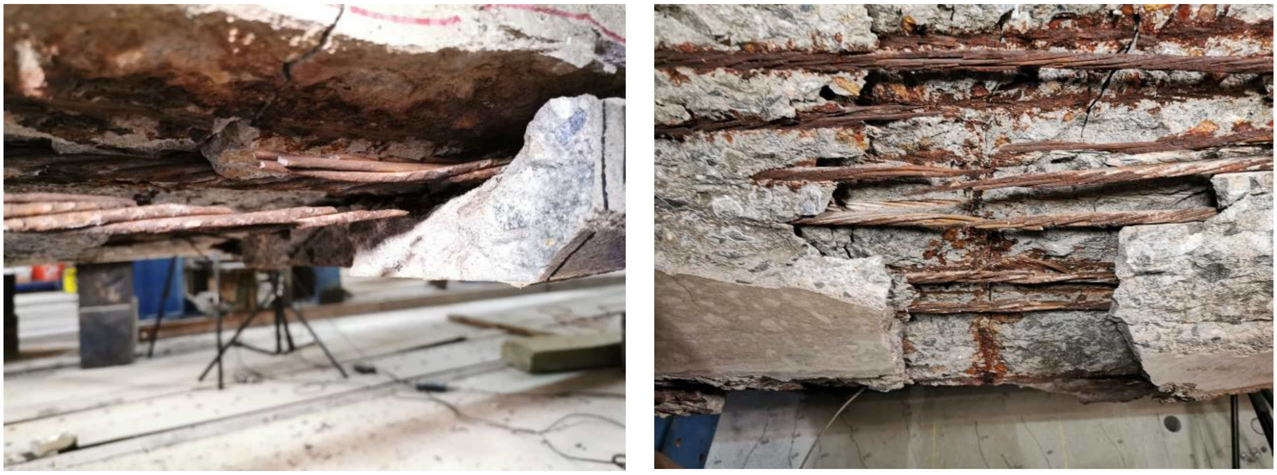


Figure 3.22. Strand Rupture in Test-2 on Aden Road Bridge Box Beam-3

3.1.4.2. Tests on Lesner Bridge I-Beam 9

3.1.4.2.1. Test-1 on Lesner Bridge I-Beam 9

The end of the beam with no corrosion damage was chosen for this test. The beam was placed on pin and roller supports with a span of 48.08 ft. The point load was applied at 8.83 ft from the undamaged end. The test setup is shown in Figure 3.23. The instrumentation on the beam and the loading protocol during the test were similar to those in the tests on the Box Beam as described in 3.1.4.1.1. The Load-Deflection response observed during this test is shown in Figure 3.24.

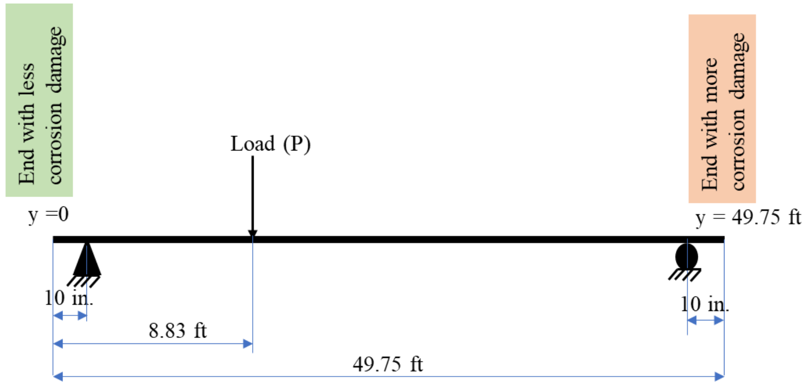

End with less corrosion damage
Load (P)
End with more corrosion damage
y =0
y = 49.75 ft
10 in.
10 in.
8.83 ft
49.75 ft

Figure 3.23. Test Setup in Test-1 on Lesner Bridge I-Beam 9

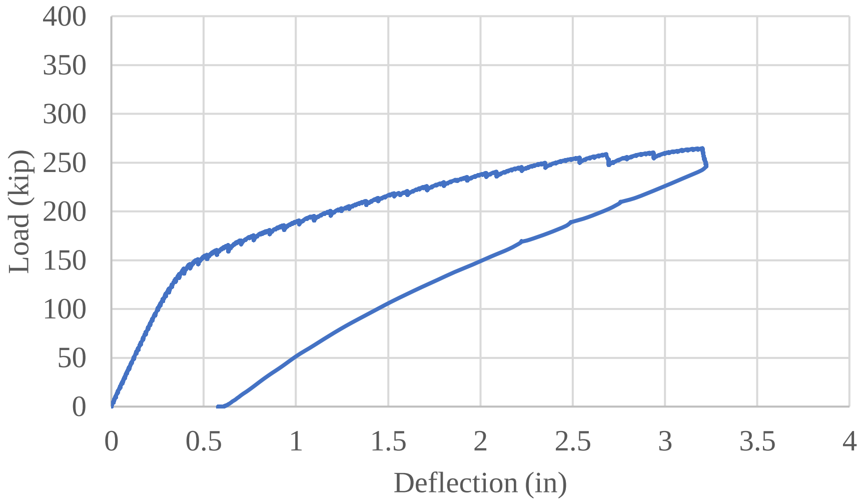


Figure 3.24. Load-Deflection Response in Test-1 on Lesner Bridge I-Beam 9

3.1.4.2.1.1. Failure Mode in Test-1

The failure observed in this end occurred due to flexural crushing in the deck of the beam, unlike the expected shear failure. The location of the point load was determined expecting a shear failure to avoid a direct compression failure between the point load and the support. But the shear span chosen caused a flexural failure instead. The failure in top slab (deck) is shown in Figure 3.25. The maximum load at which the failure occurred was 264.8 kip. Maximum moment demand due to the point load and self-weight of the beam was 1880 kip-ft.

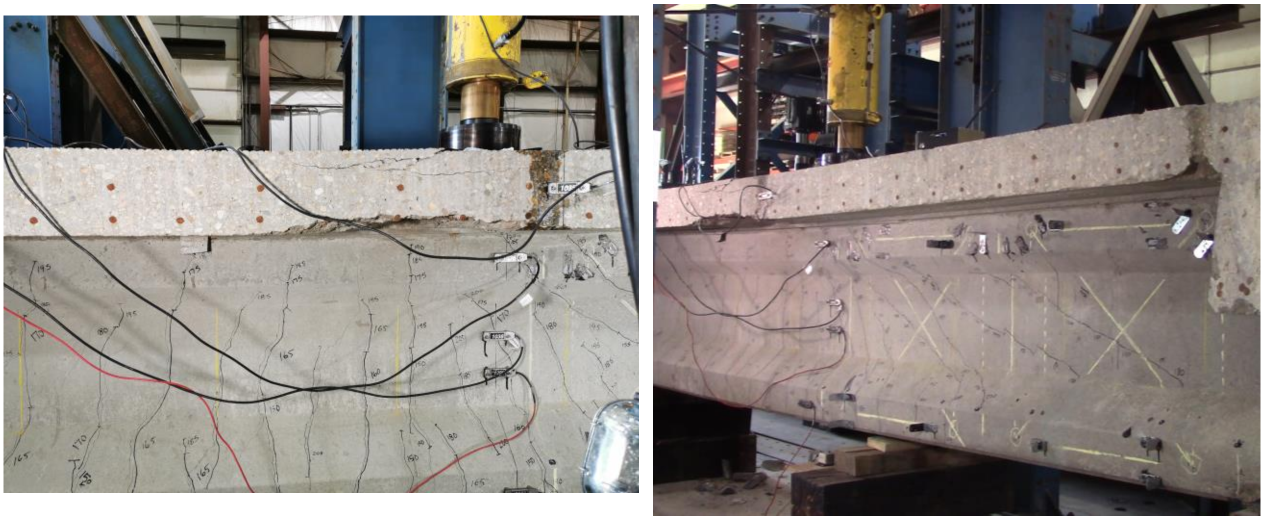


Figure 3.25. Crushing Failure in Top Slab in Test-1 on I-Beam 9

3.1.4.2.2. Test-2 on Lesner Bridge I-Beam 9

One of the supports was moved past the end that was previously tested in Test-1 (shown in Figure 3.25), creating an overhang region, so that the flexural damage did not affect the testing end in Test-2. A roller support was provided at 16.42 ft from the end tested in Test-1, and the point load was applied at 42.92 ft from this end. The span of the beam in positive bending was 32 ft long. The test setup is shown in Figure 3.26. Loading protocol and instrumentation used in this test were similar to those in the tests on the box beam. The Load-Deflection response observed during this test is shown in Figure 3.27.

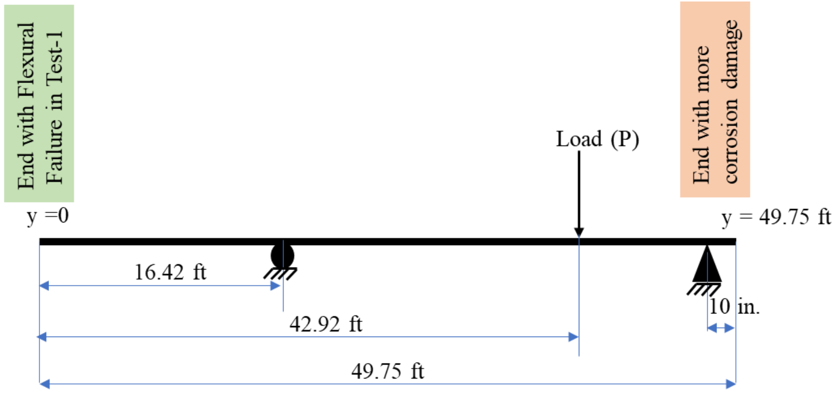

End with Flexural Failure in Test-1
End with more corrosion damage
Load (P)
y =0
y = 49.75 ft
16.42 ft
10 in.
42.92 ft
49.75 ft

Figure 3.26. Test Setup in Test-2 on Lesner Bridge I-Beam 9

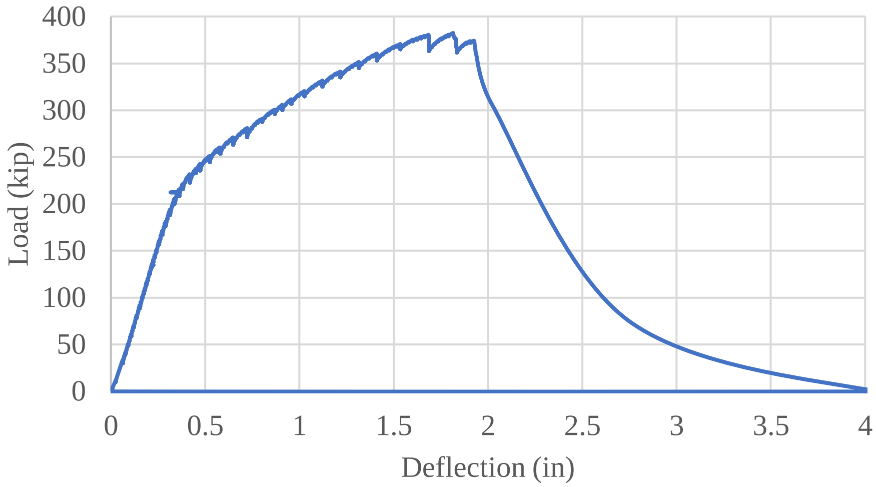


Figure 3.27. Load-Deflection Response in Test-2 on Lesner Bridge I-Beam 9

3.1.4.2.2.1. Failure Mode in Test-2

The beam at this end endured a maximum load of 382.4 kip before it failed in a shear failure. A steeply inclined crack that originated from the section of the point load was observed. This inclined crack did not extend all the way into the support but instead took the path adjacent to a spall and a crack (corrosion damage). The end of the beam after testing is shown in Figure 3.28. The moment demand corresponding to this maximum load including self-weight moment was calculated to be 1910 kip-ft.

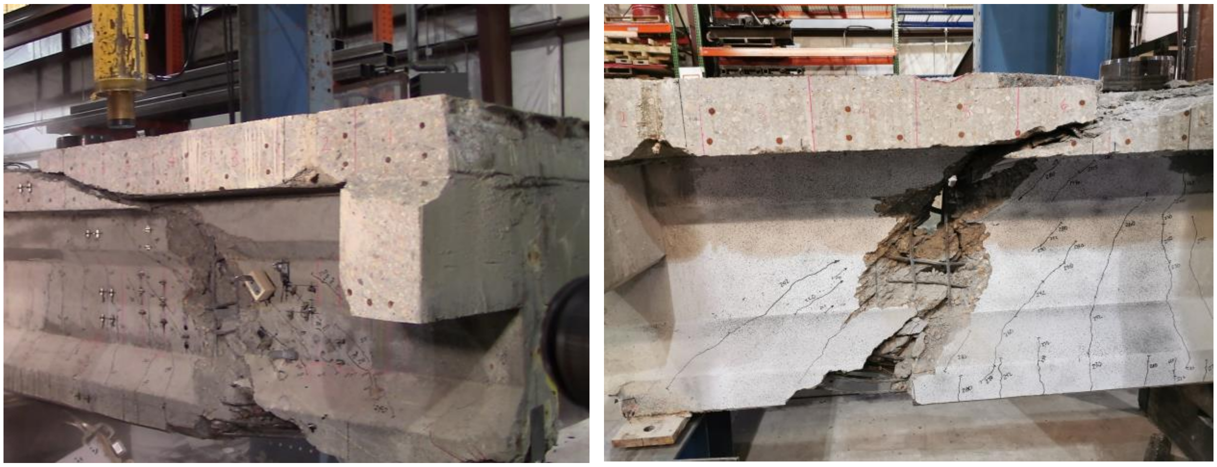


Figure 3.28. Shear Failure in Test-2 on I-Beam 9

3.1.5. Comparison of the Estimated Capacities with the Actual Capacities from the Tests

The moment capacities at the slice-sections along the length of the girders, calculated using strain compatibility analysis described in Section 3.1.3.7 were plotted for Aden Road Bridge Box Beam-3 and Lesner Bridge I-Beam 9. The calculated moment capacities at these sections were compared with the moment demands from four tests on these beams conducted by Al Rufaydah (2020). The experimental test setup, location of point load, and failure mode in each of these four tests are described in Section 3.1.4. The recommendations proposed by Naito et al. (2011) as discussed in Section 2.8 dealt only with box beams. However, in the current research, these recommendations were also used on an I-beam to validate the calculated moment capacities with the observations from the tests. Observations from the comparisons were used to understand the accuracy of the recommendations of Naito et al. (2011) and their applicability to I-beams. Furthermore, Strain Compatibility analysis was used to determine the strain in the bottom layer of strands in Test-2 on Aden Road Bridge Beam in which flexural failure occurred due to rupture of strands. The observations from these comparisons are discussed in Section 4.4.

3.2. List of Hardware and Software used in This Research

3.2.1. Resources Used in Implementing the Framework

Hardware and Software resources used in the current research and their specifications for each of the steps in the framework are listed in Table 3.5 and Table 3.6 across their purpose or use. The computational capacity of the machine is also mentioned in terms of the tasks successfully achieved by using the machine for each purpose. Agisoft Metashape was run on a computer whose specifications are mentioned in Table 3.5 for building low quality dense point clouds, surface meshes and texture mapping as described in Section 3.1.1. For trials with high quality dense point clouds, High-Performance Computing (HPC) Remote Computer facility at Virginia Tech was used with GPUs as mentioned in Table 3.5 using Python 3 module of the software. Programs for spall detection, slice extraction from the point cloud, clustering, dimensions of clusters etc. were written in Python and executed on the workstation computer mentioned in Table 3.5. Programs for training and testing the U-Net were in Python and were executed on the HPC remote computer with high GPU capability. More information on the purpose each hardware or software was used for, is mentioned in Table 3.5 and Table 3.6.

Table 3.5. Hardware Resources Used in this Research for Purposes as Mentioned

Hardware	Specifications	Purpose and Computational Capacity Extracted in Terms of the Task Successfully Executed
Camera	Nikon D7100 with AF Nikkor 20 mm f/2.8D lens	Image Acquisition for building 3D point clouds
LED lights	Bi-Color LED video light NEEWER- NL480 model with 28W power and Lumen: 3360 Lux/m	Image capture in indoor setting
Workstation (Computer)	32 GB RAM 6-Core Intel Xeon E5 Processor @3.5 GHz Dual Graphics Processing Units (GPUs)- AMD FirePro D700 of 6144 MB VRAM each	• Generation of Sparse Point Clouds (Alignment) with up to 2200 images in a single chunk • Dense Point Cloud Generation (Low Quality) with up to 2200 images in a single chunk • Mesh Generation, Texture Mapping (Texture Count =1, Texture Size = 12288) • Program for 3D level damage detection on Dense Point Clouds of the girders (voxel size = 0.15 to 0.2 and k = 300) • Other programs include clustering, slicing, cluster dimensions, etc.
Advanced Research Computing (ARC) Virginia Tech High-Performance Computing (HPC) Remote Computer (Memory, GPU requirements variable)	• Two NVIDIA P100 GPUs with 16 GB of memory each • CPU Memory of 60 GB	Training U-Net for about 10 epochs with a learning rate of 0.001 (Time taken was about four hours for a dataset of 22000 sub-images of 256 pixels x 256 pixels)
	• Three NVIDIA P100 GPUs with 16 GB of memory each • CPU Memory of 100 GB	Testing U-Net for about 1800 images (4200 x 3800 pixels) with a stride ratio of 1
	Two NVIDIA V100 (“Volta”) GPUs, 200 GB memory	Dense Point Cloud Generation, Surface Meshing and Texture Mapping on Agisoft Metashape version 1.6.2 (Python 3 Module)

Table 3.6. Software Resources Used in this Research for Purposes as Mentioned

Software	Specifications	Purpose
Agisoft Metashape Pro (Professional)	Version 1.6.2 GUI version and Python 3 Module	Point Cloud Generation (Alignment, Dense Point Cloud, Mesh Generation, Texture mapping)
CloudCompare	Version 2.10.2	Point Cloud and mesh processing include distance measurement

GNU Image Manipulation Program (GIMP)	Version 2.10.14	Ground truth labeling for the training dataset for U-Net network
Python (programming language)	3.7	Programs for 2D level damage detection, 3D level damage detection, slice extraction, clustering etc.

3.2.2. Time Taken for Implementation of the Proposed Framework for a Single Bridge Girder

In this section, the time taken for implementing the proposed framework to arrive at an estimate of the residual capacity from photogrammetry and artificial intelligence aided damage detection for a single bridge girder used in the current research is discussed. Time taken for image capture and building the point cloud, texture mapping, etc. are discussed in Section 4.1.1. It is also mentioned that images captured in the current work resulted in a high overlap between consecutive images which means that the total number of images captured was on the higher side than required. However, this also resulted in a large number of points in a low quality dense cloud, which is desirable. The total number of images captured affects the time taken for the steps such as crack detection, etc. Time taken for implementing all the steps discussed in this chapter for one beam are listed in Table 3.7. The hardware and software resources for this implementation are listed in Table 3.5 and Table 3.6.

Table 3.7. Time Taken for Implementation of the Proposed Framework for One Bridge Girder

Step of the Framework	**Time Taken, hours**
Manual Image Acquisition (assuming outdoor setting) using a handheld camera	5
Detection of crack-like features using U-Net (for 1800 images)	26
Alignment of Images, Building Dense Point Cloud (Low Quality) and Textured Mesh (for 1800 images)	3
Scaling to real-world dimensions and aligning the dense point cloud and textured mesh with the coordinate axes	0.5
Extracting cracks from the dense cloud into a new point cloud and cleaning	0.5
Detection of Spalls using point normals approach and cleaning the output	0.5
Clustering and dimension approximation of the clusters	0.5
Slicing the dense point cloud and preparing the overlays (10 slices)	1.5
Arriving at the reduced area of cross-section of strands (10 slice-locations)	2.5
Moment Capacity Calculation	0.5
TOTAL	40.5

As mentioned in Table 3.7, it took about 40 to 45 hours for implementing the proposed framework for a single beam. It can be observed that the total number of images captured significantly influences the total time taken for implementing this framework. In addition to these steps, it took about four hours for training the U-Net for 10 epochs with a learning rate of 0.001 using the dataset described in Section 3.1.2.1.1 on the High-Performance Computing (HPC) remote computer mentioned in Table 3.5. In this framework, if UAVs were used for image capture, it would add a significant amount of time for path planning, stabilizing, and dealing with disturbances such as wind, etc. More research is required to explore using UAVs in this framework.

The time taken for two people to develop damage map of a girder manually by physically inspecting the beam and documenting each crack and spall, taking manual measurements on the beam with a tape measure, and developing the CAD drawing of the damage maps, was about three hours. After this step, to calculate the reduced area of strands at the desired cross-sections, it would take longer to find the strands affected due to each of the documented cracks and spalls using eyeballing. It can be assumed that it would take about three hours for calculating the flexural capacity of the girder at multiple cross-sections using this manual approach. The total time taken for two people for this manual approach which requires physical close-in access to the girder can be estimated as six hours. When compared to this, the approach proposed in this research would take significantly longer to arrive at an estimate of the residual capacity. However, the main advantage of the approach proposed in the current research is that it does not require physical access to the girder.

CHAPTER 4 RESULTS AND ANALYSIS

In Chapter – 3, the procedure followed for each step in the framework is described. Required parameters for each step are discussed and some examples for the output of some steps were discussed. In the current chapter, the results of each step are discussed in detail. The effect of the parameters on the obtained results are presented, and the reasoning for choosing the optimal parameters to achieve the goals of the current work are discussed. Observations are made from the results and analyzed to question the effectiveness of the methodologies followed. The results and analyses are presented for each step of the framework in the same order as that followed in Chapter - 3.

4.1. Digital State Model (DSM) Reconstruction of the Bridge Girders

In this research, 3D point clouds were used as a visual tool to convey the information about corrosion damage present on each bridge girder. The point clouds can be built without physical access to the girders using UAVs to capture the images. However, to limit the scope of the current work, the pictures of the girders were manually captured as described in 3.1.1.1. The images are used to generate the point clouds which are then meshed, and texture mapped to generate digital State Models of the girders. The procedure and the steps involved in the development of the DSMs of the bridge girders using Agisoft Metashape are explained in Section 3.1.1. This section discusses the effect of the parameters chosen for each step and the results of each step implemented on the four bridge girders used in this research mentioned in Table 3.1.

4.1.1. 3D Point Clouds of the Girders

4.1.1.1. Image Acquisition

This section discusses the physical and computational resources that went into building 3D point clouds of the girders, the effect of the parameters chosen for each step, and the lessons learned. The first task in this process was to collect the images set in a way that the software could make use of them for the point cloud reconstruction. The number of images captured for each beam depended on the access available around the beam. More access was available for the beams that were outside which resulted in fewer images. The number of images captured for each beam for building point clouds, setting in which the images were captured and time taken, are shown in Table 4.1.

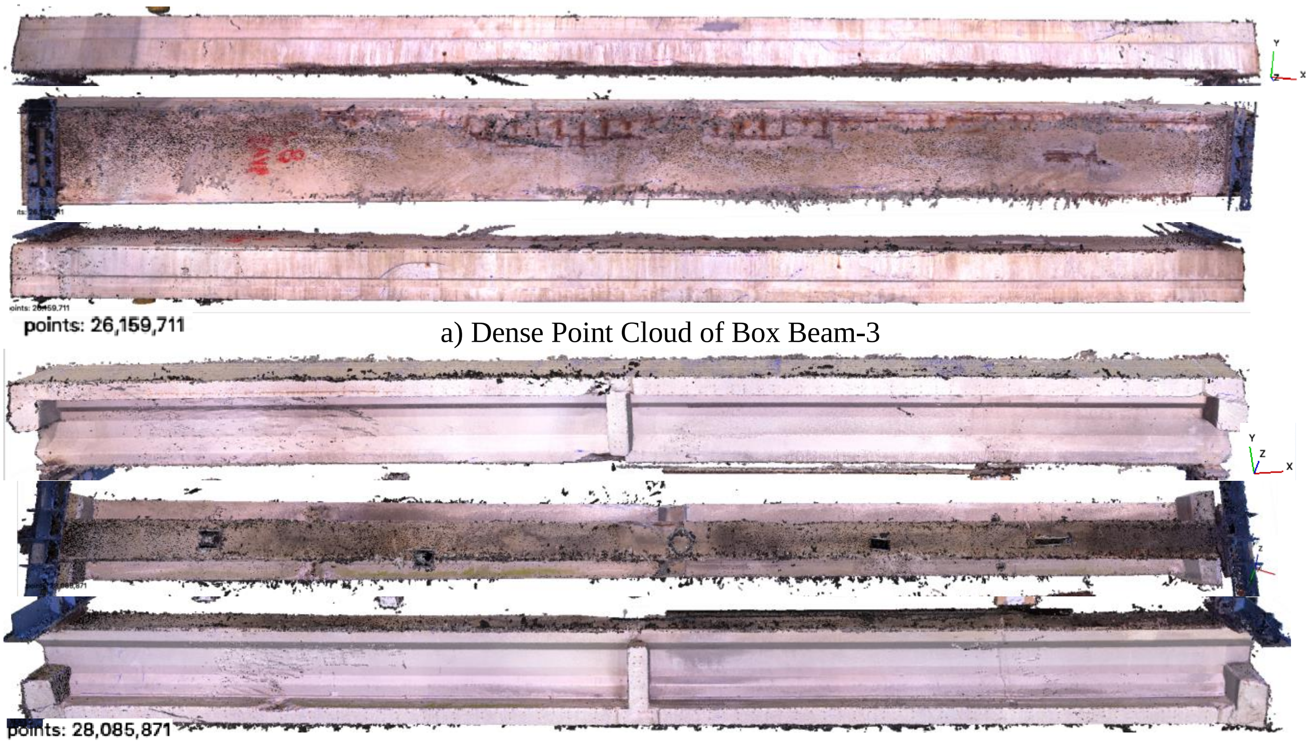
Table 4.1. Number of Images and Type of Setting for Image Capture for Building Point Clouds of the Bridge Girders

Name of the Girder	Physical Setting used for Image Capture	Number of Images Captured	Access Available around the girder / Depth of Field for Image Capture	Time Taken for Manual Image Capture
Aden Road Bridge Box Beam - 3	Indoors (Laboratory) with LED lights	1803	4 ft to 5 ft	About 8 hours
Lesner Bridge I-Beam 9	Indoors (Laboratory) with LED lights	2216		
Lesner Bridge I-Beam 5	Outdoors	988	about 7 ft	About 4 hours to 5 hours
Lesner Bridge I-Beam 1	Outdoors	1113		

As the access available to capture images inside the laboratory was less, it needed more images to cover the entire beam. The entire image capture process took about four to eight hours depending on the access available. This total time reported includes the time needed to move around the beam in small increments, to move the LED lights for multiple image capture locations, to move the ladder, and for the camera to autofocus and stabilize at every image capture frame. A report output by Agisoft Metashape showing the details about each step in the process of building the point clouds and texturing showed a graph with the number of images covering each area on the point cloud. For the point clouds of all four girders, the numbers reported in these graphs showed that almost all regions on the point cloud were visible in more than nine images, which is on the high side since the software needs each point on the beam to be in at least two images to be captured on the point cloud. So, capturing each location in more than nine cameras was found to be an overkill. At each location, five to six images were captured to cover the depth of the beam including the top of the beam and the bottom of the beam. This could be limited to fewer images to reduce the side overlap to the recommended side overlap of 60%. The forward overlap between longitudinal image capture locations was about 75% which seemed to be appropriate.

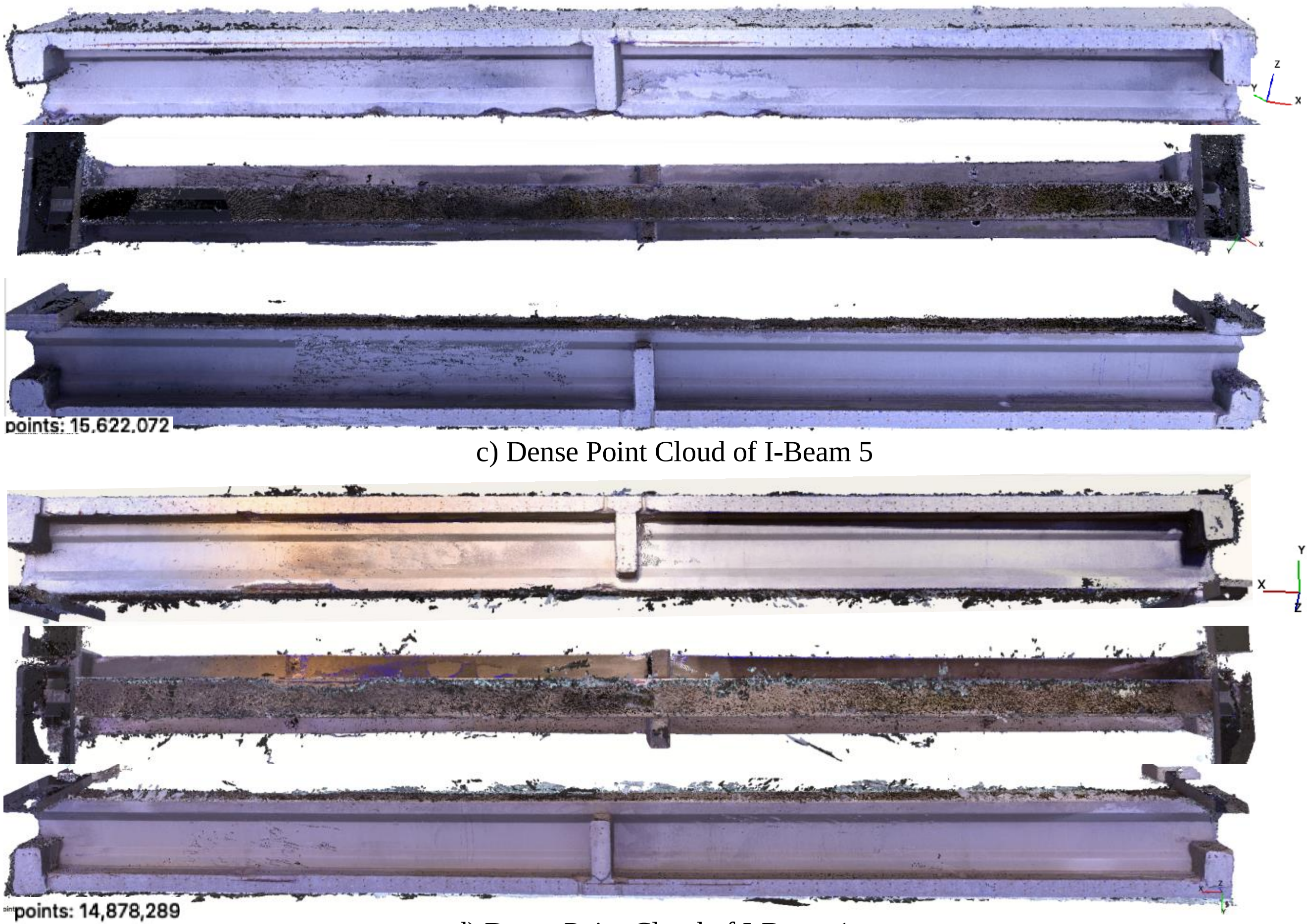
4.1.1.2. Dense Point Clouds and Textured Mesh Models of the Bridge Girders

The procedure for alignment of images and dense point cloud generation steps are described in Sections 3.1.1.2 and 3.1.1.3. The alignment step generates a sparse point cloud which is further densified in the dense point cloud generation step. The parameters used in Agisoft Metashape for these steps are shown in Figure 3.5. Building a dense point cloud involves a quality parameter. Trials were conducted on the computer whose specifications are mentioned in Section 3.2 to determine the parameters that satisfy the requirements of the current research. It was found that, for the number of images mentioned in Table 4.1, a ‘high’ quality parameter for dense cloud required high computational capacity. The high quality parameter used for dense cloud generation for Box Beam-3 with all images in a single chunk caused the computer to run out of memory. For other beams with fewer images, even if the computer successfully built the dense clouds, the next step, surface mesh generation ran out of memory. To deal with these problems, a ‘low’ quality parameter was used for dense point cloud generation in this research. For the purposes of this research, a low quality parameter was found to be satisfactory even it downscaled the original images by a factor of eight. The sparse point cloud for Lesner Bridge I-Beam 5 is shown in Figure 3.6. Dense point clouds of all four girders used in this research, built using a low quality parameter on Agisoft Metashape are shown in Figure 4.1.



a) Dense Point Cloud of Box Beam-3

b) Dense Point Cloud of I-Beam 9



c) Dense Point Cloud of I-Beam 5

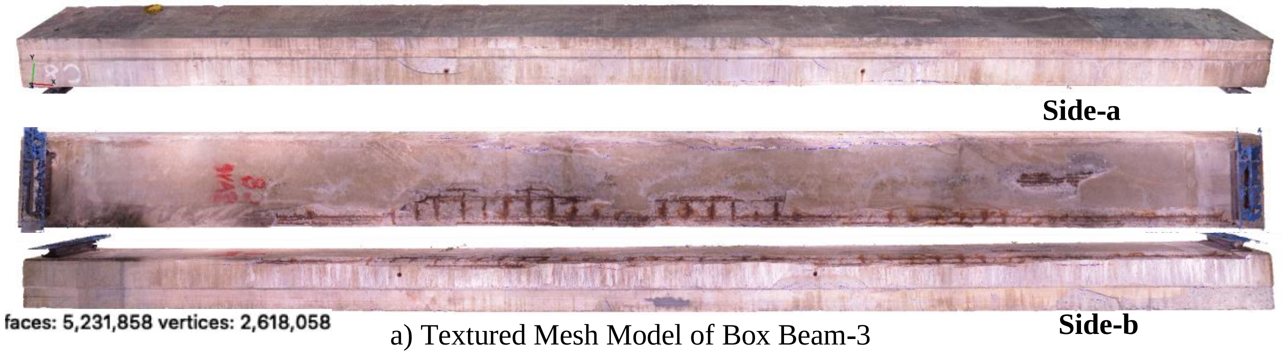
d) Dense Point Cloud of I-Beam 1

Figure 4.1. Dense Point Clouds of the Bridge Girders Reconstructed using Agisoft Metashape with a Low Quality Parameter

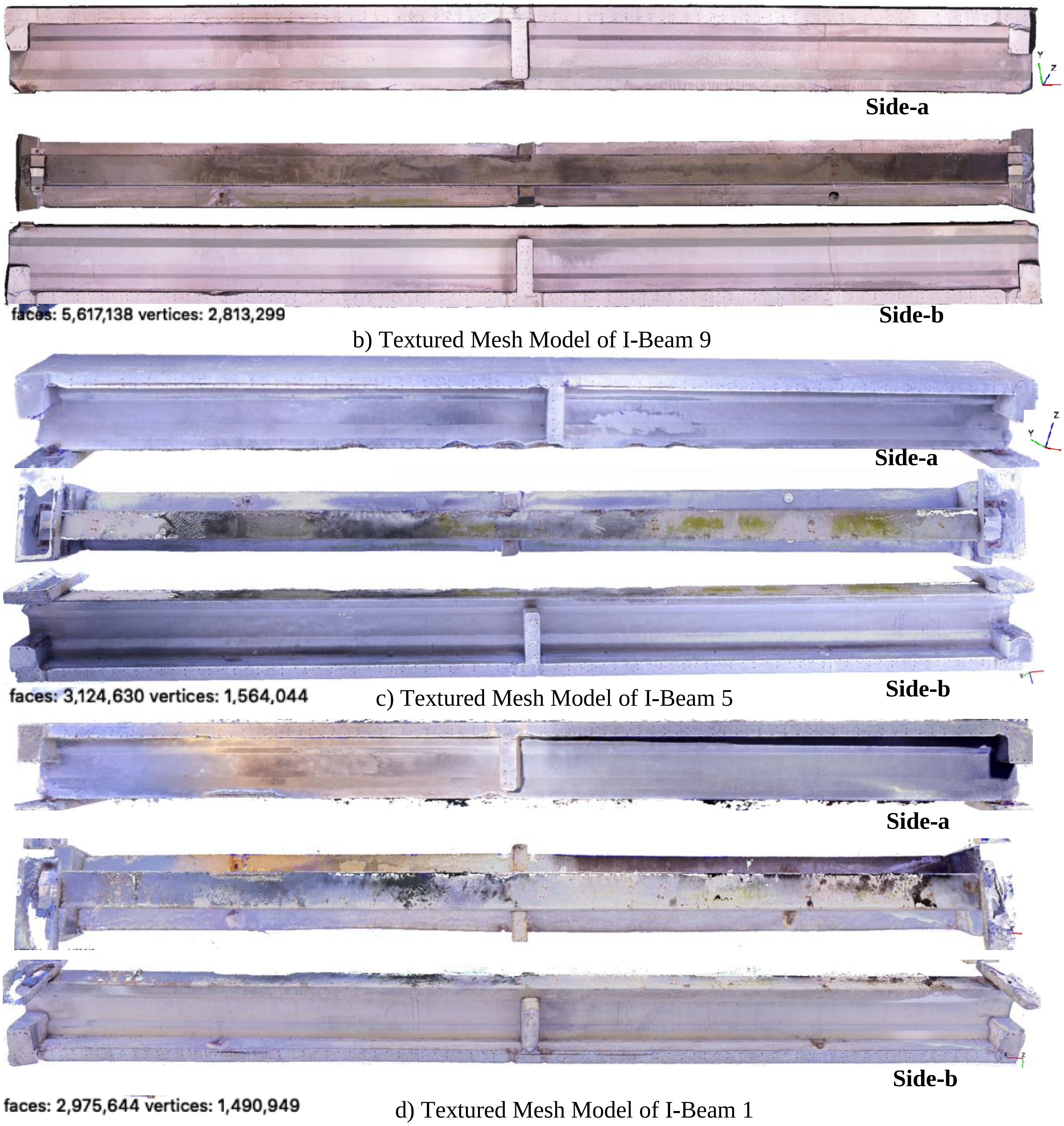
The number of points in the low quality dense point clouds of Box Beam-3, I-Beam 9, I-Beam 5, and I-Beam 1 are mentioned in Figure 4.1a, Figure 4.1b, Figure 4.1c, and Figure 4.1d, respectively. It can be observed that the number of points changed with the number of input images (mentioned in Table 4.1). Other aspects that affect the dense point clouds included the availability of light. On the point clouds of the beams located outside, the sides exposed to the sun were more illuminated than the other sides. For I-Beam 1, the side exposed to the sun had different illuminated regions because the time of the day changed from the right side to the left during the image capture process. The images on the right were too bright because of the illumination from the sun and the images on the left were too dark. After the sparse point cloud was generated, the images that were too dark were replaced with brightness corrected images. The improved model built from the brightness corrected images is shown in Figure 4.1d.

Time taken for aligning the pictures and building the dense cloud step (low quality) for Box Beam-3 are mentioned in Table 4.2. The dense point cloud in low quality was built in about an hour whereas trials conducted for building the dense cloud using high quality parameter caused the computer to run out of memory (also indicated in Table 4.2) as all images were placed in a single chunk. To deal with this problem, the images could be divided into multiple chunks processed separately and merged later. Instead, low quality dense clouds were used because they gave satisfactory results for the objectives of the current research. The time taken on the workstation computer to build the dense clouds of low quality were significantly lower than the time taken for building high quality dense clouds. A trial conducted on the sparse point cloud of I-Beam 5 to build a high quality dense cloud, took about ten hours to complete as opposed to only about forty minutes taken to build a low quality dense cloud. The high quality dense cloud for I-Beam 5 had two hundred fifty-seven million points, whereas the dense cloud of low quality had only fifteen million points. Coordinate axes shown below the beams in Figure 4.1 indicate that the point clouds at this stage were not aligned with the textured mesh models of the bridge girders. If a UAV is used in the future research for image capture, trials may be conducted to determine the optimal number of images required to build a satisfactory point cloud. If lesser number of images are captured with appropriate overlap, Medium or High quality parameters can be used in Agisoft Metashape for dense point cloud reconstruction to get higher number of points in the dense clouds.

Surface meshes were generated on the dense point clouds and textured with images on which cracks were highlighted as explained in Section 3.1.1.5. The recommended parameters for these steps are presented in Figure 3.5. Results of crack detection on the images of the bridge girders using a U-Net as described in Section 3.1.2.1. Textured mesh models of all four bridge girders are shown in Figure 4.2. Sides a and b are labeled for convenience in further steps.



a) Textured Mesh Model of Box Beam-3


Y
Z
Side-a
faces: 5,617,138 vertices: 2,813,299
Side-b
Z
Y
Side-a
faces: 3,124,630 vertices: 1,564,044
Side-b
Side-a
Side-b
faces: 2,975,644 vertices: 1,490,949

b) Textured Mesh Model of I-Beam 9

c) Textured Mesh Model of I-Beam 5

d) Textured Mesh Model of I-Beam 1

Figure 4.2. Textured Mesh Models of the Bridge Girders Reconstructed Using Agisoft Metashape

Texture count and texture size indicated the number atlas pages and the size of each atlas page generated. Using one page of size 2000 x 2000 pixels was observed to give blurry texture to the girders. Increasing the size of the pages resulted in clearer and sharper texture. It was observed that using three atlas pages of size 4096 x 4096 gave a similar quality of texture as one page of size 12288 x 12288 pixels. So, either set of parameters may be used.

It can be observed from the textured mesh models of Box Beam-3 and I-Beam 9 shown in Figure 4.2a and Figure 4.2b, built from images captured in a laboratory setting, that the bottom surfaces are well captured owing to the illumination from LED lights. The bottom surface of I-Beam 1 had some dark regions owing to a lack of lighting at the end of the day. The texture built also had holes at one location which were covered by false texture. Hence, it was very important to have good lighting to obtain good quality point clouds and textured mesh models. Access available underneath the beams was only about 3 ft for both indoor and outdoor image capture settings. In actual bridges, the access available underneath the girders would be much greater which would facilitate better illumination. Lighting is a major factor to be considered when exploring the use of this method on actual bridges. Also, the time taken to capture the images has to be reduced in order to maintain stable natural lighting conditions during image capture. As explained in Section 4.1.1.1, large overlap maintained in the images captured in the current research increased the total number of images. It is possible to get complete point clouds of the girders with reduced overlap (minimum 60%) resulting in a reduced total number of images captured. To get larger number of points, a medium or high quality parameter can be used for dense cloud reconstruction step.

The textured mesh models shown in Figure 4.2 had cracks highlighted on them. Crack detection on 2D images and on the texture are discussed in Section 4.2.1. Time taken for surface mesh generation and texture mapping steps for Box Beam-3 are shown in Table 4.2. In total, it took about three hours for building a textured model of Box Beam-3 from a low quality dense point cloud (1803 images in a single chunk) on Agisoft Metashape, as mentioned in Table 4.2. This time does not include the time taken by the Neural Network to detect crack-like features on the 2D images of girders, which is discussed in Section 4.2.1.

Table 4.2. Time Taken for Each Step for Building the Textured Mesh Models from Point Clouds of Box Beam-3 on Agisoft Metashape

Name of the Step on Agisoft Metashape	Parameters	Specifications of the Machine	Time Taken, hours
Align Photos (1803 Images in a single chunk)	Accuracy: High, Key Point Limit: 40000, Tie Point Limit: 4000	Computer with 32 GB RAM and dual GPUs of 6144 MB each, see Section 3.2	1
Build Dense Point Cloud	Quality: Low		1
	Quality: High		Out of memory

Build Mesh	Source data: Dense cloud (Low Quality), Surface Type: Arbitrary, Depth maps quality: Ultra high		0.3
Build Texture	Texture Count: 1, Texture Size: 12288		0.7
Total time taken with Low Quality Dense Point Cloud			3 hours

4.2. Development of 3D Damage Maps of the Girders

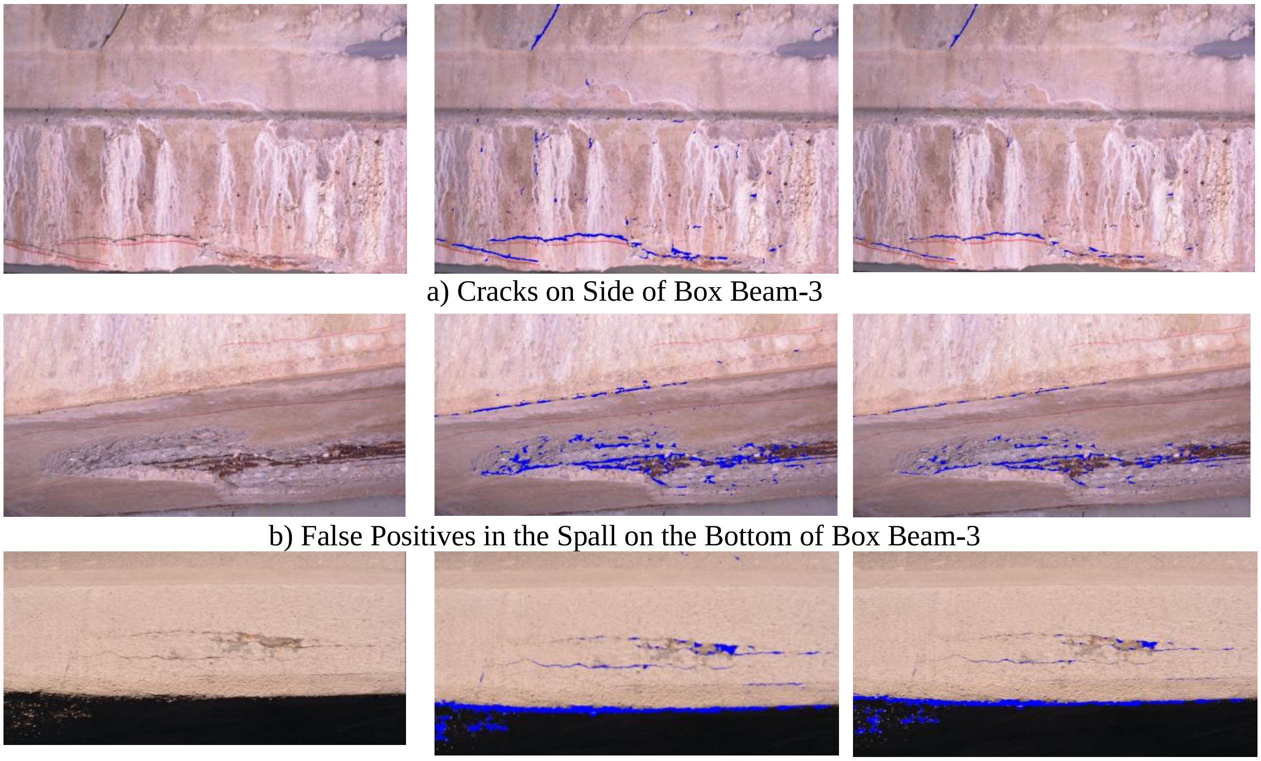
In the current work, the textured mesh models of the bridge girders overlaid with cracks and spalls point clouds were used as the 3D damage maps. The first step towards this goal was to detect cracks on 2D images that were used to build the 3D point clouds of the girders. Then, the cracks detected were transferred to 3D point clouds using texture mapping as explained in Section 3.1.1.5. Detection of crack-like features on the images is discussed in this section.

4.2.1. Crack Detection on 2D Images using U-Net (Fully Convolutional Network)

As explained in Section 3.1.2.1, a Fully Convolutional Network with a U-Net architecture was trained using labeled images collected from opensource datasets. Qualitative performance evaluation was done to arrive at appropriate parameters for the testing phase. Parameters chosen for the training process are mentioned in Section 3.1.2.1.2. The model was trained for 10 epochs and the value of loss for training and validation datasets was recorded for every epoch. The model which gave minimum validation loss was used for the testing phase to avoid overfitting. Overfitting leads to the model memorizing the training dataset and performs poorly when presented with new testing images.

Parameters in the testing process included a threshold value, which was used to cut off some noise or false positives from the detection as explained in Section 3.1.2.1.1 and a stride ratio to feed the large images of size 4800 x 3200 pixels as patches of size 256 x 256 pixels on which the model was trained on. These parameters were determined using trial and error. Some of the images used to build the point clouds and their predicted damage from U-Net are shown in Figure 4.3. The effect of the threshold parameter was studied for values of 0.3 and 0.5. It can be observed that the model suffers from false positives (Figure 4.3a). In Figure 4.3a, more reduction in background noise can be observed for a threshold value of 0.5 when compared to a threshold value of 0.3. Major sources of noise included portions of edges of the beams (Figure 4.3b and Figure

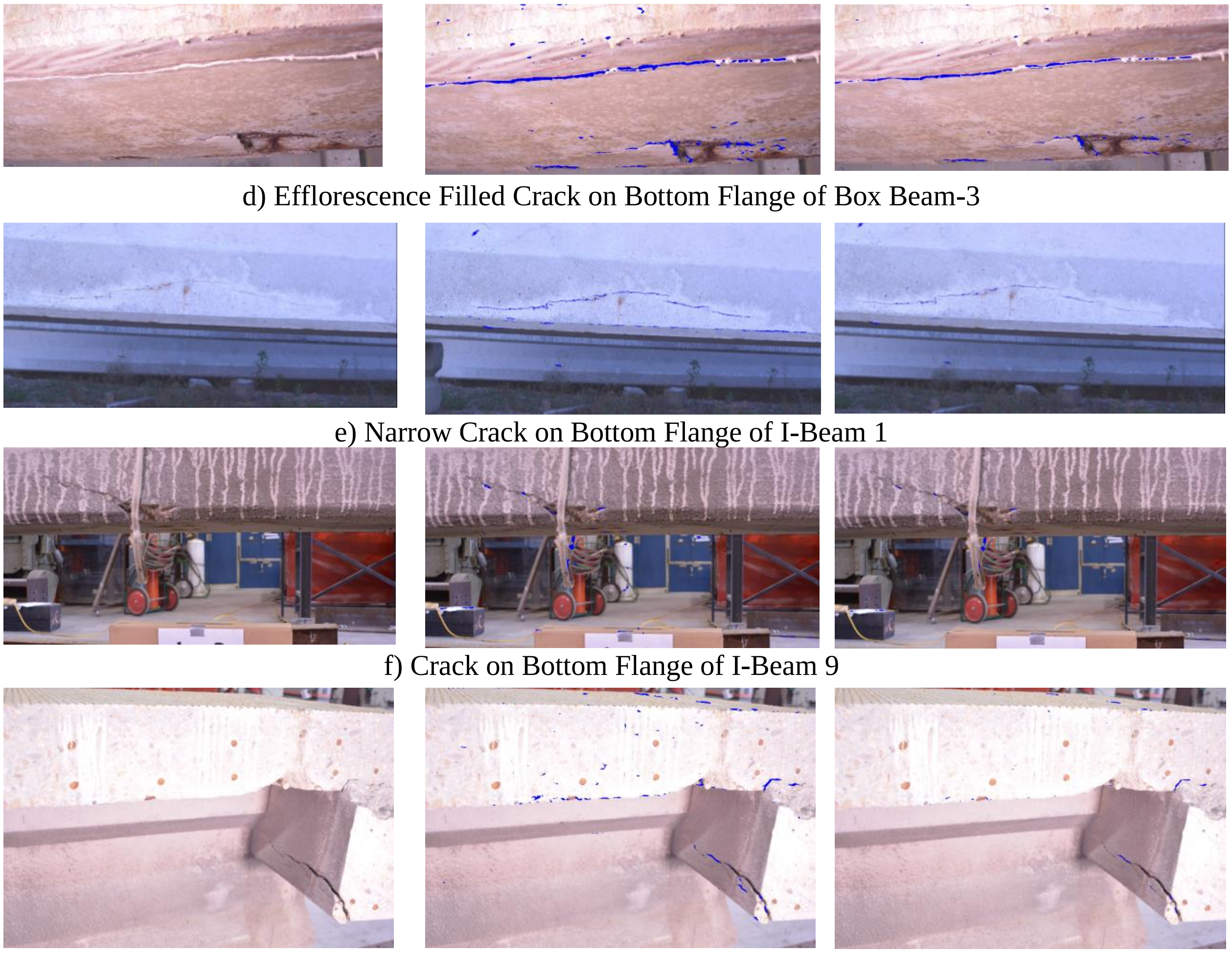
4.3c), portions of spalls (Figure 4.3b), and other crack-like features. Cracks in Figure 4.3f and Figure 4.3g were missed to a large extent. The model, in general, did not perform well for very wide cracks such as in Figure 4.3g. As explained in Section 3.1.2.1.1, six images from Box Beam-3 dataset were divided into patches of 256 x 256 and data augmentation was done to increase the number of sub-images in the dataset and to improve the performance of the model in detecting the cracks filled with efflorescence (a labeled image is shown in Figure 3.7). The improved performance of the model in detecting such cracks after adding these sub-images to the dataset can be observed in Figure 4.3d. It can be observed in Figure 4.3b that very narrow cracks on the bottom surface of the beam (highlighted with a red marker) with efflorescence were only partly detected. When a threshold value of 0.5 was used, some portions of the cracks that were detected with a threshold of 0.3, went undetected. This can be observed in Figure 4.3c. In order not to miss portions of cracks, a threshold value of 0.3 to 0.4 was used in the testing phase in the current research. More noise was observed on darker images and more cracks were missed from blurry images. The effect of stride ratio is demonstrated in Figure 4.4.



a) Cracks on Side of Box Beam-3

b) False Positives in the Spall on the Bottom of Box Beam-3

c) Narrow Cracks on Bottom Flange of I-Beam 1



d) Efflorescence Filled Crack on Bottom Flange of Box Beam-3

e) Narrow Crack on Bottom Flange of I-Beam 1

f) Crack on Bottom Flange of I-Beam 9

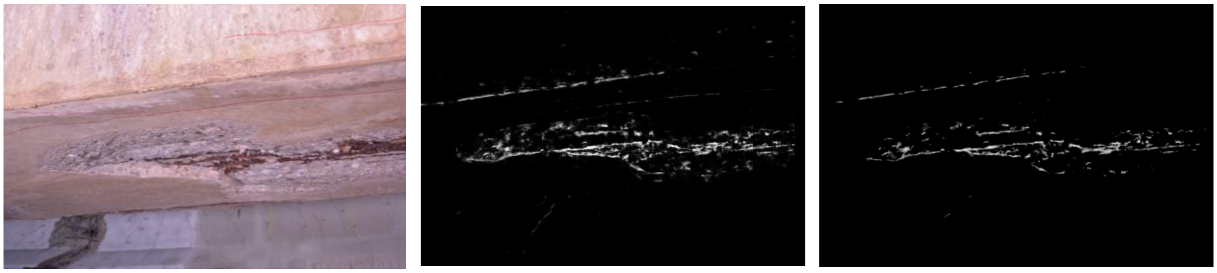
g) Wide Crack on Diaphragm of I-Beam 9

Original Images

Threshold = 0.3
Stride Ratio = 1

Threshold = 0.5
Stride Ratio = 1

Figure 4.3. Output of Crack Detection Using U-Net Trained for 10 Epochs and Tested with Threshold and Stride Ratio as Specified

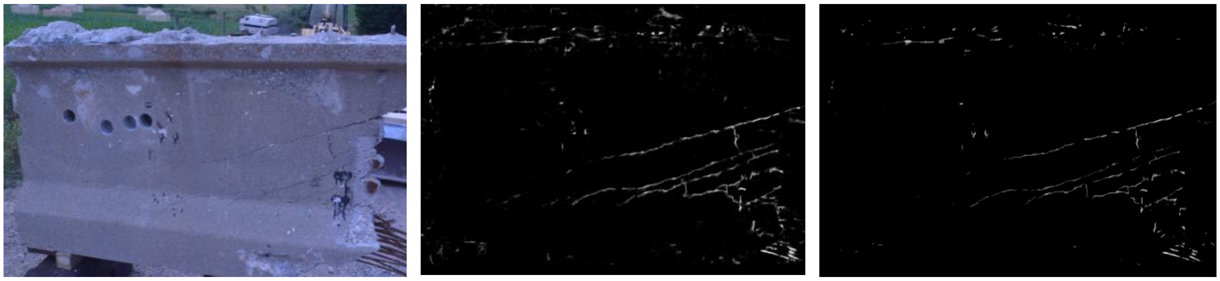


Original Image

Threshold = 0.3
Stride Ratio = 1

Threshold = 0.3
Stride Ratio = 0.1

a) Spall on the Bottom of Box Beam-3



Original Image

Threshold = 0.3
Stride Ratio = 1

Threshold = 0.3
Stride Ratio = 0.1

b) Beam Chunk Used for Trial

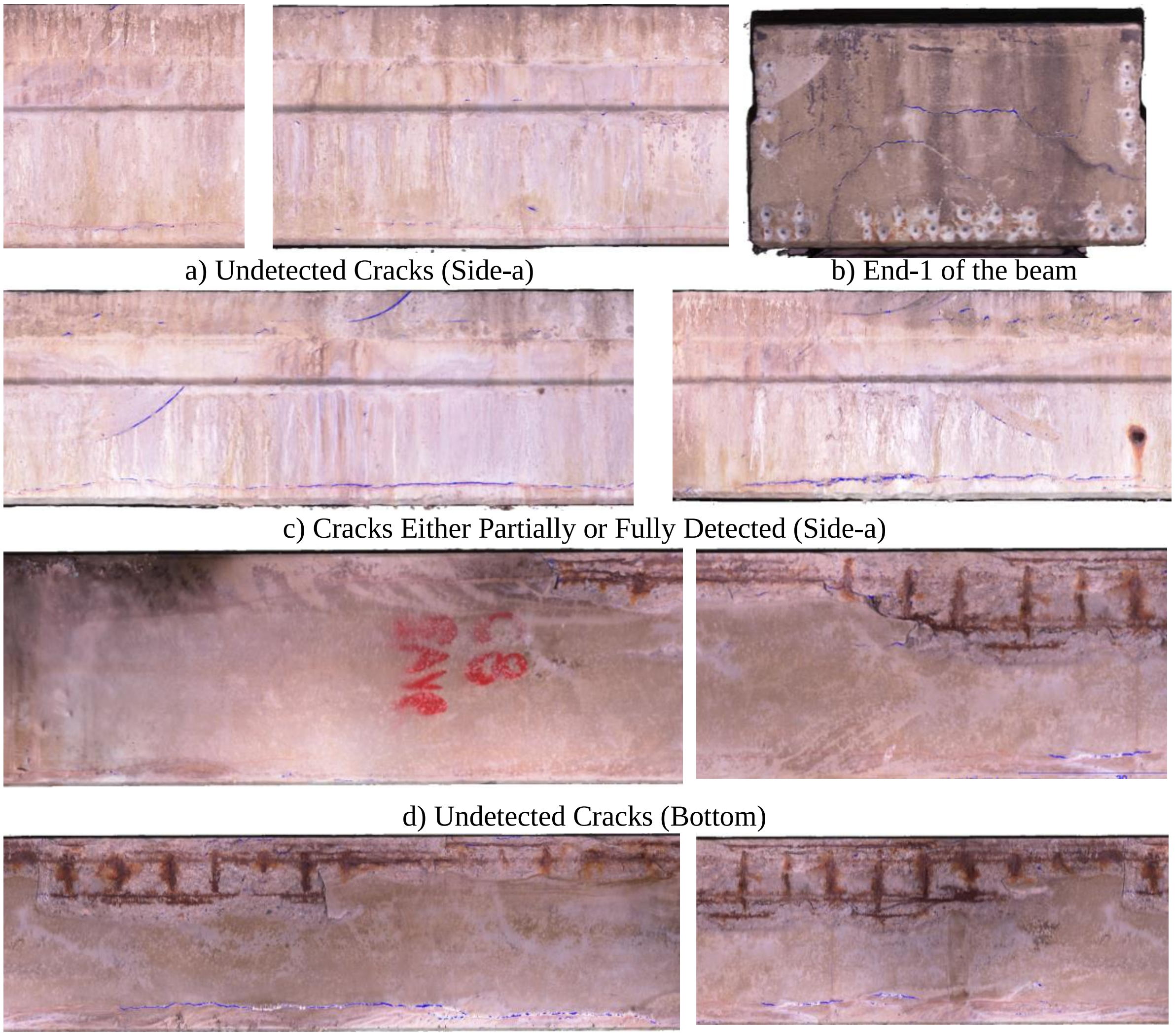
Figure 4.4. Effect of Stride Ratio during Testing Phase on Crack Detection

A stride ratio of 0.1 was observed to give a slightly more reduction in background noise when compared to a stride ratio of 1, as can be observed in Figure 4.4. A narrow crack observed on the bottom of the Box Beam that was slightly detected when a stride ratio of 1 was chosen, was omitted when a stride ratio of 0.1 was chosen. So, a stride value of 1 was adopted and the noise was cleaned after transferring the cracks on to 3D point clouds.

In the current research, a quantitative evaluation of the performance of the crack detection model (as explained in Section 2.4.1.3.1.1) was not done. To determine the most appropriate parameters that work for the current dataset such as the threshold parameter, it is important to evaluate the performance of the model quantitatively. To do this, labeled images from the current dataset are required to compare the prediction to the ground truth label and determine the true positives, false positives, true negatives, and false negatives. These labeled images should not be used to train the model. Performance metrics such as accuracy, precision, recall, Intersection over Union (IoU) or Jaccard Index, etc. can be evaluated in further research. These metrics are introduced in Section 2.4.1.3. To determine the best parameters for the model, the variation of the parameters can be plotted against IoU values, and the parameters that result in a trade-off between the false positives and the false negatives in the crack detection can be determined.

4.2.2. Transfer of Cracks Detected to the Surface Mesh Models on the 3D Point Clouds

In this section, a qualitative evaluation of cracks detected on the textured mesh models of the girders is presented. This is achieved by texture mapping. Parameters used and the output of the texturing step are shown in Section 4.1.1.2. Textured mesh models of all four girders are shown in Figure 4.2. Sides a and b mentioned in Figure 4.2 are referred to in this section to identify the damage on each side of the beams. Screenshots of the textured mesh models with cracks highlighted in blue color are shown for girders Box Beam-3, I-Beam 9, and I-Beam 1 in Figure 4.5, Figure 4.6, and Figure 4.7, respectively. Cracks missed in this detection are mentioned. It should be noted that the cracks on Box Beam-3 had been previously mapped on the beam using a red marker before the images were captured to build the point cloud for a different purpose which had nothing to do with this framework. Drawing along the cracks on a beam with a marker is not required for crack detection process proposed in this research.



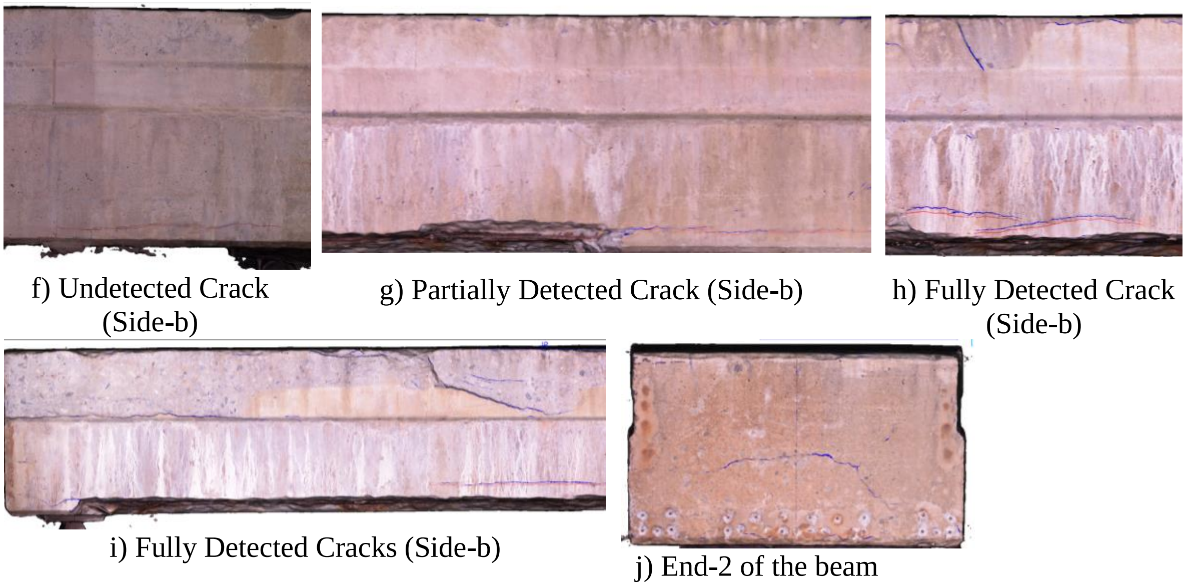
a) Undetected Cracks (Side-a)

b) End-1 of the beam

c) Cracks Either Partially or Fully Detected (Side-a)

d) Undetected Cracks (Bottom)

e) Partially Detected Cracks (Bottom)



f) Undetected Crack (Side-b)

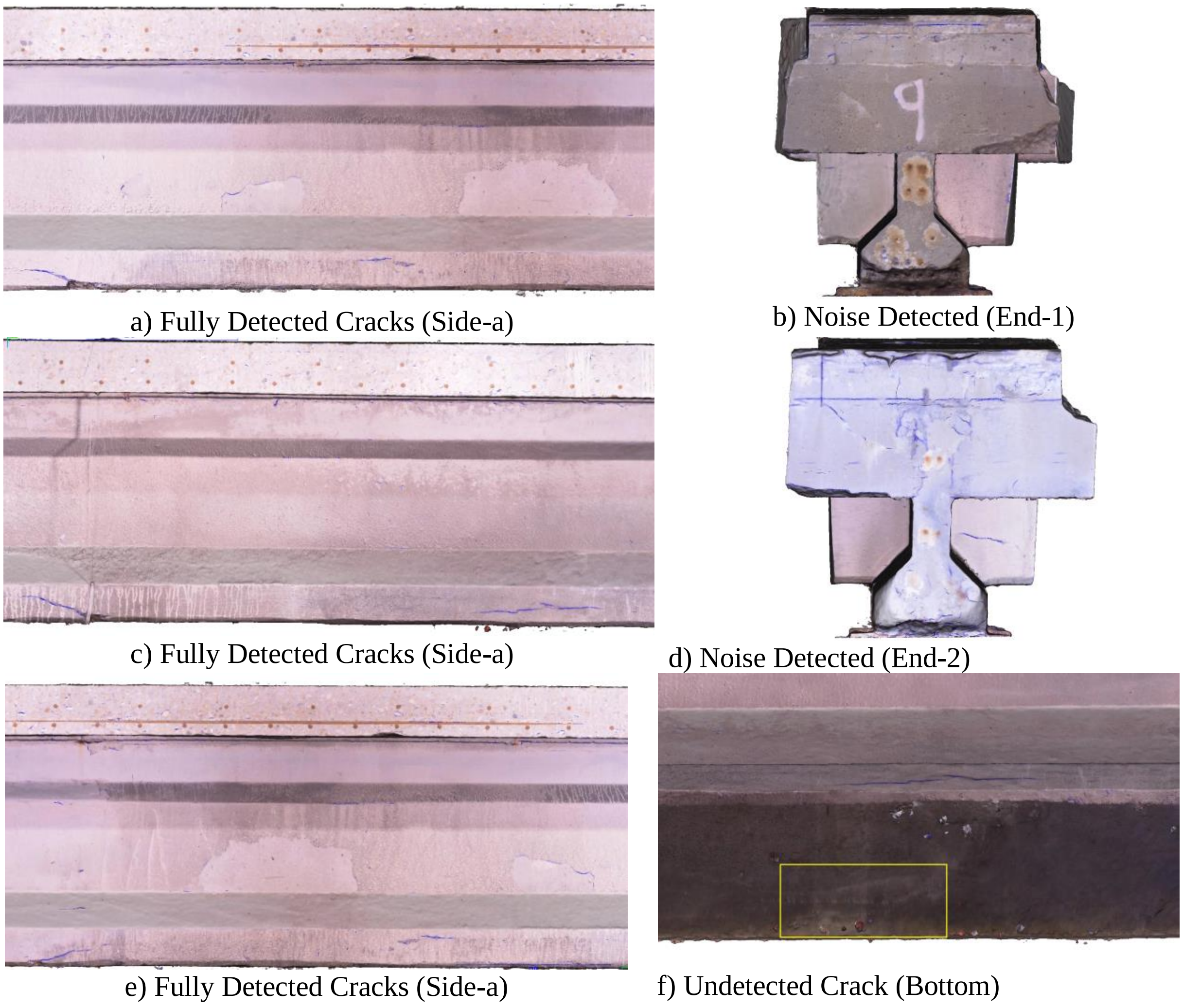
g) Partially Detected Crack (Side-b)

h) Fully Detected Crack (Side-b)

i) Fully Detected Cracks (Side-b)

j) End-2 of the beam

Figure 4.5. Cracks on the Textured Mesh Model of Box Beam-3



a) Fully Detected Cracks (Side-a)

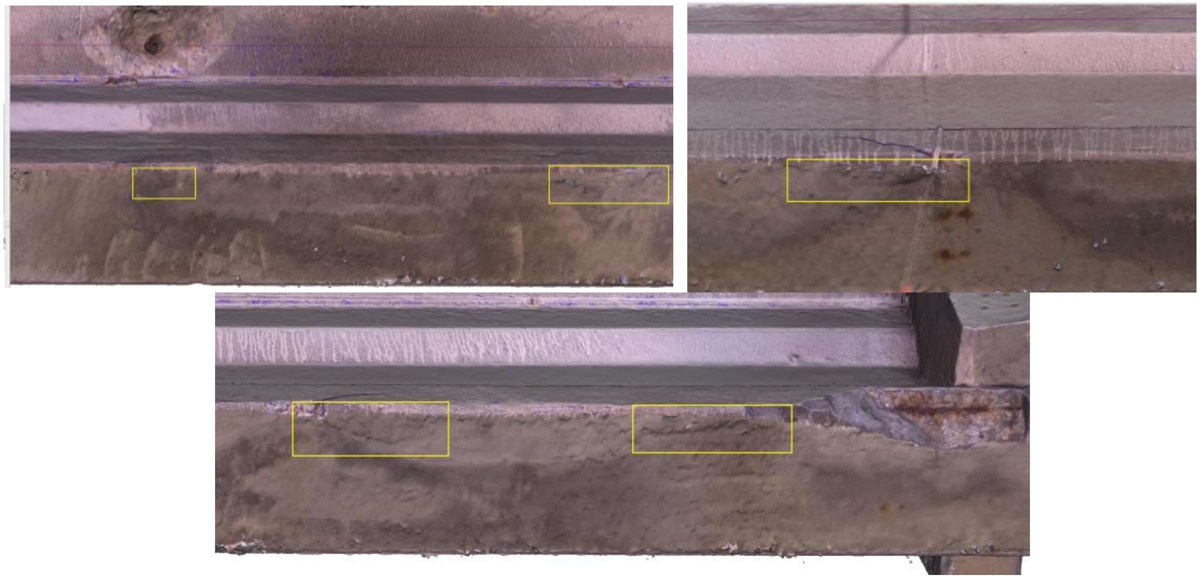
b) Noise Detected (End-1)

c) Fully Detected Cracks (Side-a)

d) Noise Detected (End-2)

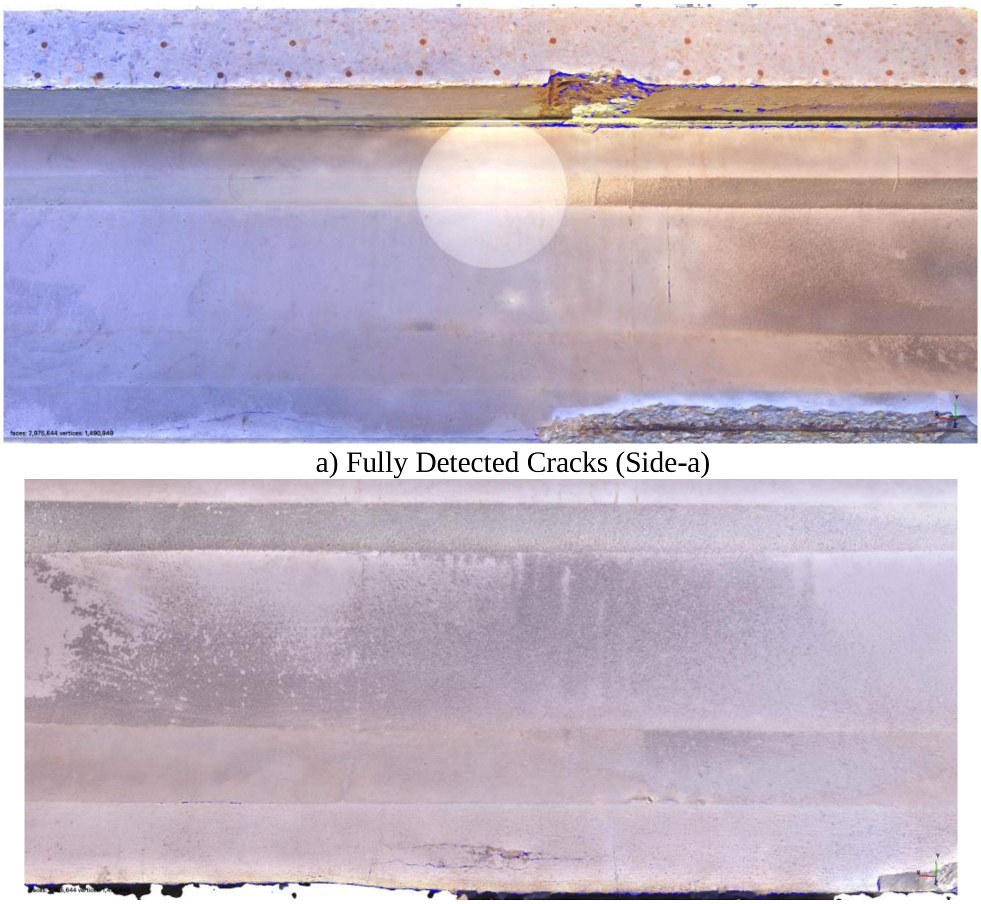
e) Fully Detected Cracks (Side-a)

f) Undetected Crack (Bottom)



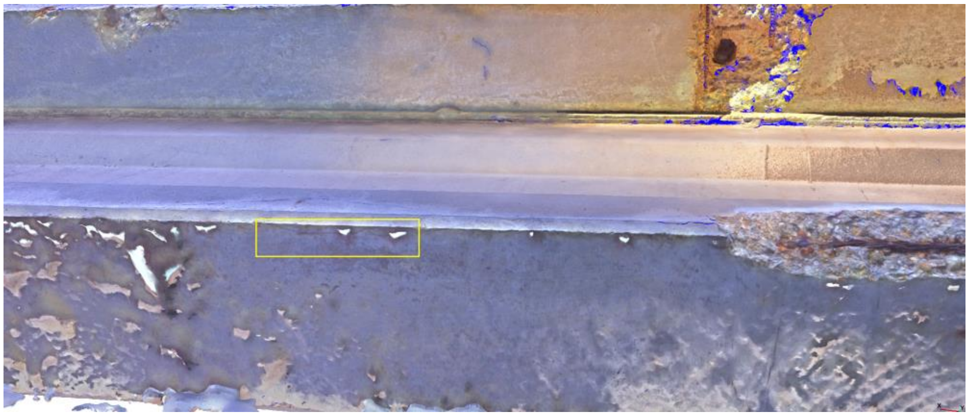
g) Undetected Cracks (Bottom) Highlighted Using Bounding Boxes

Figure 4.6. Cracks on the Textured Mesh Model of I-Beam 9



a) Fully Detected Cracks (Side-a)

b) Fully Detected Cracks (Side-a)



c) Undetected Crack (Bottom)

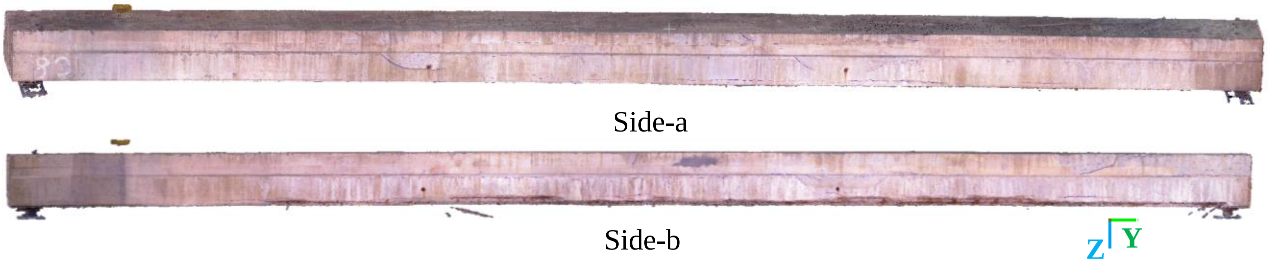
Figure 4.7. Cracks on the Textured Mesh Model of I-Beam 1

The cracks on the bottom of the beams were sometimes the extensions of the damage present on the sides of the bottom flange, as can be observed in Figure 4.6. So, these regions had to be checked thoroughly to locate missing cracks. One of the reasons for the cracks on the bottom surface going undetected could be due to the angle at with the images of the bottom surfaces of the beams were captured during image acquisition. The model was not trained on such pictures. So, more training images from the current dataset can improve the performance of this model. It was observed that, for a crack to be detected on the textured mesh, the crack did not need to be detected on all images that it was seen in. For example, if a crack feature were seen in nine different images, but it was detected by the model only in two of the nine images, this crack could still be included in the texture. In the next step, colors from the crack detected images were applied to the dense point clouds of the girders. The points belonging to the cracks are isolated using thresholding and cleaned to eliminate the background noise. This is demonstrated in the following sub-section.

4.2.2.1. Highlighting Cracks on the Dense Point Cloud

Agisoft Metashape enables applying the colors from the crack highlighted images to the dense point cloud. Using this feature, the cracks that were highlighted in blue color on the new images were transferred on to the 3D point clouds, highlighting the points corresponding to the cracks in blue color. The purpose of doing this was to isolate points corresponding to individual cracks and identify their locations and dimensions etc. The effect of each crack was evaluated during the process of residual strength estimation as described in Section 4.3. Points that constitute the cracks on the point cloud were saved as a separate point cloud ('.ply' file) and was used for clustering as described in Section 3.1.2.7. For this step, the dense point clouds shown in Figure 4.1

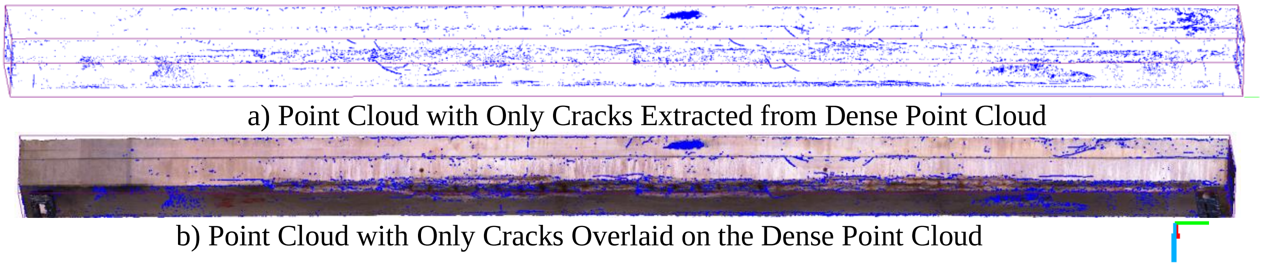
were used. It can be observed that these dense point clouds had some black colored noise points which had to be cleaned first. For cleaning the point cloud Statistical Outlier Removal (SOR) described in 2.5.2 was used. Two manually defined parameters in this process were neighborhood size ‘k’ and the multiplier to standard deviation ‘α’. Trials were conducted with different values of these parameters. A ‘k’ value of 800 and ‘α’ value of 0.75 were found to give better results for this step. Cleaned dense point cloud of Box Beam-3 cleaned using SOR with k = 800 and α = 0.75 is shown in Figure 4.8. Then, the dense point cloud was scaled to the real-world dimensions and rotated to align with the coordinate axes as explained in Section 3.1.2.3. The scaling factor was determined by measuring the width of the beam at two different sections using a tape measure and measuring the corresponding distances at the same sections on the dense point clouds using distance measurement tool on CloudCompare. The dense point cloud and the textured mesh were aligned with the coordinate axes such that the longitudinal direction of the point cloud is along the Y-axis, the depth direction of the cross-section is along the Z-axis and width direction is aligned with the X-axis. The positive Z-axis was facing towards the bottom surface of the beam. The cleaned dense point cloud of Box Beam-3 shown in Figure 4.8 is aligned with the axes as explained above. The dense point cloud of Box Beam-3 before cleaning is shown in Figure 4.1a.


Side-a

Side-b

Figure 4.8. Dense Point Cloud of Box Beam-3 Cleaned Using Statistical Outlier Removal with k = 800 and α = 0.75

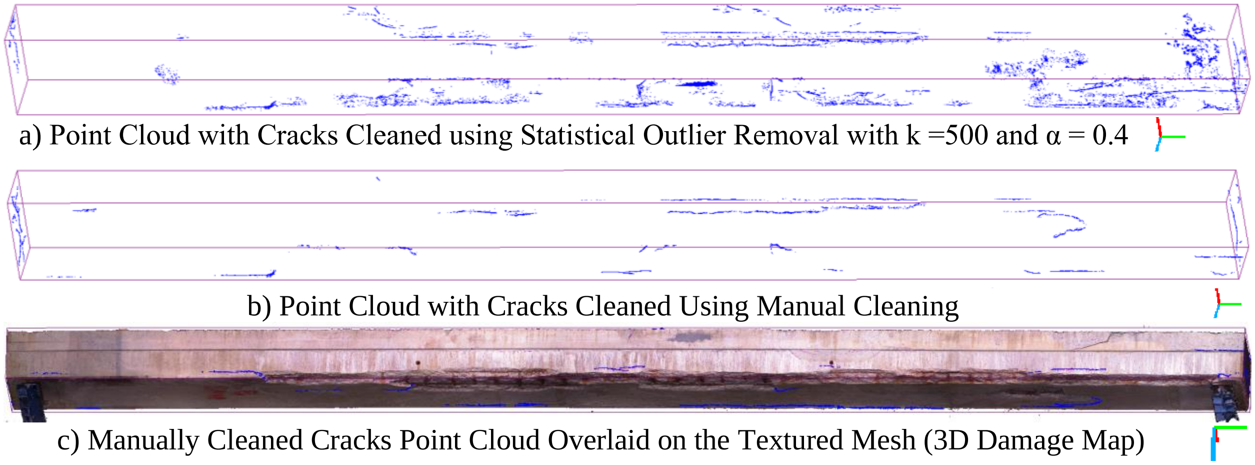
After the dense point cloud was colorized using the crack highlighted images, the points corresponding to cracked regions (in blue color) were extracted into a new point cloud file using thresholding on R,G,B values as described in Section 3.1.2.7. The result of this step is shown in Figure 4.9. This cracks only point cloud was again cleaned using SOR and manual cleaning. The parameters that worked best for cleaning the cracks only point cloud were k = 500 and α = 0.4. Reducing the value of α further, resulted in losing some portions of the cracks, which is not desirable. So, manual cleaning was done using CloudCompare. In this process, cracks that were omitted after cleaning using SOR were manually added back to the cracks point cloud. The results of cleaning using SOR and manual cleaning for the point cloud with cracks for Box Beam-3 are shown in Figure 4.10. Cracks point cloud extracted from I-Beam 9 before and after cleaning is shown in Figure 4.11. There were no cracks on I-Beam 5. Cracks point cloud extracted from I-Beam 1 before and after cleaning is shown in Figure 4.12. The parameters for cleaning (k and α) depend on the density of the point clouds with cracks. This, in turn, depends on the accuracy of the 2D level damage detection step. A more accurate model would generate fewer false positives which would reduce the noise points in the cracks point cloud. The values of parameters for cleaning the point cloud would be affected due to this and have to be determined using trial and error. Qualitative evaluation should be used to determine the appropriate parameters that do not lead to cracks being omitted. In the next step, cracks were subjected to clustering to classify them into individual clusters for use in further steps of the framework. This is discussed in Section 4.2.4.



a) Point Cloud with Only Cracks Extracted from Dense Point Cloud

b) Point Cloud with Only Cracks Overlaid on the Dense Point Cloud

Figure 4.9. Cracks Transferred to the Dense Point Cloud of Box Beam-3 Extracted Using Thresholding on R, G, B Color Values of the Vertices with R<153, G<153 and B>153

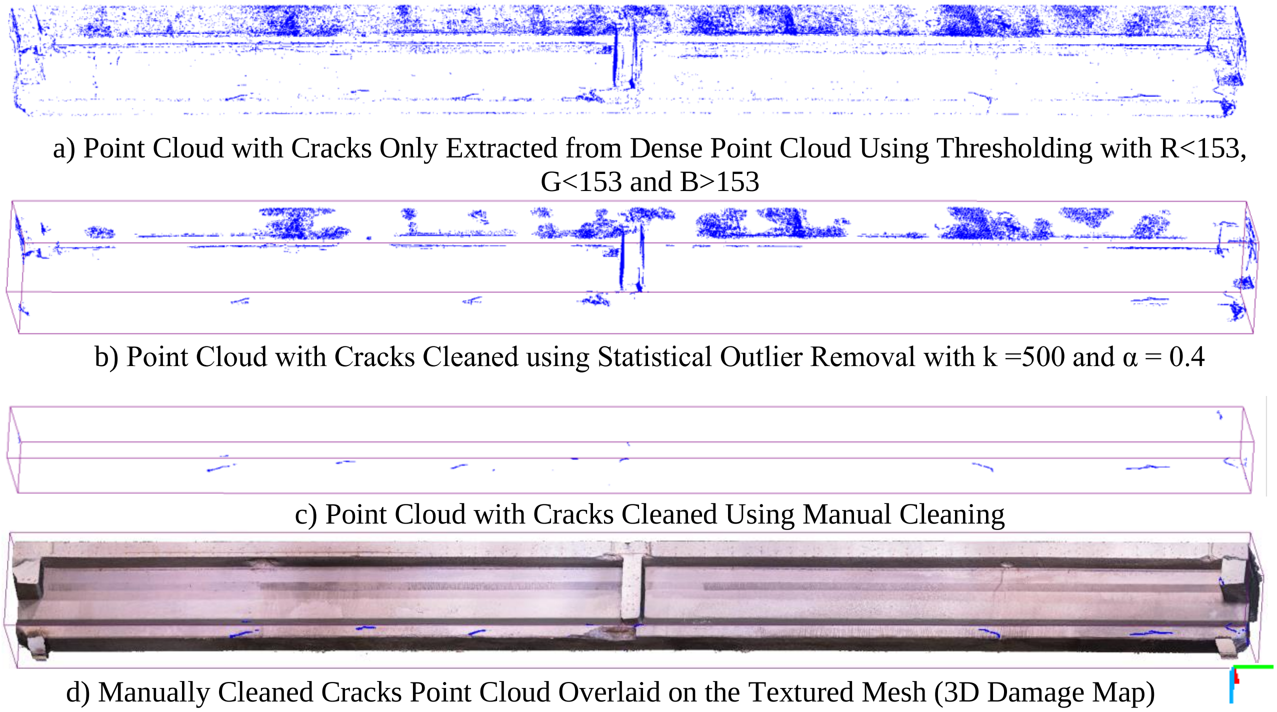


a) Point Cloud with Cracks Cleaned using Statistical Outlier Removal with k =500 and α = 0.4

b) Point Cloud with Cracks Cleaned Using Manual Cleaning

c) Manually Cleaned Cracks Point Cloud Overlaid on the Textured Mesh (3D Damage Map)

Figure 4.10. Point Cloud with Cracks on Box Beam-3 Cleaned Using SOR with k = 500 and α = 0.4 and Manual Cleaning



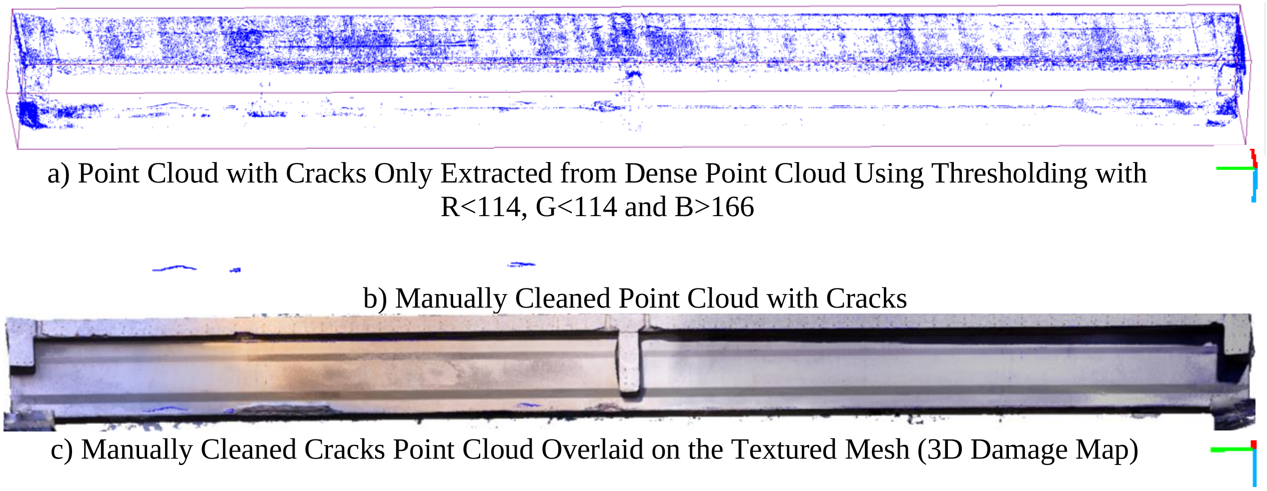
a) Point Cloud with Cracks Only Extracted from Dense Point Cloud Using Thresholding with R<153, G<153 and B>153

b) Point Cloud with Cracks Cleaned using Statistical Outlier Removal with k =500 and α = 0.4

c) Point Cloud with Cracks Cleaned Using Manual Cleaning

d) Manually Cleaned Cracks Point Cloud Overlaid on the Textured Mesh (3D Damage Map)

Figure 4.11. Point Cloud with Cracks for I-Beam 9



a) Point Cloud with Cracks Only Extracted from Dense Point Cloud Using Thresholding with R<114, G<114 and B>166

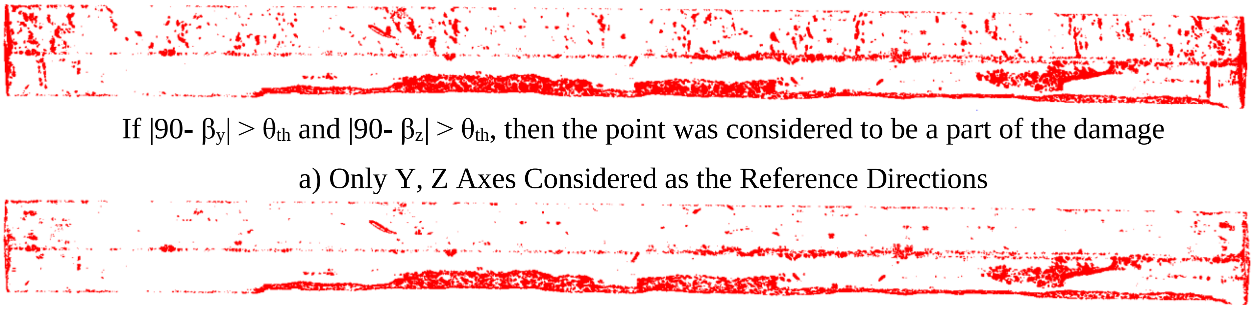
b) Manually Cleaned Point Cloud with Cracks

c) Manually Cleaned Cracks Point Cloud Overlaid on the Textured Mesh (3D Damage Map)

Figure 4.12. Point Cloud with Cracks for I-Beam 1

4.2.3. Spall Detection on 3D Point Clouds

In this report, this is also referred to as 3D level damage detection as the damage was detected directly on the 3D point cloud as opposed to crack detection. The objective of this step was to detect the spalled regions on the concrete surface with minimum noise or false positives. The orientation of surface normals, calculated using the method described in Section 3.1.2.6, with respect to the reference directions (X, Y, Z axes) was subject to thresholding to identify the locations where the normals deviate from being exactly perpendicular to the reference directions, by the amount greater than the threshold angle. The reference directions were arrived at using two trials. In the first trial, only Y, Z directions were used, and in the second trial, X, Y, Z axes were chosen to be the reference directions. The results of these trials are presented in Figure 4.13. Using all three axes as reference was observed to give less noise. This allowed using a low value for threshold angle (θ_{th}) without getting too much background noise. There are two manual parameters considered in this algorithm. The first one is the size of the neighborhood around each point to be used to fit a plane and find the normal to the plane. The second manual parameter is the threshold angle to which the orientations of surface normals are compared. One other step that had to be done before doing spall detection was reducing the size (total number of points) of the point cloud to reduce computational time and achieve efficiency as described in Section 3.1.2.6.1. This process involves a manually defined parameter called 'voxel size' (vs). The effects of all these manual parameters are discussed in the following sub-sections.



If $|90-\beta_y| > \theta_{th}$ and $|90-\beta_z| > \theta_{th}$, then the point was considered to be a part of the damage

a) Only Y, Z Axes Considered as the Reference Directions

If $|90-\beta_x| > \theta_{th}$ and $|90-\beta_y| > \theta_{th}$ and $|90-\beta_z| > \theta_{th}$, then the point was considered to be a part of the damage

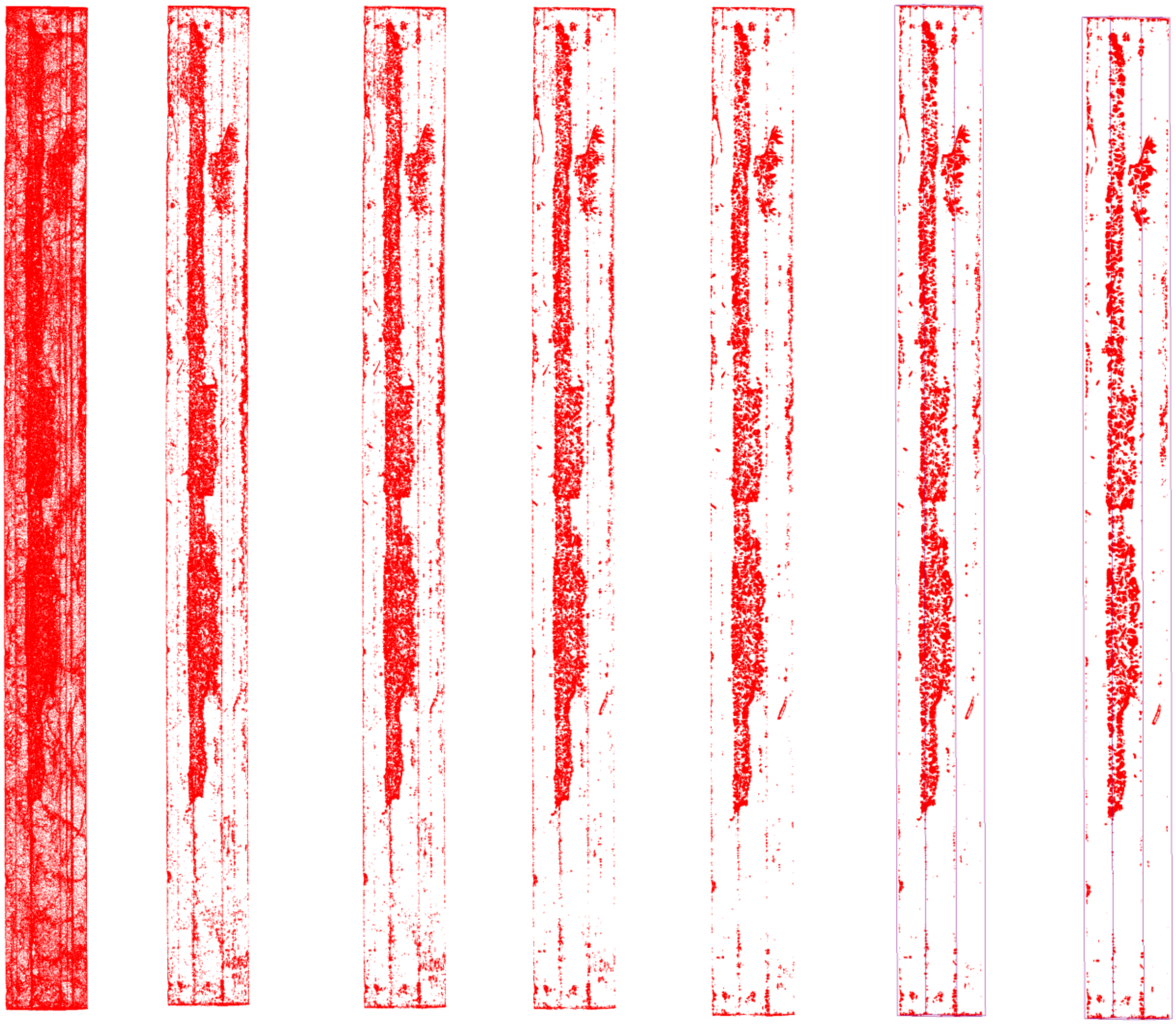
b) X, Y, Z Axes Considered as the Reference Directions

Figure 4.13. Effect of Reference Axes Chosen on the 3D Level Damage Detection on Box Beam-3 Using k =150 and Threshold Angle = 2 degrees

4.2.3.1. Effect of Neighborhood Size 'k'

For higher values of 'k' chosen, larger neighborhoods of points were used for plane fitting which reduced the effect of sharp local variation. On the other hand, if a plane was fit to a smaller neighborhood, the effect of local variation was observed to be high, capturing more local roughness. As the neighborhood size was only a number of points, the optimal value of 'k' also depended on the density of the points in the point cloud. For the same value of 'k', a point cloud of higher density gave more local roughness in the detection, compared to a sparser point cloud. In the current work, only spalls were desired to be retained and the value of k was incrementally increased to study its effect. As the value of ‘k’ was increased, the computational time and power to run the program also increased. So, it was important to put a limit on the value of ‘k’ to avoid running out of computational memory. Also, as the value of ‘k’ was increased, the boundaries of the damage were not captured accurately making the boundaries of spalls detected on the point clouds were larger than those of the actual spalls. This would cause a problem in the subsequent steps as the spall vertices were projected on to the slice locations as explained in Section 3.1.3.3. This could result in more strands being predicted as a part of the spalls when overlaid with the cross-section drawing as explained in Section 3.1.3.4. For this reason, it was established that the value of ‘k’ should not be too high. The damage detected in this step was saved as separate point cloud files (‘.ply’). These results of the 3D level damage detection for spalls, for different values of ‘k’ are shown for the 3D point cloud of Box Beam-3 in Figure 4.14. For these results, a threshold value of two degrees and a voxel size (vs) of 0.2 were used. The effect of the threshold parameter on the results is discussed in Section 4.2.3.2.

From Figure 4.14, it was evident that a low value of ‘k’ caused too much noise in the damage detection which is not desirable. As the value of ‘k’ was increased, the noise was reduced. The detected damage at this stage was overlaid on top of the textured mesh of the box beam in Figure 4.15. As shown in Figure 4.15, all the damage detected from this approach was not corrosion damage as it included some of the surface roughness, surface holes, and wear and tear on the girders. So, the detected damage was cleaned using the statistical outlier removal (SOR) approach described in Section 2.5.2. These parameters were also determined using trial and error. For further cleaning to retain only spalling due to corrosion damage, manual cleaning was done using CloudCompare before the results were subject to clustering. A ‘k’ value of 150 was chosen for 3D level damage detection in the current work. However, higher values of ‘k’ can also be chosen to avoid cleaning up too much. The result of 3D level damage detection for Lesner Bridge I-Beam 9 using k = 150 and threshold angle = 2 deg. is shown in Figure 4.16. The result of 3D level damage detection for Lesner Bridge I-Beam 5 and I-Beam 1 using k = 150 and threshold angle = 4 deg. are shown in Figure 4.17 and Figure 4.18, respectively. Cleaned-up versions of the 3D-level damage detection on Box Beam-3, I-Beam 9, I-Beam 5 and I-Beam 1 using SOR filter with parameters k =2000 and Multiplier for standard deviation $\alpha = 0$ and using manual cleaning are shown in Figure 4.19. Cleaned up spall point clouds overlaid on top of the textured mesh models of the girders (shown in Figure 4.19) are used as 3D damage maps in the subsequent steps.



k =3, threshold = 2 deg.

k=30, threshold = 2 deg.

k=50, threshold = 2 deg.

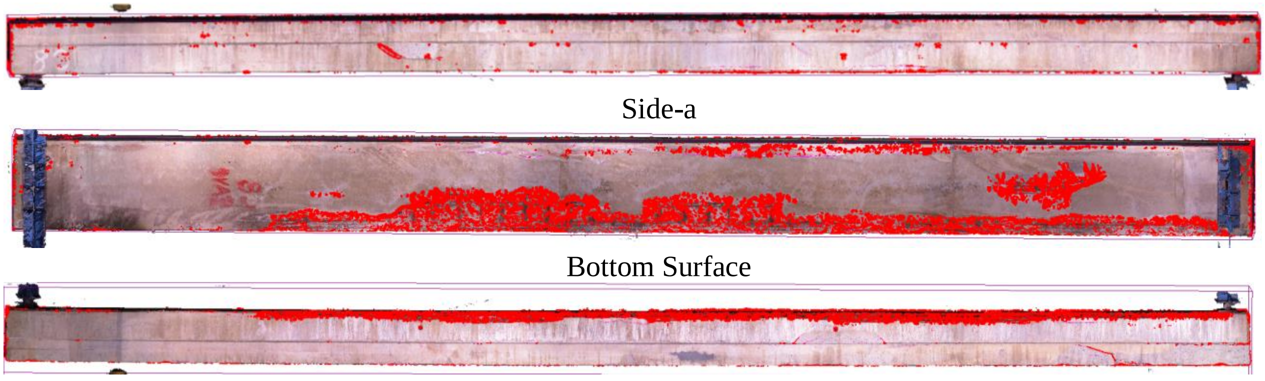
k=100, threshold = 2 deg.

k=150 threshold = 2 deg.

k=200 threshold = 2 deg.

k=300 threshold = 2 deg.

Figure 4.14. Effect of Neighborhood Size (k) on 3D-Level Damage Detection on the Dense Point Cloud of Box Beam-3 with X, Y, Z Reference Directions

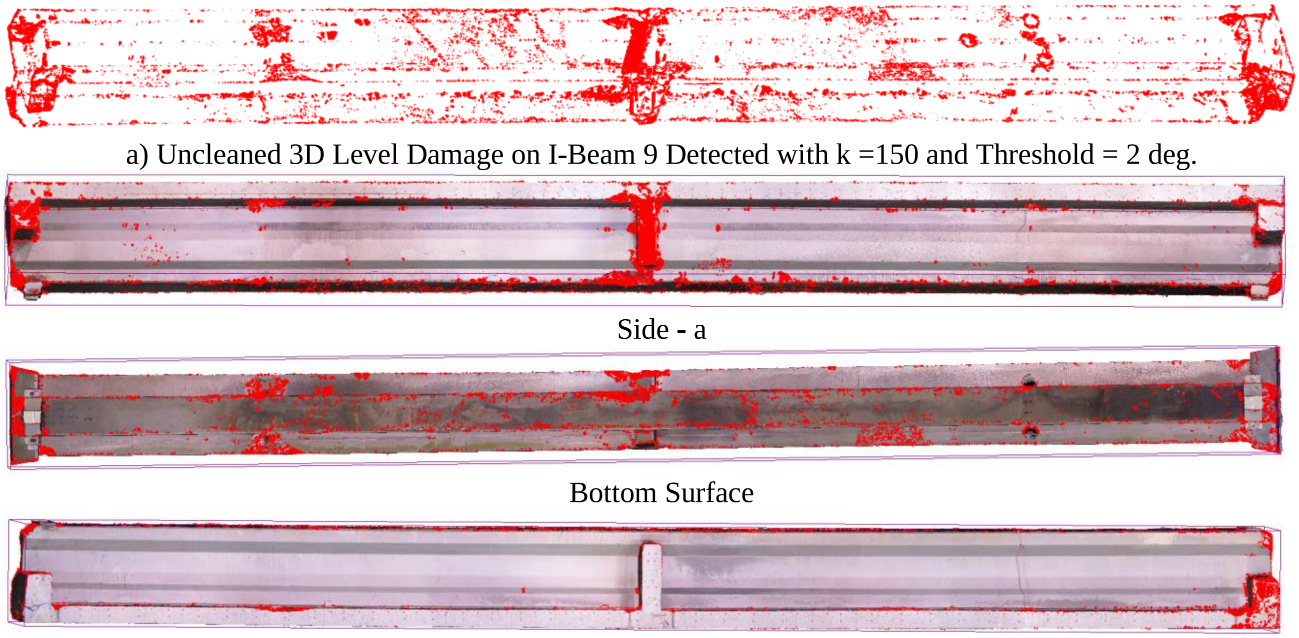


Side-a

Bottom Surface

Side-b

Figure 4.15. Uncleaned 3D Level Damage Detected (k=150, threshold =2 deg.) with X, Y, Z Reference Directions on the Textured Mesh Model of Box Beam-3



a) Uncleaned 3D Level Damage on I-Beam 9 Detected with k =150 and Threshold = 2 deg.

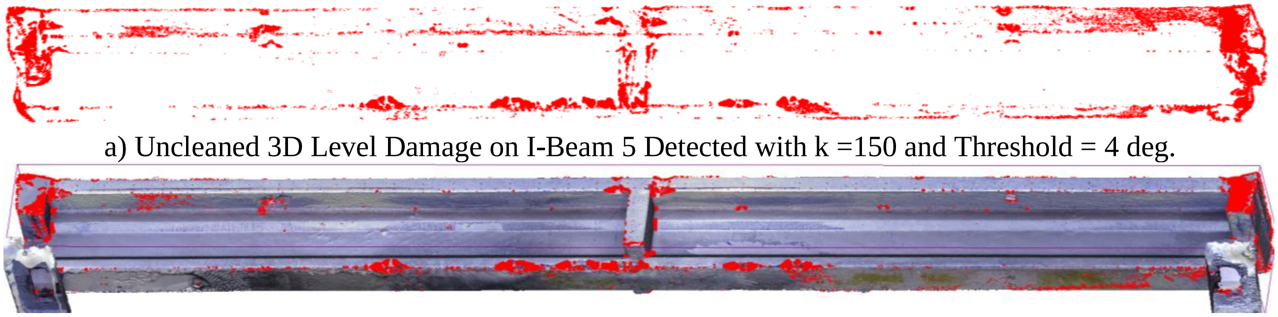
Side - a

Bottom Surface

Side-b

b) 3D Level Damage on the Textured Mesh of I-Beam 9

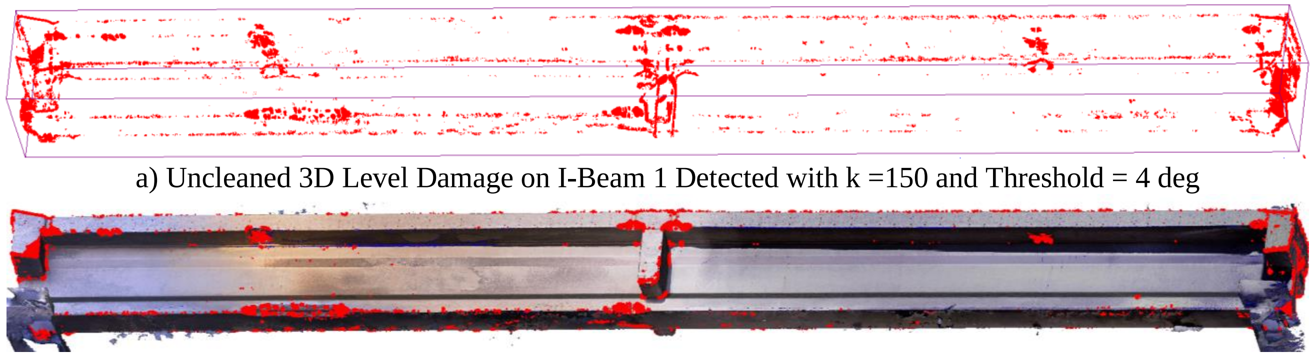
Figure 4.16. 3D Level Damage Detected (k=150, threshold =2 deg.) with X, Y, Z Reference Directions on the Textured Mesh Model of I-Beam 9 before Cleaning



a) Uncleaned 3D Level Damage on I-Beam 5 Detected with k =150 and Threshold = 4 deg.

b) 3D Level Damage on the Textured Mesh of I-Beam 5

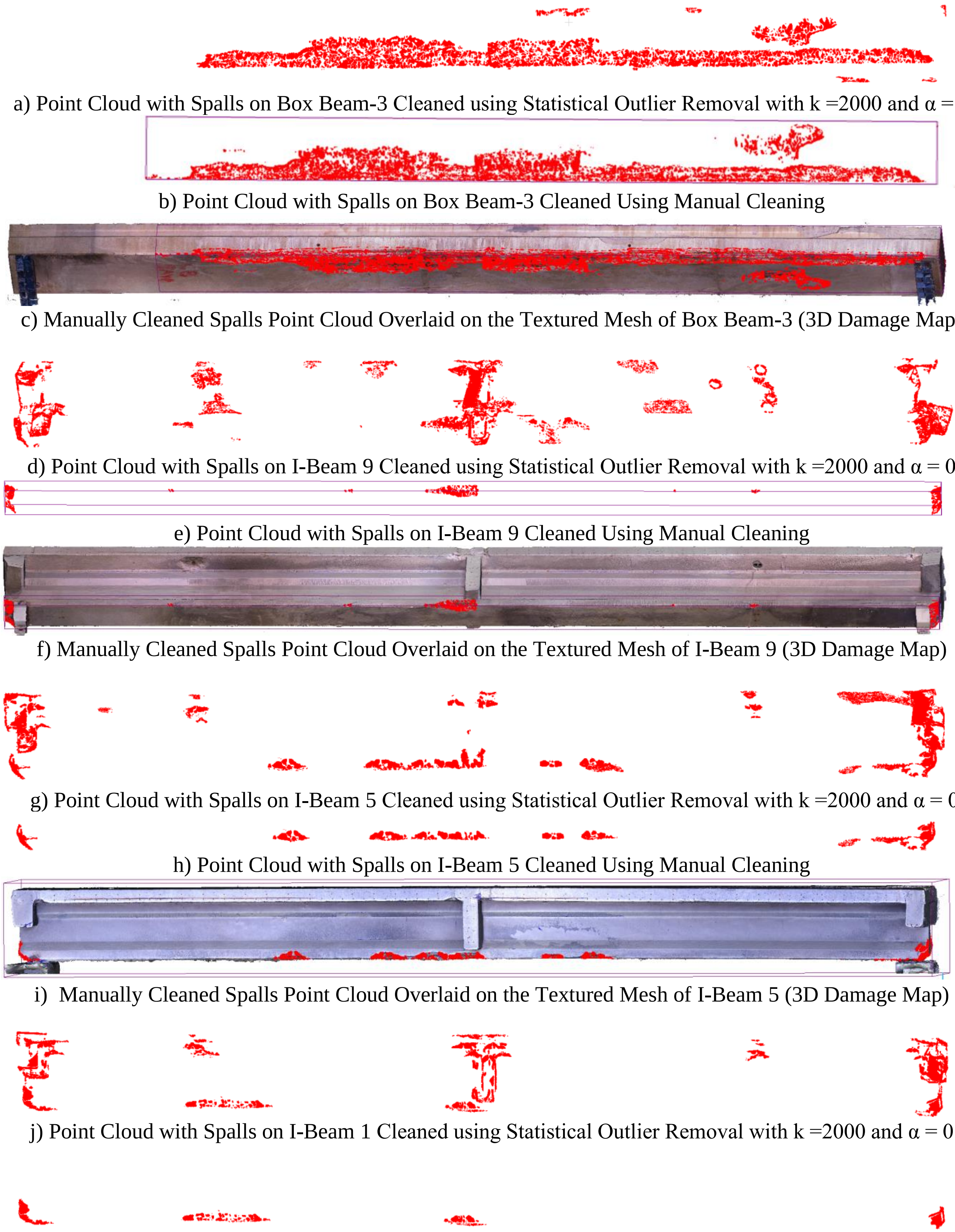
Figure 4.17. 3D Level Damage Detected (k=150, threshold =4 deg.) with X, Y, Z Reference Directions on the Textured Mesh Model of I-Beam 5 before Cleaning



a) Uncleaned 3D Level Damage on I-Beam 1 Detected with k =150 and Threshold = 4 deg

b) 3D Level Damage on the Textured Mesh

Figure 4.18. 3D Level Damage Detected (k=150, threshold =4 deg.) with X, Y, Z Reference Directions on the Textured Mesh Model of I-Beam 1 before Cleaning



a) Point Cloud with Spalls on Box Beam-3 Cleaned using Statistical Outlier Removal with k =2000 and α = 0

b) Point Cloud with Spalls on Box Beam-3 Cleaned Using Manual Cleaning

c) Manually Cleaned Spalls Point Cloud Overlaid on the Textured Mesh of Box Beam-3 (3D Damage Map)

d) Point Cloud with Spalls on I-Beam 9 Cleaned using Statistical Outlier Removal with k =2000 and α = 0

e) Point Cloud with Spalls on I-Beam 9 Cleaned Using Manual Cleaning

f) Manually Cleaned Spalls Point Cloud Overlaid on the Textured Mesh of I-Beam 9 (3D Damage Map)

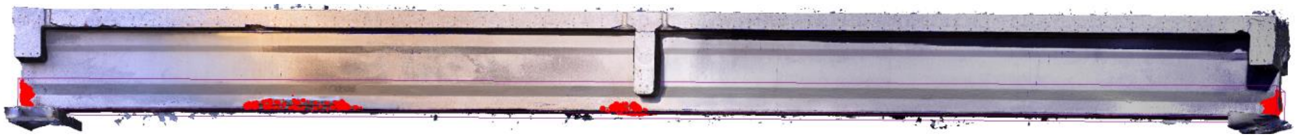
g) Point Cloud with Spalls on I-Beam 5 Cleaned using Statistical Outlier Removal with k =2000 and α = 0

h) Point Cloud with Spalls on I-Beam 5 Cleaned Using Manual Cleaning

i) Manually Cleaned Spalls Point Cloud Overlaid on the Textured Mesh of I-Beam 5 (3D Damage Map)

j) Point Cloud with Spalls on I-Beam 1 Cleaned using Statistical Outlier Removal with k =2000 and α = 0

k) Point Cloud with Spalls on I-Beam 1 Cleaned Using Manual Cleaning



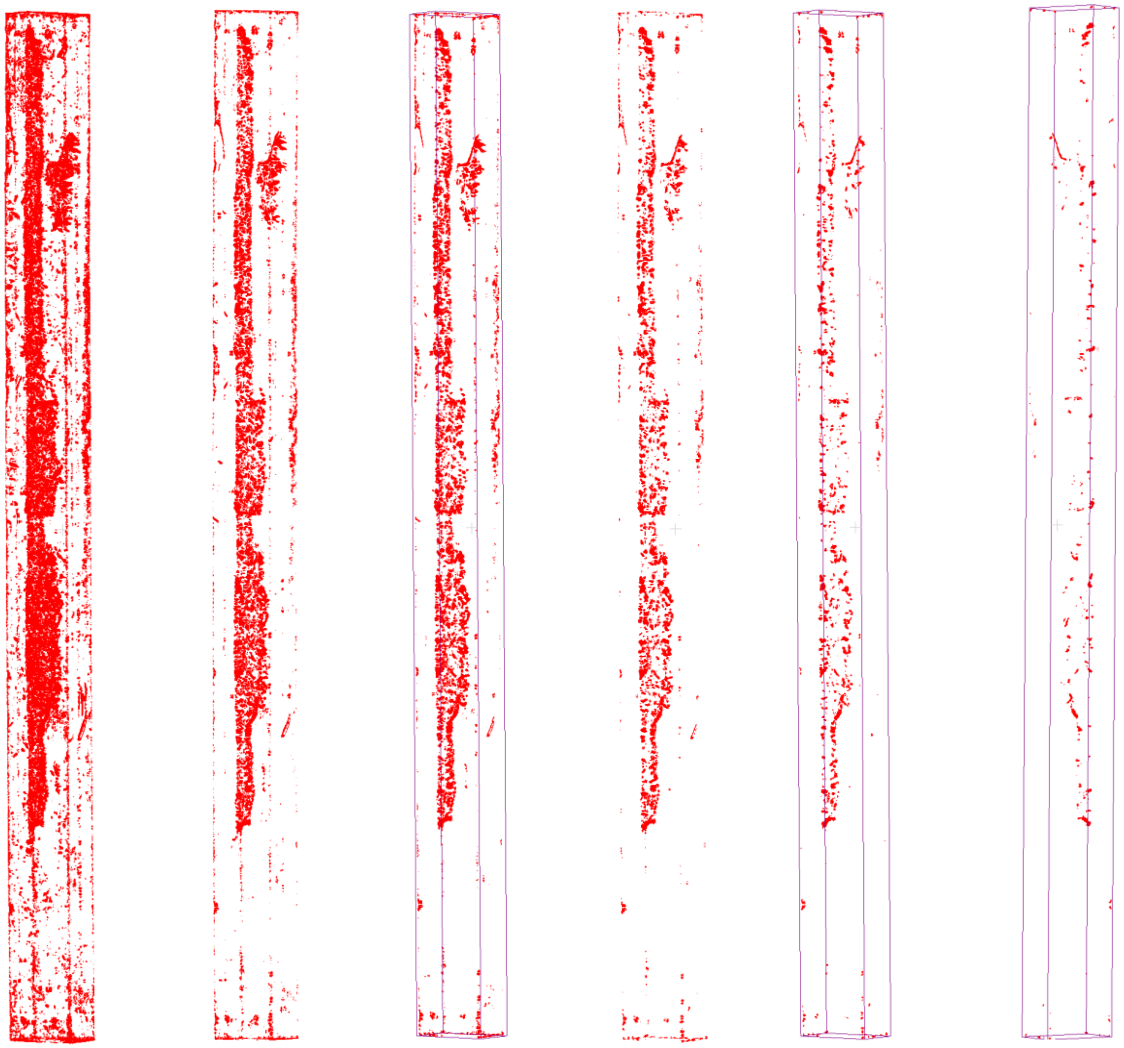
l) Manually Cleaned Spalls Point Cloud Overlaid on the Textured Mesh of I-Beam 1 (3D Damage Map)

Figure 4.19. Point Clouds with Spalls on Box-Beam 3, I-Beam 9, I-Beam 5, and I-Beam 1 (k=150, Threshold = 2 to 4 deg.) Cleaned Using Statistical Outlier Removal with k = 2000 and Multiplier to Standard Deviation α = 0 and Manual Cleaning

Along with a reduction in the noise, the values of 'k' and threshold were also chosen by validating if the parameters used accurately detect the visible spalling on the point clouds. Side-a and Side-b are labeled on the point clouds of the beams in Figure 4.15 and Figure 4.16 for convenience. It can be observed from Figure 4.19c that the spall on the bottom surface of Box Beam-3 located towards Side-b and the spall in the middle (towards the right) were captured well. There was some noise on the side-a which had to be manually cleaned. For I-Beam 9 (shown in Figure 4.19f) there was a spalling at the midspan below the diaphragm which was captured well in this detection. At the ends of the beam, there were patched regions where the normals violated the thresholding criteria. These regions were also captured well along with the locations on the deck where holes were drilled. However, there was some noise detected on the diaphragm at midspan which had to be cleaned manually. On I-Beam 5 and I-Beam 1, spalls were well captured. For automatic cleaning of this 3D level damage detection, SOR parameters k = 2000 and α = 0 were found to work well. Manual cleaning was used to retain only the damage that affected the capacity of the girders.

4.2.3.2. Effect of the Threshold Parameter

The effect of the threshold angle parameter is demonstrated in Figure 4.20 for a chosen value of k = 150 points. As the threshold value is increased, false positives were greatly reduced. But, along with the reduced false positives, large portions of the spalled regions were also excluded from the 3D level damage detection. For this reason, a lower value of the threshold angle was chosen. As demonstrated in in Section 4.2.3.1, in this research, a threshold angle of two to four degrees was adopted and all three axes (X, Y, Z) were used as reference directions to compare the orientation of each point normal. The criteria of comparing the orientation of each point normal to the three reference directions were joined using 'and' operator. A higher value of threshold can result in less cleaning. For any new dataset, these parameters should be determined by trial and error to arrive at optimal parameters because the optimal value of ‘k’ would depend on the density of the point cloud.



k=150 threshold = 1 deg.

k=150 threshold = 2 deg.

k=150 threshold = 3 deg.

k=150 threshold = 4 deg.

k=150 threshold = 6 deg.

k=150 threshold = 10 deg.

Figure 4.20. Effect of Threshold Angle Parameter on 3D-Level Damage Detection with X, Y, Z Reference Directions on the Dense Point Cloud of Box Beam-3

4.2.3.3. Voxel Size Parameter for Downsizing the Dense Point Clouds

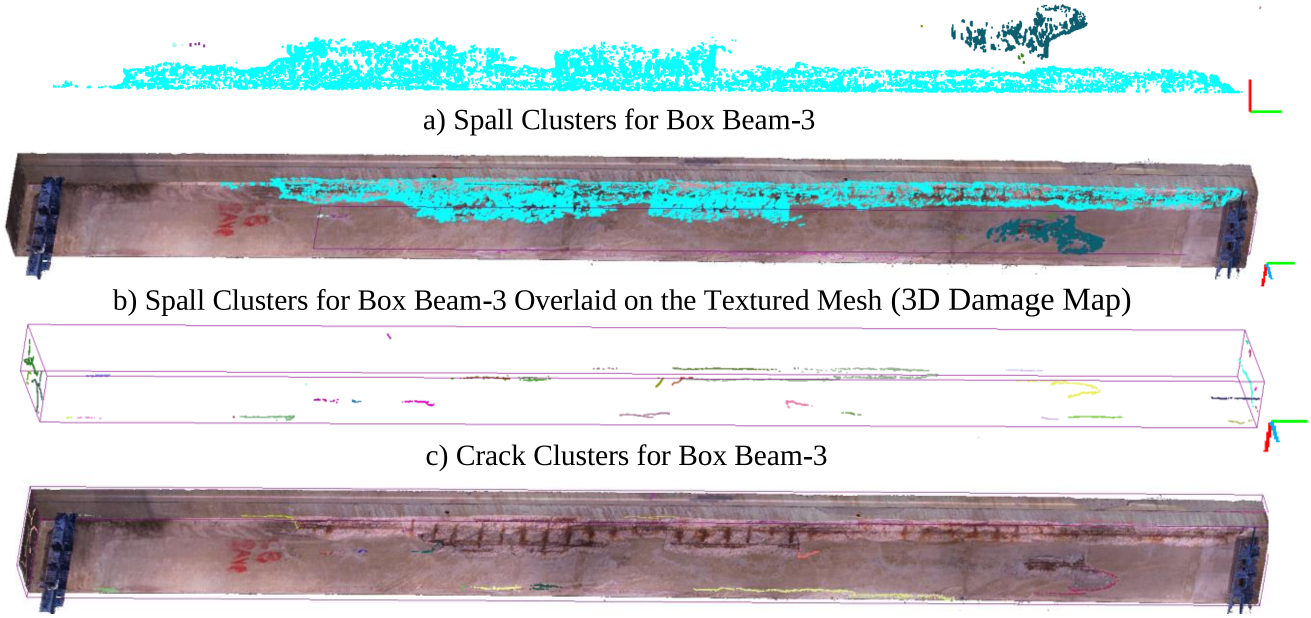
In this research, a voxel size of 0.15 to 0.2 was used to downsample the dense point clouds of the girders as explained in Section 3.1.2.6.1. This reduction in size was done before the computation of surface normals associated with all the points. The program written in Python for this step was run on the workstation computer whose specifications are described in Section 3.2 across the purposes it was used for. If the dense point clouds were not downsized for this step, it was observed that the computer ran out of memory. Hence, downsizing was the logical solution to this problem. As mentioned in Section 3.1.1.3, the sizes of dense clouds of the bridge girders used in this research were between fifteen million to twenty-six million points. Using a voxel size of 0.2 reduced the total size of the dense point cloud to about 10% to 15% of the original size. Using a voxel size of 0.2 reduced the computational time to about eight minutes to fifteen minutes to compute the surface normals of the reduced number of points, find their orientation with respect to the reference axes, and subject them to thresholding to arrive at the damage detection. Time taken to run the program also varied with the value of ‘k’. For a voxel size of 0.2, a ‘k’ value of 30 resulted in computational time of 9 minutes and a ‘k’ value of 300 took about 15 minutes. For a voxel size of 0.2, the computer was not able to handle a ‘k’ value of 500. So, for higher values of ‘k’, downsizing the dense clouds further with a voxel size greater than 0.2 is recommended.

4.2.4. Clustering the Damage to Isolate Each Crack or Spall

The objective of this step was to divide the detected damage points into individual clusters so that they can be used to extract information such as the location (in coordinates), the dimensions of cracks and spalls, etc. which would be useful in identifying damage that affects the strength of the beam at any section. In the current research DBSCAN function of ‘scikit-learn’ Python library was used. The procedure followed for clustering is described in Section 3.1.2.7. Clustering involves two manual parameters ‘eps’ and ‘minimum samples’. Trials were conducted to determine appropriate values for these parameters. As explained before, ‘minimum samples’ controls noise filtering. A larger value of this parameter results in more noise filtering. However, it was observed that using a higher value for eps resulted in losing some parts of cracks especially in locations where they were only partially detected. For some cracks, only a few points sparsely distributed along the length of the cracks were detected. Such cracks were eliminated as noise if a higher value of ‘minimum samples’ was used. To avoid this, cleaning of cracks and spalls point clouds were done, even if it had to be done manually so that the damage information is not lost. However, if the performance of the neural network for crack detection is improved, partially detected cracks would be low in number. The need for manual cleaning would also be reduced. In such a case, a higher value of ‘minimum samples’ could be used in the DBSCAN algorithm to achieve some noise elimination. In this research, as the 2D level damage detection (cracks) had more noise, they had to be manually cleaned and so, a low value of five was chosen for ‘minimum samples'. Trials for parameter 'eps' showed that a value of eps = 3 caused most cracks in the point cloud to be in separate clusters. The same parameters of five minimum samples and eps value of three were also used for the clustering of spalls. However, there were some instances in which multiple cracks were in the same cluster. The effect of the 'eps' parameter is demonstrated, showing the number of clusters generated, in Table 4.3 for clustering cracks on the cleaned point cloud with cracks of Box Beam-3. The total number of crack and spall clusters generated for these parameters for Box Beam-3 and I-Beam 9 are shown in Table 4.4. Different Crack and Spall Clusters for Box Beam-3 and I-Beam 9 are demonstrated in different colors in Figure 4.21. Damage overlaid on Textured mesh models constitute the 3D Damage Maps of the girders to be used in the subsequent steps.

Table 4.3. Number of Crack Clusters Generated for Minimum Samples = 5 and Varying 'eps' Values for Cracks Point Cloud of Box Beam-3

No. of Minimum Samples	eps	No. of Cracks Clusters Generated
5	1.5	73
5	1.8	60
5	1.9	53
5	2	50
5	2.8	38
5	3	36

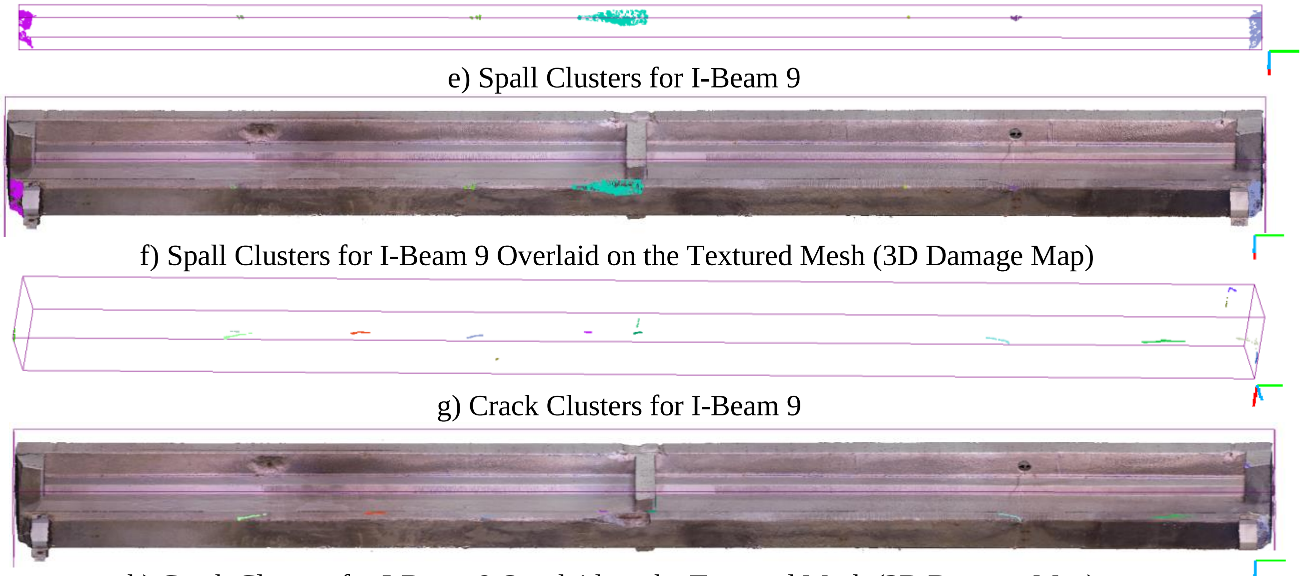


a) Spall Clusters for Box Beam-3

b) Spall Clusters for Box Beam-3 Overlaid on the Textured Mesh (3D Damage Map)

c) Crack Clusters for Box Beam-3

d) Spall Clusters for Box Beam-3 Overlaid on the Textured Mesh (3D Damage Map)



e) Spall Clusters for I-Beam 9

f) Spall Clusters for I-Beam 9 Overlaid on the Textured Mesh (3D Damage Map)

g) Crack Clusters for I-Beam 9

h) Crack Clusters for I-Beam 9 Overlaid on the Textured Mesh (3D Damage Map)

Figure 4.21. Cracks and Spalls Clusters for Box Beam-3 and I Beam-9

Table 4.4. Number of Clusters Generated for Box Beam-3 and I-Beam 9 for Minimum Samples = 5 and eps =3

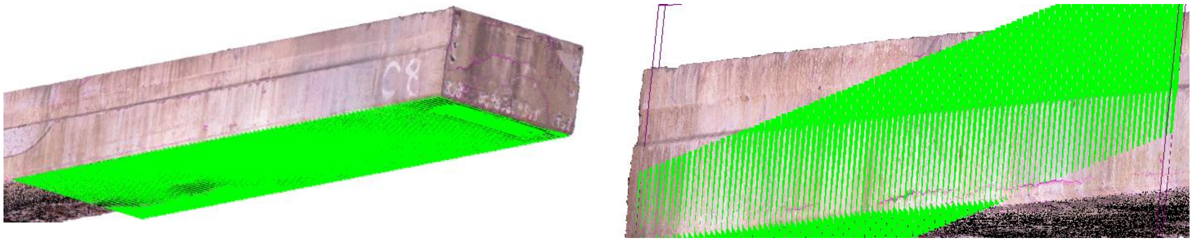
Name of the Beam	For Minimum Samples = 5 and eps = 3	
	Number of Crack Clusters	Number of Spall Clusters
Box Beam-3	36	10
I-Beam 9	15	7

It was observed that there were some noise clusters of both cracks and spalls. These did not affect the strength of the beams and were neglected in calculations. A minimum number of points threshold can be defined and clusters with points less than the specified threshold can be merged with the cluster that is the nearest to them. A distance threshold can also be specified for this distance to the nearest cluster parameter. This will result in a fewer clusters. Before proceeding to the strength estimation step, the clusters were projected on to the planes fit to the undamaged side surface of the beam and the bottom surface of the beam. This was done to roughly estimate the dimensions of the damage clusters. These results are discussed in the following sub-section.

4.2.4.1. Attempt at Estimating the Dimensions of the Damage Clusters

As explained in Section 3.1.2.7.2, planes were fit to the bottom and side surfaces of the beams to project each of the damage clusters on to both these planes in order to estimate the dimensions of damage clusters. Bounding rectangles drawn around the projected clusters were used to obtain estimates length, width, and depth of the clusters in two dimensions. The objective was to get an idea about the extent of damage clusters (their starting point and their endpoint). For cracks, the only dimension approximated here is their length. Then, an attempt was made to approximately determine the length of the cracks in three dimensions by slicing each of the crack clusters into a prescribed number of slices and replacing the vertices between any two consecutive slices with the centroid of these points. The distances between all the consecutive centroids were added to give an approximate length of the crack in three dimensions. This improves the accuracy of measurement for inclined cracks. Here, the number of slices taken was chosen arbitrarily as fifty. If this parameter was chosen to be too high and the crack shifted in direction transversely, larger values of the total length were inaccurately obtained. Fifty slices were found to give satisfactory estimates of crack length in three dimensions. For spalls, the dimensions of interest are length, width and depth, area, and volume. Here, an attempt was made to approximately estimate only the lengths, widths, and depths of the spalls from the projections on the bottom and side planes. The bottom plane need not be exactly fit to the undamaged bottom surface of the beam and any horizontal plane would serve the purpose. Planes fit to the bottom and side surfaces of Box Beam-3 are shown in Figure 4.22.

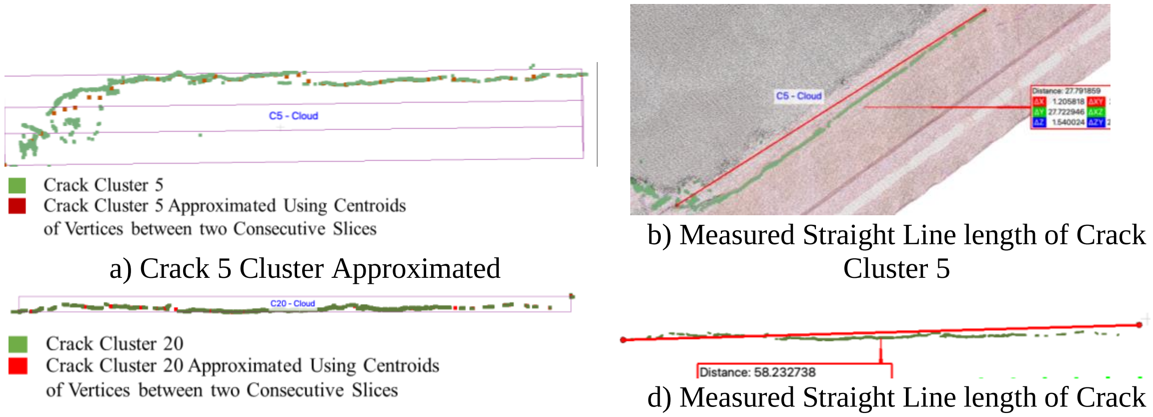
Results of the dimension estimation for Box beam-3 are shown in Figure 4.23, Figure 4.24, Table 4.5, and Table 4.6. In Figure 4.23a and Figure 4.23c, Crack Cluster 5 and Crack Cluster 20 approximated using the centroids of vertices between two consecutive vertices (in red color) are shown. In Figure 4.23b and Figure 4.23d, straight-line dimensions of the cracks measured using CloudCompare are shown to validate the automated measurements obtained from the program. Sample Lengths of Cracks Clusters 5 and 20 measured in two dimensions were 30.76 in and 58.23 in, respectively are shown in Table 4.5. As Crack 5 was shifting from being horizontal, its length measured in three dimensions (33.61 in) was slightly longer than its 2D approximation. The length of Crack cluster 20 is unchanged between two and three dimensions because it was almost horizontal. This is only shown here to demonstrate that this method works and gives an estimate close to the actual dimension. The automatically measured dimensions for Spall 0 cluster shown in Table 4.6 were validated using measurements on the point cloud in Figure 4.24. It can be observed that automated measurements provide good estimates of the actual dimensions of the Spall cluster. The measurements on the Spall 0 point cloud may not be taken exactly at the same locations that the program is measuring the length at. There could be some outliers for each spall and crack cluster that affect the automatic measurements and can be removed automatically using an outlier detection algorithm to get better estimates of the actual dimensions. This method can be improved further to get better estimates of the dimensions and can be extended to measure the crack widths automatically.



a) Plane Fit to the Bottom Surface Shown Using Some Representative Points

b) Plane Fit to the Side Surface Shown Using Some Representative Points

Figure 4.22. Planes Fit to Bottom and Side Surfaces of Box Beam-3


C5 - Cloud
🟩 Crack Cluster 5
🟥 Crack Cluster 5 Approximated Using Centroids of Vertices between two Consecutive Slices
C5 - Cloud
Distance: 27.791859
ΔX 1.205818 ΔXY
ΔY 27.722946 ΔXZ
ΔZ 1.540024 ΔZY
C20 - Cloud
🟩 Crack Cluster 20
🟥 Crack Cluster 20 Approximated Using Centroids of Vertices between two Consecutive Slices
Distance: 58.232738

a) Crack 5 Cluster Approximated

b) Measured Straight Line length of Crack Cluster 5

c) Crack 20 Cluster Approximated

d) Measured Straight Line length of Crack Cluster 20

Figure 4.23. Example Cracks Approximated with the Centroids of Vertices Between Two Consecutive Slices

Table 4.5. Sample Lengths of Cracks Clusters Measured in Two and Three Dimensions

Name of the Crack Cluster	Length Estimated in Two Dimensions, in	Length Estimated in Three Dimensions, in
Crack Cluster 5	30.76	33.61
Crack Cluster 20	58.23	57.83

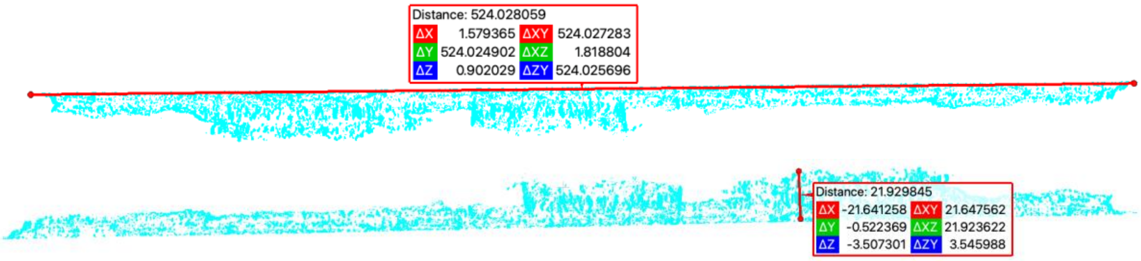

Distance: 524.028059
ΔX 1.579365 ΔXY 524.027283
ΔY 524.024902 ΔXZ 1.818804
ΔZ 0.902029 ΔZY 524.025696
Distance: 21.929845
ΔX -21.641258 ΔXY 21.647562
ΔY -0.522369 ΔXZ 21.923622
ΔZ -3.507301 ΔZY 3.545988

Figure 4.24. Figure Showing Dimensions Spall Cluster 0 of Box Beam-3 Measured on the Point Cloud Using CloudCompare

Table 4.6. Dimensions of Example Spall Cluster Estimated by Projecting it on to the Planes Fit to the Bottom and Side Surfaces of Box Beam-3

Name of the Spall Cluster	Length Estimated in Two Dimensions, in	Width Estimated in Two Dimensions, in	Depth Estimated in Two Dimensions, in
Spall Cluster 0	527	24.73	6.42

4.3. Overlays and Strength Estimation

Textured mesh models of the beams with the damage highlighted served as 3D damage maps to be used along with the recommendations of Naito et al. (2011) to arrive at a reduced area of strands for strength estimation at a slice-section. Corrosion damage that has to be accounted for in moment capacity estimation at each slice-section, was determined using these recommendations. In this part of the report, the damage accounted for in the residual capacity estimation at each of the sections where failures occurred in the tests described in Section 3.1.4 is shown and discussion is presented to arrive at a total reduced area of prestressing strands.

In this part, firstly, slice obtained from the dense point cloud, closest to the location of failure in each of the tests overlaid on top of the design cross-section drawing of the beam is shown. Side-a and Side-b are marked on either side of this slice. Prestressing strands are numbered for convenience. Text file output by the program used to extract the slices is also shown to identify the damage clusters to be considered. Next, the portion of the textured mesh inside the inspection window along with the superimposed damage clusters is shown. Side-a and Side-b are marked on the window. Strands that were affected due to the damage clusters lying inside the inspection window for each of the test failure slice-sections and the total reduced area of strands for each of the failure sections in the tests are shown in a table for each of the cross-sections.

4.3.1. Aden Road Bridge Box Beam-3

4.3.1.1. Section of Failure in Test-2

The beam failed at a section 45.46 ft away from the left end (in Section 3.1.4.1.2.1) with rupture in the strands in the bottommost layer. Among the slices extracted from the beam, the slice that was the closest to this section of failure was Slice-18, taken at 45 ft from the left end. Overlay of Slice-18 with the design cross-section drawing of the beam is shown in Figure 4.25 with numbered strands, and Side-a and Side-b, marked. The development length of the strands in the two bottom layers was approximately 5.5 ft using Equation 3.1 in this thesis. Hence, an Inspection Window of two times 5.5 ft was chosen. Damage clusters that fall inside the Inspection Window for Slice-18 were identified automatically in the text-file, shown in Figure 4.26, output by the program that generated the slices. In Figure 4.27, a portion of the textured mesh of the Box Beam-3, that falls inside the Inspection Window for Slice-18 is shown. Side-a and Side-b in Figure 4.27 correspond to the Side-a and Side-b in Figure 4.25. The damage (spalls and cracks) points lying inside the Inspection Window for Slice-18 were projected on to the slice-section. The overlay (Figure 4.25) shows these points in red and blue colors in the bottom portion of the beam, which correspond to the spalls and cracks present inside the Inspection Window, respectively.

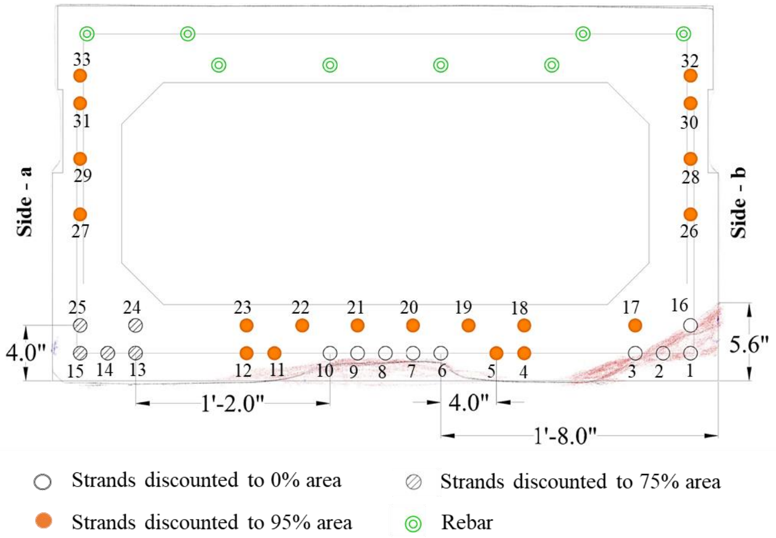
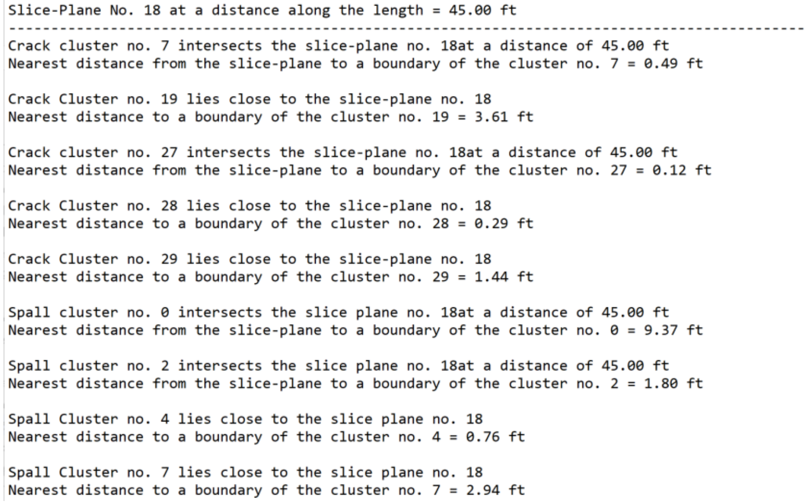


Figure 4.25. Overlay of Slice No. 18 (Near the Section of Failure in Test-2) from Box Beam-3 Dense Point Cloud with the Cross-Section Drawing



```
Slice-Plane No. 18 at a distance along the length = 45.00 ft
-----------------------------------------------------------------------------------------------
Crack cluster no. 7 intersects the slice-plane no. 18at a distance of 45.00 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 7 = 0.49 ft

Crack Cluster no. 19 lies close to the slice-plane no. 18
Nearest distance to a boundary of the cluster no. 19 = 3.61 ft

Crack cluster no. 27 intersects the slice-plane no. 18at a distance of 45.00 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 27 = 0.12 ft

Crack Cluster no. 28 lies close to the slice-plane no. 18
Nearest distance to a boundary of the cluster no. 28 = 0.29 ft

Crack Cluster no. 29 lies close to the slice-plane no. 18
Nearest distance to a boundary of the cluster no. 29 = 1.44 ft

Spall cluster no. 0 intersects the slice plane no. 18at a distance of 45.00 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 0 = 9.37 ft

Spall cluster no. 2 intersects the slice plane no. 18at a distance of 45.00 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 2 = 1.80 ft

Spall Cluster no. 4 lies close to the slice plane no. 18
Nearest distance to a boundary of the cluster no. 4 = 0.76 ft

Spall Cluster no. 7 lies close to the slice plane no. 18
Nearest distance to a boundary of the cluster no. 7 = 2.94 ft
```

Figure 4.26. The Portion of the Text File Showing the Damage Clusters Lying Inside the Inspection Window for Slice -18 of the Box Beam-3 Point Cloud

On the slice-18 overlay in Figure 4.25, spalled regions (red points) may be observed on the Side - b and in the middle portion of the section, and crack points (blue) are on Side – a and Side-b of the section. The corresponding spalls on Side-b and cracks on Side-a can be found on the textured mesh model of the beam in Figure 4.27. The damage clusters listed in Figure 4.26 can be seen superimposed on top of the textured mesh model in Figure 4.27. Each of the damage clusters was examined to determine the strands affected due to the cluster to come up with a reduced cross-sectional area for each strand based on the recommendations from Section 2.8. These observations are recorded in Table 4.7.

Crack C7 lay on Side – a, and cross-sectional areas of strand numbers 15 and 25 had to be reduced by 25%, as they were located close to the crack at a distance less than 3 in. Crack C19 on Side-a shifted transversely on the bottom of the beam, from one strand to another affecting Strands14, 15, and 25. As per the recommendations, strands in different layers that lie directly on a longitudinal crack had to be reduced to 75% of their cross-sectional areas. For the same reason, Crack C27, located beside the spall in the middle (S2), affected both strands 13 and 24. Distance from the edge of the point cloud to this crack was measured on CloudCompare to confirm that the strands in both the layers have to be reduced by 25%. Cracks C28 and C29 on Side – b affected Strand 16 in the second layer. However, the Spall S0 completely exposed strand 16 along with the Strands 1, 2, 3 on the bottommost layer. All these strands had to be neglected completely from the strength calculation. Similarly, the five exposed strands, 6, 7, 8, 9, 10 due to the Spall Cluster S2 were taken to 0% of their areas. Strand 10 was partially exposed and was still reduced to 0% to be conservative. Spall S4 was wrongly classified as a separate cluster from S2 by the clustering algorithm, and Spall S7 was observed to be noise. As per the recommendations, all other strands in the cross-section had to be reduced to 95%, as the beam had a significant amount of longitudinal cracking. Strands reduced to 0%, 75% and 95% of the areas are shown in Figure 4.25. After making all these reductions, the total reduced area of prestressing strands at this section is calculated in Table 4.8. The number of strands in each layer to be reduced to 0%, 75%, and 95% of the cross-sectional areas are also shown in this table.

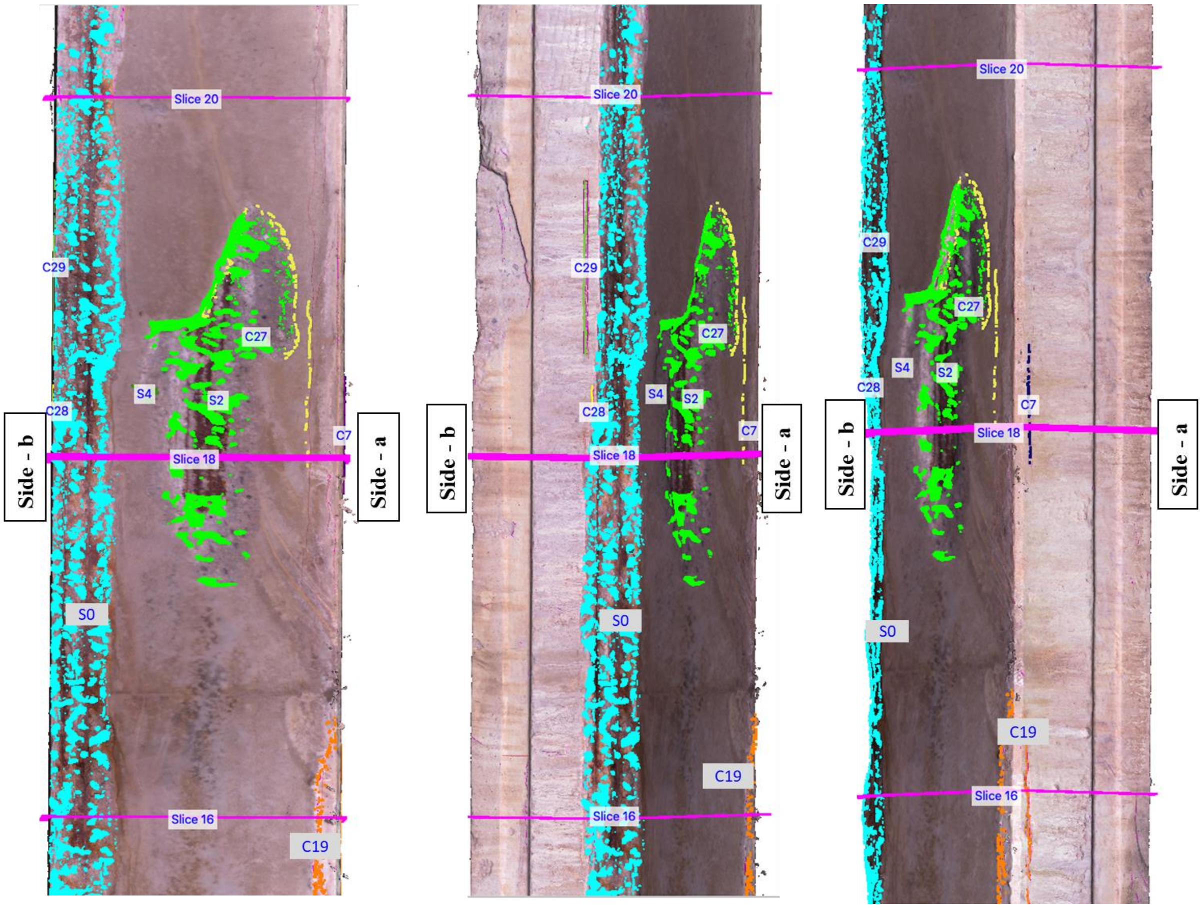


Figure 4.27. Damage Clusters in the Inspection Window to be Considered for Strength Estimation at Slice-18 of Box Beam-3

Table 4.7. Effect of the Damage Clusters in the Inspection Window on the Prestressing Strands at Slice-18 of Box Beam-3, as per the Recommendations of Naito et al. (2011)

Name of the Damage Cluster in the Inspection Window	Color of the Damage Cluster	Location of the Damage Cluster	Strands Affected due to the Damage Cluster	Exposure of the strands in the Inspection Window	Percentage of Area of each strand for Strength Calculations
Crack cluster 7 (C7)	Black	Side - a	15, 25	Unexposed	75%
Crack Cluster 19 (C19)	Orange	Side - a	14, 15, 25	Unexposed	75%
Crack cluster 27 (C27)	Yellow	Middle + Side - a	13, 24	Unexposed	75%
Crack Cluster 28 (C28)	Yellow	Side - b	16	Exposed	0%
Crack Cluster 29 (C29)	Green	Side -b	16	Exposed	0%
Spall cluster 0 (S0)	Cyan	Side -b	1, 2, 3, 16	Exposed	0%
Spall cluster 2 (S2)	Green	Middle	6, 7, 8, 9, 10	Exposed (Strand 10 is slightly exposed)	0%

Spall Cluster 4 (S4)	Dark Green	Should have been a part of Spall Cluster 2		
Spall Cluster 7 (S7)	-	Not considered as it was noise		
-	-	-	Other unexposed strands unaffected due to cracks	95%

Table 4.8. Calculation of Reduced Area of Strands for Residual Strength Estimation at Failure Section in Test-2 on Box Beam-3

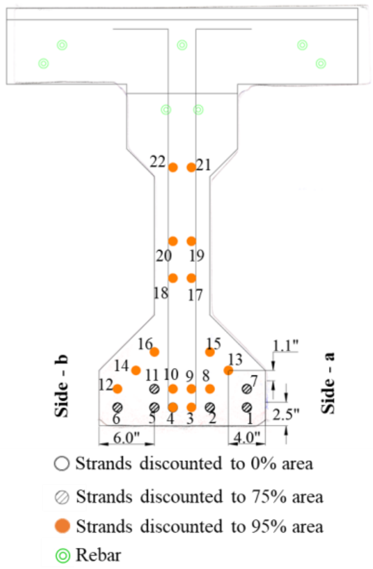
Strand Pattern: All strands are 7/16 in. dia. Stress Relieved Grade 270						
Area of each Strand (A_{ps}): 0.115 in^2						
Row	**Distance from the bottom to each layer (in)**	**Number of strands in each layer**	**No. of strands discounted to 0% area**	**No. of strands discounted to 75% area**	**No. of strands discounted to 95% area**	**Area of Strands in Each Layer (in^2)**
1	2	15	8	3	4	0.696
2	4	10	1	2	7	0.937
3	12	2	0	0	2	0.219
4	16	2	0	0	2	0.219
5	20	2	0	0	2	0.219
6	22	2	0	0	2	0.219
TOTAL		33				2.507
Modified Centroid of the Strands: 8.15 in.						

As mentioned before, a total of 22 slices were extracted from the dense cloud of Box Beam-3 at locations shown in Figure 3.9. The results of the estimation of strength at slice-sections other than Slice No. 18 are demonstrated in Appendix-A. In the Test-1 on Box Beam-3, the shear failure was initiated at a section 4.83 ft from the left end, which was close to Slice No. 2 extracted at 5 ft from the left end. Results of the corrosion damage affecting the strength at this section are also presented in Appendix-A along with the other slices.

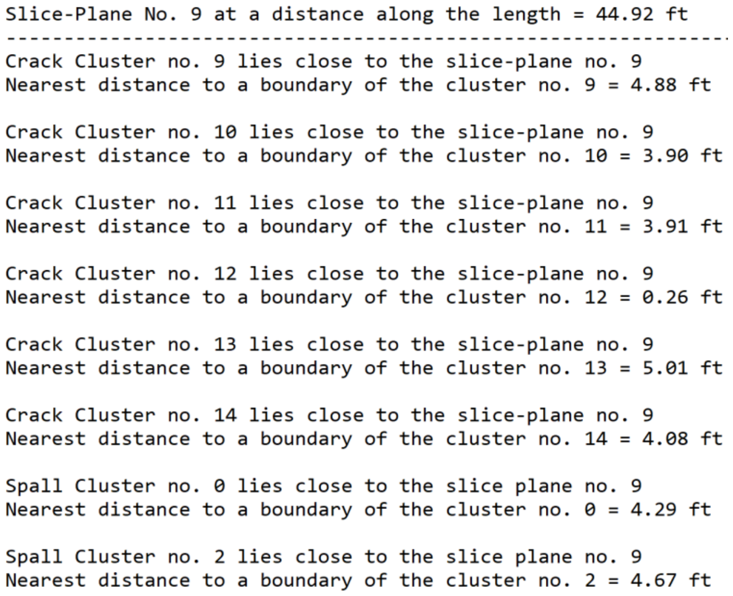
4.3.2. Lesner Bridge I-Beam 9

4.3.2.1. Section of Failure in Test-2

The shear failure initiated at the point of loading at a section 43.07 ft away from the left end (described in Section 3.1.4.2.2.1). Among the slices extracted from the beam, the slice that was the closest to this section of failure was Slice-9, taken at 44.92 ft from the left end. Overlay of Slice-9 with the design cross-section drawing of the beam is shown in Figure 4.28a with numbered strands, and Side-a and Side-b, marked. Development Length of the strands in the two bottom layers was approximately 5.5 ft using Equation 3.1. Hence, an Inspection window of 2 times 5.5 ft was chosen. Damage clusters that fall inside the Inspection Window for Slice-9 were identified in the text-file shown in Figure 4.28b. In Figure 4.30, a portion of the textured mesh of the Box Beam-3, that falls inside the Inspection Window for Slice-9 is shown. Side-a and Side-b in Figure 4.30 correspond to the Side-a and Side-b in Figure 4.28a. The damage (spalls and cracks) points lying inside the Inspection Window for Slice-9 were projected on to the slice-section. The overlay (Figure 4.28a) shows these points in red and blue colors in the bottom portion of the beam, which correspond to the spalls and cracks present inside the Inspection Window, respectively.


22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
1.1"
2.5"
6.0"
4.0"
Side - b
Side - a
○ Strands discounted to 0% area
⊘ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

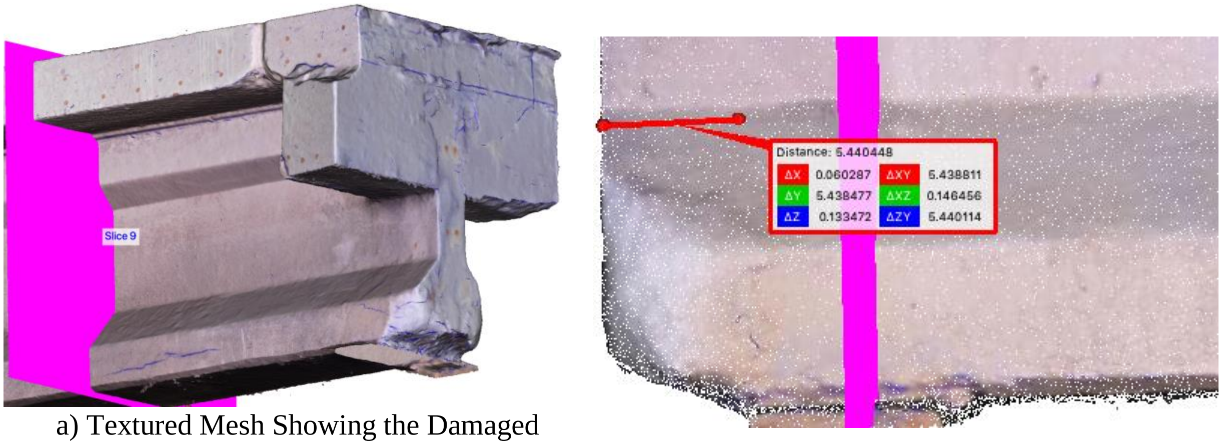
a) Overlay of Slice No. 9 with the Cross-Section Drawing near the failure section


```
Slice-Plane No. 9 at a distance along the length = 44.92 ft
---------------------------------------------------------------
Crack Cluster no. 9 lies close to the slice-plane no. 9
Nearest distance to a boundary of the cluster no. 9 = 4.88 ft

Crack Cluster no. 10 lies close to the slice-plane no. 9
Nearest distance to a boundary of the cluster no. 10 = 3.90 ft

Crack Cluster no. 11 lies close to the slice-plane no. 9
Nearest distance to a boundary of the cluster no. 11 = 3.91 ft

Crack Cluster no. 12 lies close to the slice-plane no. 9
Nearest distance to a boundary of the cluster no. 12 = 0.26 ft

Crack Cluster no. 13 lies close to the slice-plane no. 9
Nearest distance to a boundary of the cluster no. 13 = 5.01 ft

Crack Cluster no. 14 lies close to the slice-plane no. 9
Nearest distance to a boundary of the cluster no. 14 = 4.08 ft

Spall Cluster no. 0 lies close to the slice plane no. 9
Nearest distance to a boundary of the cluster no. 0 = 4.29 ft

Spall Cluster no. 2 lies close to the slice plane no. 9
Nearest distance to a boundary of the cluster no. 2 = 4.67 ft
```

b) The Portion of the Text File Showing the Damage Clusters Inside the Inspection Window for Slice No.9

Figure 4.28. Tools for Strength Estimation at Section of Failure in Test-2 on Lesner Bridge I-Beam 9

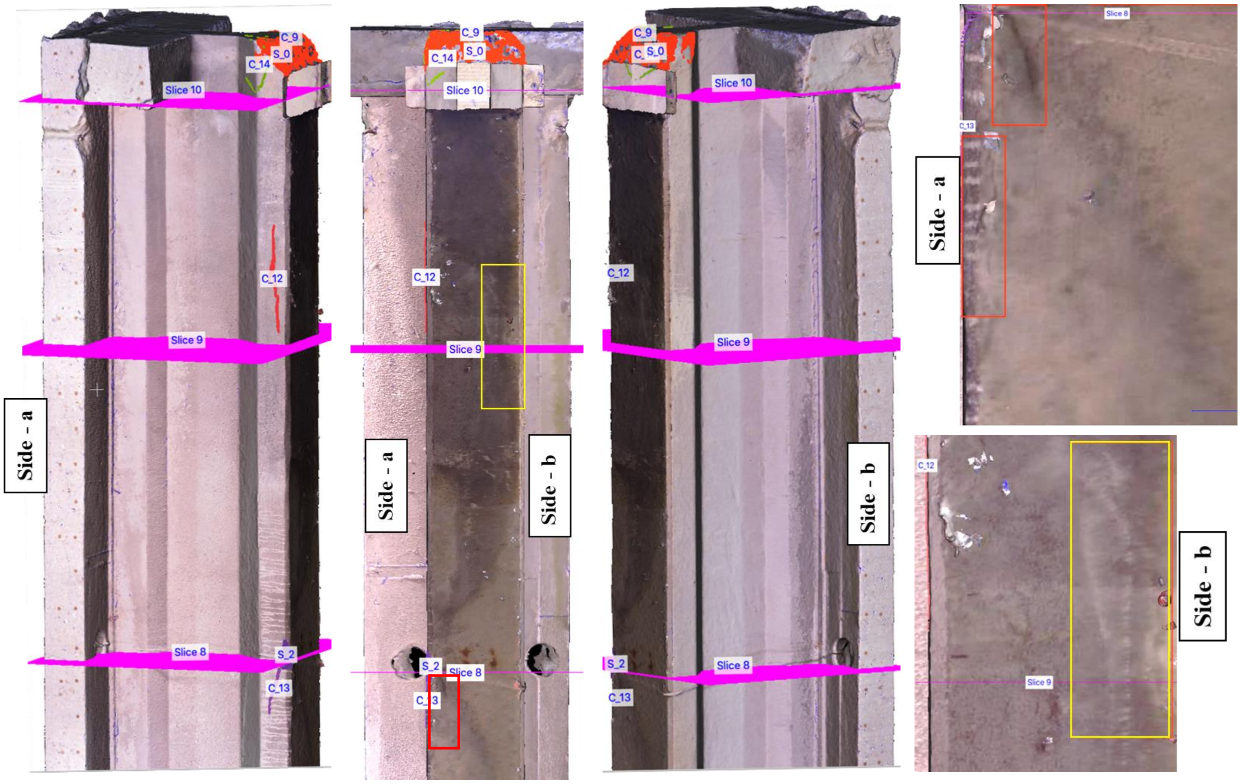
On the Slice No. 9 overlay in Figure 4.28a, spalled regions (red points) from the spall at the end of the beam in Figure 4.30 were barely seen in the middle portion of the section, and crack points (in blue) are on Side - a of the section. The distance (2.5 ft) from the bottom of the beam to some of the projected blue points on Side-a is shown in Figure 4.28a. In this research, the damage at the end of the beam was accounted for by altering the transfer length and development length of the prestressing steel. The region at the nearest end to the slice-section appeared to be patched with repair concrete and the surface looked like a spall which can be observed on the textured mesh shown in Figure 4.29a. This damage was captured in the spall detection as shown in Figure 4.19f . The depth of this spall was measured as 5.5 in. on the dense point cloud in CloudCompare which can be seen in Figure 4.29b due to this spall, the bottom four layers of strands were considered to have zero stress up to 5.5 in. The stress in the strands was assumed to develop from zero at 5.5 in. to f_{ps} at one development length distance. Reduction in strand cross-sectional area inside the Inspection Window was not considered due to this end damage. In Figure 4.30, all the detected damage clusters on Side-a and Side-b of the beam are shown on the portion of the textured mesh of the beam that falls inside the Inspection window for Slice No. 9. Each of the damage clusters was examined to determine the strands affected due to the cluster to come up with a reduced cross-sectional area for each strand based on the recommendations from Section 2.8. These observations are recorded in Table 4.9.



a) Textured Mesh Showing the Damaged (Patched) Region at the end of I-Beam 9 Nearest to Slice No. 9

b) Depth of the Spall at the End Measured on the Dense Point Cloud

Figure 4.29. End Region Near Slice No. 9 Shown on the Textured Mesh of I-Beam 9



□ Undetected Crack located at about 6 in. from the edge on Side-b □ Undetected Cracks at Slice-8

Figure 4.30. Damage Clusters in the Inspection Window to be Considered for Strength Estimation at Slice-9 of I-Beam 9

From Figure 4.30, the effect of each damage cluster is identified. Crack C9 and Spall S0 were a part of the end damage and were already accounted for as explained above. Cracks C10 and C11 lie on the end diaphragm and did not affect the strength of the beam. C12, C13, and C14 lay on Side – a of the beam, and cross-sectional areas of strand numbers 01 and 07 were reduced by 25%, as they were located close to the cracks at a distance less than 3 in. It was assumed that Strand 13 in the third layer was not affected due to C13 and C14 as it was located at a distance of 4 in. from the nearest side surface of the bottom flange as shown in Figure 4.28a. Spall S2 located at the same location as the Crack C13, was observed to be too small to affect any strands. In addition to these clusters, some cracks went undetected at Slice-8 (highlighted with a bounding box in Figure 4.30) and were measured to be 3.3 in. from the Side-a surface. This affected strands 1 and 2, and so, Strand 2 was taken to 75%. There were two undetected cracks at the location of Slice-9 on Side-b highlighted with bounding boxes in Figure 4.30. One of these cracks was observed to be less than 3 in. from the surface (Side-b) and the second undetected crack was observed to be shifting transversely on the bottom of the beam, from 3 in. from the surface (Side-

b) to more than 6 in. from the surface from distance measurements on the dense point cloud using CloudCompare. These cracks were considered to affect strands 5, 6, and 11. Strand 11 was reduced because the recommendation suggested reducing the strands in different layers that lie directly on a longitudinal crack to 75% of their cross-sectional areas. Also, all other strands in the cross-section were reduced to 95% following the recommendations. Strands reduced to 0%, 75%, and 95% of the areas are shown in Figure 4.28a. After making all these reductions, the total reduced area of prestressing strands at this section is calculated in Table 4.10. Number of strands in each layer to be reduced to 0%, 75%, and 95% of the cross-sectional areas are shown in Table 4.10 and in Figure 4.28.

Table 4.9. Effect of the Damage Clusters in the Inspection Window on the Prestressing Strands at Section of Failure in Test-2 of I-Beam 9, as per the Recommendations of Naito et al. (2011)

Name of the Damage Cluster in the Inspection Window	**Location of the Damage Cluster**	**Strands Affected due to the Damage Cluster**	**Exposure of the strands in the Inspection Window**	**Percentage of Area of each strand for Strength Calculation**
C9, S0	At the end of the beam	Effect on the strength due to end damage clusters was already accounted for	-	-
C10, C11	End diaphragm	Do not affect the strength	-	-
C12, C13, C14	Side-a	1, 7 It was assumed that Strand 13 was not affected due to C13 and C14	Unexposed	75%
S2	Side-a	Does not affect any strands	-	-
Undetected Crack at Slice-8	Side-a	1, 2	Unexposed	75%
Undetected Cracks (Slightly Visible on Texture)	Side-b to Middle	5, 6 and 11	Unexposed	75%
-	-	Other unexposed strands unaffected due to cracks		95%

Table 4.10. Calculation of Reduced Area of Strands for Residual Strength Estimation at Failure Section in Test-2 on I-Beam 9

Strand Pattern: All strands are 7/16 in. dia. Stress Relieved Grade 270						
Area of each Strand (A_{ps}): 0.117 in^2 (From Design Drawing)						
Row	**Distance from the Bottom to Each Layer (in)**	**Number of strands in Each Layer**	**No. of strands discounted to 0% area**	**No. of strands discounted to 75% area**	**No. of strands discounted to 95% area**	**Area of Strands in Each Layer (in^2)**
1	2	6	0	4	2	0.573
2	4	6	0	2	4	0.620
3	6	2	0	0	2	0.222
4	8	2	0	0	2	0.222
5	16	2	0	0	2	0.222
6	20	2	0	0	2	0.222
7	28	2	0	0	2	0.222
TOTAL		22				2.305
Modified Centroid of the Strands: 9.1 in.						

A total of 10 slices were extracted from the dense cloud of I-Beam 9 at locations. At all these slice-sections, reduced cross-sectional areas of the strands and the modified centroids of strands were determined after accounting for corrosion damage as explained in this section. These were used to estimate the flexural capacity of I-Beam 9 at these slice-sections. The reduced areas of strands and the results of the estimation of strength at slice-sections other than Slice No. 9 are demonstrated in Appendix-B. In the Test-1 on I-Beam 9, the flexural failure with crushing in the deck occurred at a section 8.83 ft from the left end, which was close to Slice No. 2 extracted at a distance of 9.98 ft from the left end. Results of the corrosion damage affecting the strength at this section are also presented in the Appendix-B along with the other slices.

4.4. Comparison of Computed Moment Capacities with the Demand from Tests

This section compares the positive moment capacity calculations for Aden Road Bridge Box Beam-3 and Lesner Bridge I-Beam 9, at the sections where failures occurred in the tests on these beams (described in Section 3.1.4), with the maximum moment demands from the tests. These three-point bending tests were conducted at the ends of the beams with an intention to determine the shear capacity of the beams. However, in only two of these tests, failures occurred due to exceeding the shear capacity of the beams. In this research, only the ultimate positive moment capacities of the beams at the slice-sections are estimated and compared to the maximum moment demands from the tests. Comprehensive analysis to arrive at the shear capacities of these beams using different approaches were done by Al Rufaydah (2020) and are not discussed in this report. Maximum moment demand at the section where the failure occurred and the type of failure observed in each of these tests are summarized in Table 3.3.

4.4.1. Comparison for Aden Road Bridge Box Beam-3

The test setup, loading protocol, procedure, and the failure modes in these tests are described in Section 3.1.4.

4.4.1.1. Comparison at the Section of Failure in Test-1

The end of the beam at which external load was applied in this test, failed in a shear failure. This is discussed in Section 3.1.4.1.1.1. The maximum moment demand (at a distance of 4.83 ft from the left end) including self-weight moment to cause this shear failure was 1029 kip-ft. The capacities at the slice-sections calculated using Strain Compatibility analysis using CON-WHT-THR-R material parameters described in Section 3.1.3.7.1 (with a Whitney stress block model for compressive stress distribution in concrete) following the recommendations of Naito et al. (2011) and the moment demand along the length of the beam corresponding to the maximum external load and self-weight moment are shown in Figure 4.31. The moment capacity of the beam calculated at the location of the point load in this test is summarized in Table 4.11.

Table 4.11. Moment Capacity Calculated at the section of failure in Test-1 on Aden Road Bridge Box Beam-3

Location of Strength Estimation	Strain Compatibility Analysis	Ultimate Strain in bottommost layer prestressing strands (ε_{ps})	Ultimate Strain in Concrete (ε_{cu})	Moment Capacity Calculated, kip-ft	Moment Demand at Failure, kip-ft
At the section of failure	CON-WHT-THR-R	0.0216	0.003	1255	1029

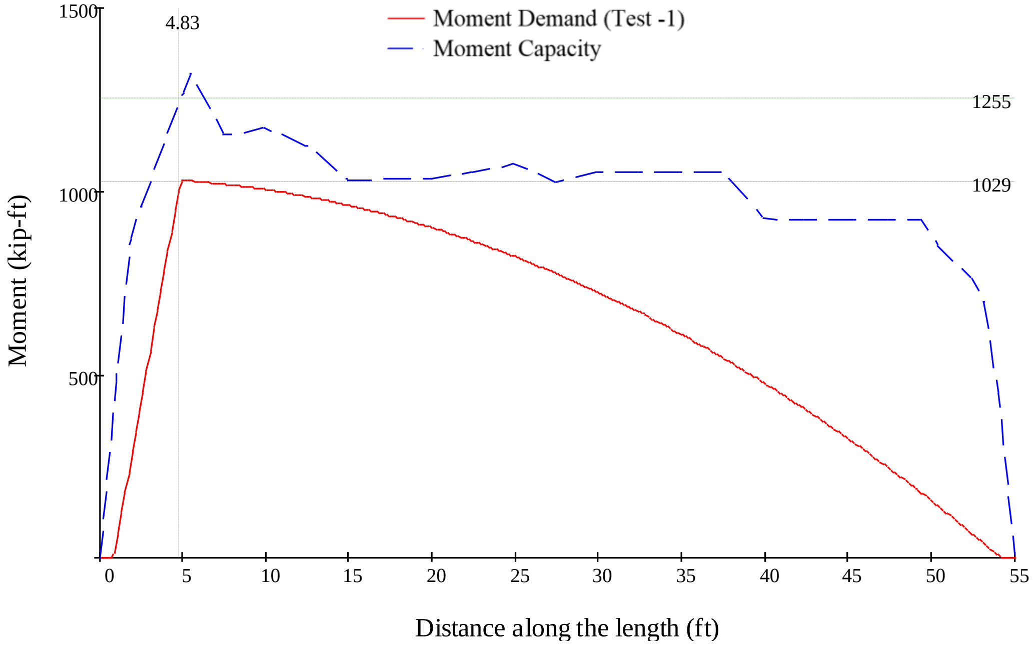


Figure 4.31. Moment Capacity (Strain Compatibility analysis with CON-WHT-THR-R) vs. Demand Comparison at the section of failure in Test-1 on Aden Road Bridge Box Beam-3

4.4.1.1.1. Observation

The moment capacity of the beam in positive bending at the section of the point load was 1255 kip-ft. It was observed that the flexural capacity of the beam at this section was higher than the shear demand. This caused the beam to fail in shear before it reached its ultimate moment capacity. Analysis for estimating the shear capacity is done by Al Rufaydah (2020).

4.4.1.2. Comparison at the Section of Failure in Test-2

The beam failed at a section 45.46 ft away from the left end with rupture of strands in the bottommost layer. The maximum moment demand including the self-weight moment at the failure section was 848.6 kip-ft. The total moment demand at the section (at a distance of 50.42 ft from the left end) where the point load was applied was 896.3 kip-ft. The positive moment capacities at the slice-sections calculated using Strain Compatibility analysis with CON-WHT-THR-R material parameters described in Section 3.1.3.7.1, following the recommendations of Naito et al. (2011) and the moment demand along the length of the beam, corresponding to the maximum external load and self-weight moment are shown in Figure 4.32. The moment capacities calculated at these two locations are summarized in Table 4.12.

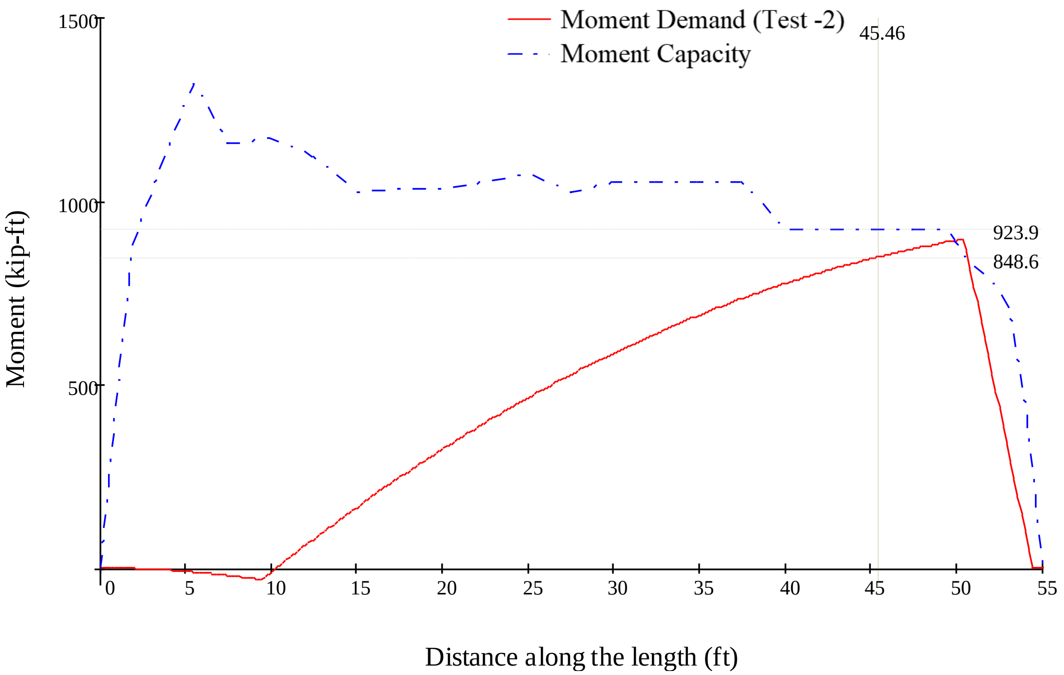


Figure 4.32. Moment Capacity (Strain Compatibility Analysis with CON-WHT-THR-R) vs. Demand Comparison at the section of failure for Test-2 on Aden Road Box Beam-3

Table 4.12. Moment Capacity Calculated at the section of failure and the section of loading in Test-2 on Aden Road Bridge Box Beam-3

Location	**Strain Compatibility Analysis Material Parameters**	**Ultimate Strain in bottommost layer prestressing strands (ε_{ps})**	**Ultimate Strain in Concrete (ε_{cu})**	**Moment Capacity Calculated, kip-ft**	**Moment Demand at Failure, kip-ft**
At strand rupture failure	CON-WHT-THR-R	0.0257	0.003	923.9	848.6
At the point load	CON-WHT-THR-R	0.027	0.003	848	896.3

4.4.1.2.1. Observations at the Section of Failure

The moment capacity calculation with CON-WHT-THR-R using the recommendations proposed by Naito et al. (2011) predicted the beam to have more flexural capacity than the demand from the test (actual ultimate flexural capacity) at the section of failure. The calculation overpredicted the actual moment capacity of the beam at the section of failure. Strain compatibility analysis with CON-WHT-THR-R material parameters considers concrete crushing with an ultimate strain (ε_{cu}) of 0.003 in the extreme compression fiber of the section, as the failure mechanism. As the mode of failure observed in the test was due to the rupture of multiple strands in the bottommost layer before the concrete crushed, the calculation using CON-WHT-THR-R did not capture the actual failure mode observed in the test.

To represent the actual mode of failure, strain compatibility analysis was done using STL-HOG-THR-R and STL-HOG-THR-ORG parameters, choosing a prescribed value for ultimate strain in the bottom layer of prestressing strands (ε_{ps}) as described in Sections 3.1.3.7.2 and 3.1.3.7.3, and ultimate flexural capacity of the beam at this section was calculated. The objective of this analysis was to arrive at an estimate of the actual strain in the bottommost layer of strands at this section that caused them to rupture during the test. The analysis was repeated by varying the value of prescribed strain in the bottommost layer of strands to obtain a value of the moment capacity that matches with the maximum moment demand that caused the strand rupture during the test.

From the strain compatibility analysis with STL-HOG-THR-R (with strands reduced in area as per Naito's recommendations), the predicted ultimate strain in the strands in the bottommost layer was 1.305% to reach a moment capacity of 848.5 kip-ft (the maximum total moment demand from the test). The corresponding depth to the Neutral axis from the top of the beam at this ultimate capacity was 4.16 in. The results of these calculations are summarized in Table 4.13. As per strand area reduction described in Section 4.3.1.1, the strands with numbers 4, 5, 11, and 12 in Figure 4.25 were reduced to 95% of their cross-sectional area in the strength calculations. After the test, the failure section on the beam was examined to verify the actual condition of the strands that were unexposed prior to the test. It was observed that the unexposed strands (shown in Figure 4.33) near the spalling had a degree of corrosion that can be categorized as Heavy Pitting (DI = 3) as per the levels of corrosion damage in prestressing strands proposed by Naito et al. (2011), described in Section 2.8. As per the tension tests on individual wires separated from corroded prestressing strands from decommissioned bridge girders conducted by Naito et al. (2006), a Heavy Pitting (DI = 3) type of strand corrosion damage was shown to reduce the strength of the strand by almost 28%. As per the actual condition of these strands found after testing, instead of considering 95% of their areas, the strain compatibility analysis (STL-HOG-THR-R) was repeated by reducing the areas of strands 4, 5, 11, 12 to 75% of their cross-sectional areas. This analysis increased the previously computed strain in the bottommost layer strands at fracture, to 1.61% to develop a moment capacity of 848.8 kip-ft that closely matches with the maximum total demand from the test. The corresponding depth to the Neutral axis from the top of the beam was 3.66 in. From this analysis, the predicted ultimate strain in the bottommost layer of strands was estimated to be in the range of 1.305% to 1.61%.

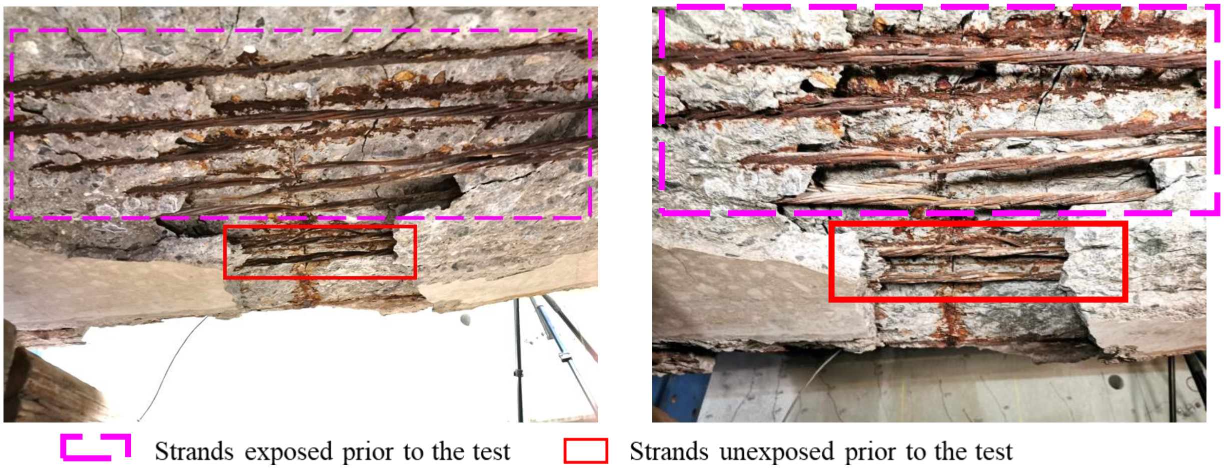

Strands exposed prior to the test
Strands unexposed prior to the test

Figure 4.33. Failure Section in Test-2 on Box Beam-3 Showing Strands Unexposed before the Test

In the strain compatibility analysis with STL-HOG-THR-R, the strands were reduced in cross-sectional area to account for the corrosion damage along with limiting the ultimate strain in the prestressing steel, assuming that the prestressing steel contributed a partial cross-sectional area all along in the analysis until their rupture. The strain compatibility calculation was repeated using STL-HOG-THR-ORG material parameters with the full (undamaged) areas of strands. The objectives of this exercise were to verify the previously calculated value of the ultimate strain in the bottommost layer prestressing strands and to study the cross-sectional area of strands contributing to the capacity of the beam, in the portion of the load-deflection curve of the test before strand rupture. In this analysis, a value of ultimate strain was prescribed to capture the failure occurred by strand rupture. Using the original (undamaged) area of strands, the ultimate strain in the bottommost layer of prestressing strands was calculated as 0.63% to give an ultimate flexural capacity (848.5 kip-ft) that matches the total moment demand from the test. At this value of ultimate strain in the prestressing steel, the depth to the neutral axis from the top fiber was calculated as 15.77 in. as mentioned in Table 4.13. This ultimate value of strain was less than 50% of the previously calculated estimates of ultimate strain using STL-HOG-THR-R material parameters. To check which material parameters used in the strain compatibility analysis gave a better prediction of the ultimate strain in the prestressing strands at fracture, observations from the test and previous tension tests on strands extracted from corrosion damaged box beams and I-Beams, conducted by Alfailakawi (2019) described in Section 2.9 were used.

Table 4.13. Results of Strain Compatibility Analysis with Various Material Parameters for the Section of Failure in Test-2 on Box Beam-3

Strain Compatibility Analysis Material Parameters	**Ultimate Strain in bottommost layer prestressing strands (ε_{ps})**	**Ultimate Strain in Concrete (ε_{cu})**	**Depth of Neutral Axis at Ultimate Capacity, in**	**Moment Capacity Calculated, kip-ft**	**Maximum Moment Demand, kip-ft**
CON-WHT-THR-R	0.0257	0.003	3.28	923.9	848.6
STL-HOG-THR-R	0.01305	0.00144	4.16	848.5	
STL-HOG-THR-R with unexposed strands at 75% area	0.0161	0.00176	3.66	848.8	
STL-HOG-THR-ORG	0.0063	0.00059	15.77	848.5	

4.4.1.2.1.1. Observations from the Flexural Cracking Pattern from the Test

Observations from the Test-2 were investigated to validate the value of the ultimate strain. Firstly, the Neutral axis depth corresponding to an ultimate strain value of 0.63% in the prestressing steel obtained when full cross-sectional areas of strands were used, was 15.77 in. The total depth of the cross-section was 27 in. Secondly, the cracking pattern on the beam shown in Figure 4.34 was observed. This picture was captured at the end of the test whereas, the ultimate strain in the strands was being calculated at the maximum applied load of 230.5 kip before the load started dropping (Refer to the load-deflection curve in Figure 3.21). Marking the cracks on the beam and writing the load at which they propagated were stopped at 215 kip because of safety reasons. From these markings on the beam in Figure 4.34, it could be noticed that the cracks at the end of the test propagated beyond the cracks marked at a load of 215 kip. The goal was to arrive at an estimate of the Neutral axis depth from the cracking pattern. As the crack propagation at a load of 230.5 kip was not recorded, this could only be estimated by eyeballing. At the end of the test, it was observed that the flexural cracks propagated into the top flange. One of the cracks had propagated into the top flange up to a distance of 4.4 in. from the top of the beam at the end of the test. The distance of the tip of the same crack at the load of 215 kip from the top was about 9 in. The crack should have propagated to anywhere between 4.4 in and 9 in. from the top at the load of 230.5 kip. Flexural cracking occurs where the tensile capacity of concrete is exceeded. As concrete has some tensile capacity, the Neutral axis cannot be located exactly at the tip of the crack, but it would be located at some distance away from the tip of the crack. From this exercise, it can be concluded that the Neutral axis depth at the load of 230.5 kip could be approximately between 4 in. to 8.5 in. By comparing this estimated Neutral axis depth to the calculated value of 15.77 in, it can be concluded that using a full cross-sectional area of strands in the analysis with STL-HOG-THR-ORG gave a wrong prediction of the ultimate strain in the prestressing steel.

On the other hand, using a reduced strand cross-sectional area that predicted a bottommost layer ultimate strain of 1.305% gave a Neutral axis depth of 4.16 in. This could have been closer to the actual value of the ultimate strain. Furthermore, reducing the cross-sectional area of strands further to incorporate the actual condition of the unexposed strands in Figure 4.33 predicted an ultimate strain of 1.61% but reduced the Neutral axis depth to 3.66 in. This value of Neutral axis depth is a little less when compared to the estimated value of 4 in. to 8.5 in. Hence, the estimated ultimate strain of 1.61% could be slightly on the higher side. Considering the large range of estimated Neutral axis depth of 4 in. to 8.5 in., it is possible that the actual value of the ultimate strain was slightly lower than 1.305%. It is also possible that the strands are actually contributing a cross-sectional area slightly greater than 2.507 in^2 that was considered in this calculation as mentioned in Table 4.8. To sum up, it can be understood that the strands contributed a cross-sectional area that is less than the full or undamaged area even in the initial portion of the load-deflection of the curve until it ruptured. This could have also resulted in a reduction in the stiffness of the beam in the initial region of the load-deflection curve. This has to be investigated further using finite element analysis.

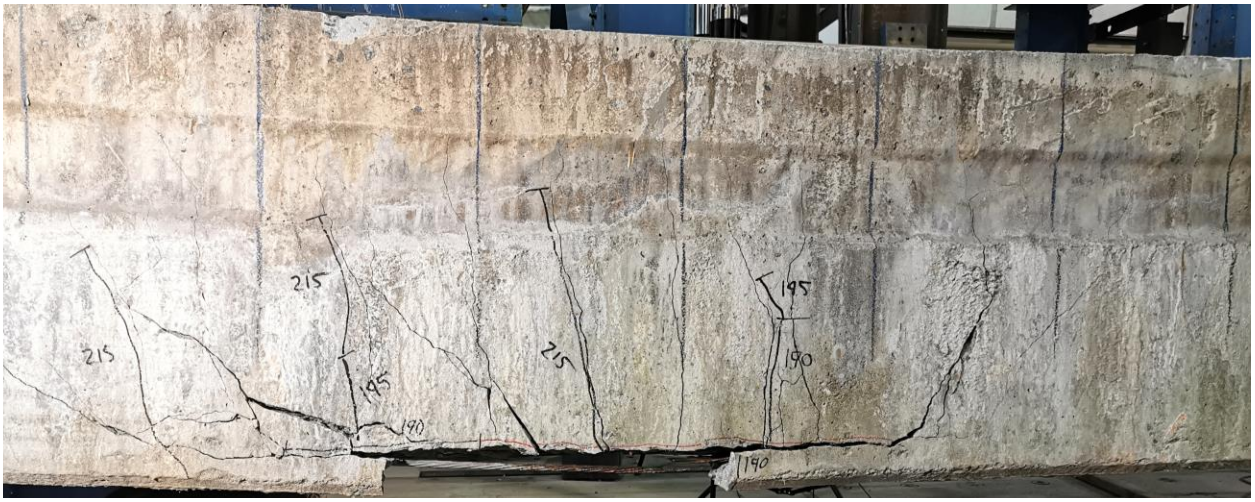


Figure 4.34. Region of Failure in Test-2 on Box Beam-3 from Al Rufaydah (2020)

4.4.1.2.1.2. Observations from Other Tests

In Section 2.8, the results of tension tests conducted by Naito et al. (2006) on individual wires separated from strands collected from corrosion damaged box beams are presented. The average wire capacities are presented for each degree of corrosion in Table 2.1. In Test -2 on Box Beam-3, the strands in the bottommost layer were observed to be at different degrees of corrosion. As discussed in Section 4.3.1.1, the strands 1, 2, 3 were completely exposed and fractured. Strands 6, 7, 8, 9, 10 were exposed due to the spalling in the middle and were in a Wire Loss type of damage. Strands 4, 5, 11, and 12 were unexposed before testing. Upon visual inspection of the photographs taken post-testing (shown in Figure 4.33), all unfractured strands in the bottommost layer were found to be in either a Heavy Pitting or Wire Loss level of corrosion.

Among the strand samples tested by Alfailakawi (2019) (shown in Section 2.9), samples from I-Beam 6 and Box Beam-2 were observed to have a Heavy Pitting level of corrosion. From Table 2.2, it can be seen that these strands failed at ultimate strains of 0.713% and 0.844%, respectively. It can be observed from the experimental stress-strain curves of these strand samples shown in Figure 2.10a and Figure 2.9a, that both these strand samples had fractured before yielding. The relative capacities of these strand samples compared to the maximum strand capacity observed from these tension tests (strength of the sample extracted from I-Beam 3) are calculated in Column 3 of Table 4.14. These relative strand capacities from Column 3 of Table 4.14 are compared to the relative average capacities of wires with Pitting and Heavy Pitting types of corrosion with respect to that of Light Corrosion, reported by Naito et al. (2006) (shown in column 3 of Table 2.1). From this comparison, a prediction of the level of corrosion is made for each strand sample. It should be noted that the relative strengths reported by Naito et al. (2006) are presented for individual wires and not for individual strands. From this comparison shown in Table 4.14, it can be observed that predictions of levels of corrosion by comparing the relative strand capacities with the relative wire strengths agree fairly well with the visual predictions from the photographs of the strand samples tested by Alfailakawi (2019) except for the sample from Box Beam-2. From this discussion, an important takeaway is that the ultimate stresses and strains from tension tests of strands can be attributed to different levels of strand corrosion.

Table 4.14. Comparison of Ultimate Strand Capacities from the tests conducted by Alfailakawi (2019) with Relative Average Wire Strengths by Naito et al. (2006)

Girder from which the Strand was Extracted	**Ultimate Strength, ksi**	**Relative Strength to the Strength of the Strand Sample from I-Beam 3, %**	**Ultimate Strain, %**	**Prediction based on Relative Wire Strengths by Naito et al. (2006)**	**Visual Prediction from Photographs**
I-Beam 6	188.5	69.3	0.713	Worse than Heavy Pitting	Heavy Pitting
Box Beam 2	224	82.3	0.844	Pitting	Heavy Pitting
Box Beam 1	269	98.9	1.675	Light Corrosion	Light Corrosion
I-Beam 7	261.5	96.1	3.462	Worse than Light Corrosion, No Pitting	Light Corrosion
I-Beam 3	272	100	6.122	Light / No Corrosion	No Corrosion

From strain compatibility analysis with STL-HOG-THR-ORG (with the original area of strands), the ultimate stress in the bottommost layer strands corresponding to a failure strain of 0.63% was found to be 176.4 ksi. This stress when compared to the relative wire strengths (Column 3 of Table 2.1) by Naito et al. (2006), predicts a level of corrosion that is worse than Heavy Pitting as mentioned in Table 4.15. These ultimate stresses indicate that the strands which failed in the bottommost layer ruptured before yielding. The comparison with relative average wire strength shows that the ultimate strand stress calculated using the undamaged area of strands gave a predicted level of corrosion that was close to the actual level of corrosion in strands observed in the photographs. Also, an ultimate strain of 0.63% calculated using strain compatibility analysis with the STL-HOG-THR-ORG (using the original area of strands), is closer to the ultimate strains of 0.713% and 0.844% in the strand samples that exhibited Heavy Pitting type of corrosion in the tests conducted by Alfailakawi (2019).

On the other hand, using a reduced area of strands predicted an ultimate strain of 1.305% to 1.61% in the bottommost layer of strands. The corresponding stress in the strands was 242 ksi to 250.5 ksi, which indicated that the strands had reached a post yielding stage. Furthermore, this stress corresponds to a level of corrosion that is worse than Light Corrosion but not Pitting type, when compared to the relative average wire strength by Naito et al. (2006). On the contrary, visual inspection of the photographs of the strands shown in Figure 4.33, indicated that type of corrosion to be Wire Loss in the exposed strands and Heavy Pitting in the unexposed strands. In Section

4.4.1.2.1.1, it was concluded that using a reduced cross-sectional area of prestressing strands in the analysis with STL-HOG-THR-R gave an estimate of the ultimate strain in the bottommost layer of strands as 1.305% which could be close but slightly higher when compared to the actual value of ultimate strain. Hence, there is a mismatch between the conclusions drawn using the discussions presented in Sections 4.4.1.2.1.1 and 4.4.1.2.1.2. Further analysis has to be conducted to reach a more accurate value of the ultimate strain in the bottommost layer prestressing strands. These observations are summarized in Table 4.15.

Table 4.15. Prediction of Level of Strand Corrosion in the Bottommost Layer of Box Beam-3

Strain Compatibility Analysis Material Parameters	**Ultimate Stress in Strand (f_{ps}), ksi**	**Relative Strength to the Strength of the Strand Sample from I-Beam 3, %**	**Ultimate Strain, %**	**Prediction based on Relative Wire Strengths by Naito et al. (2006)**	**Visual Prediction from Photographs**
STL-HOG-THR-R	242	89	1.305	Worse than Light Corrosion, No Pitting	Wire Loss and Heavy Pitting
STL-HOG-THR-R with unexposed strands at 75% area	250.5	92.1	1.61	Worse than Light Corrosion, No Pitting	
STL-HOG-THR-ORG	176.4	64.8	0.63	Worse than Heavy Pitting	

Finally, it can be observed from the load-deflection curve for Test-2 shown in Figure 3.21 that, the load increases for a short period post cracking. Then, the load starts dropping which indicates the onset of strand rupture in the bottommost layer. From this behavior, it is possible that the strands in the beam had barely yielded. This could mean that the actual ultimate strain in the bottommost layer corroded strands (stress-relieved strands) should have been close to the yield strain of about 0.82% to 0.85%, depending on the actual modulus of elasticity. All these observations emphasize the need for further analysis in determining the ultimate strain corresponding to the heavily corroded state of the strands.

In this direction, more tension tests are to be done on prestressing strands with different levels of corrosion to map ultimate strains with the levels of corrosion observed on the prestressing strands. From such experimental testing, a good estimate of the average ultimate strain can be found for each level of strand corrosion. Using these estimates of the ultimate strain in the bottommost layer of strands, if the corrosion damage in a bridge beam is found to be similar to that observed at the failure section of Box Beam-3 (shown in Figure 4.33), strain compatibility analysis performed with STL-HOG-THR-R material parameters is likely to give a better approximation of ultimate flexural capacity than the analysis with CON-WHT-THR-R. The former also captures the failure mechanism of the strands in a better way. However, more testing of prestressed beams should be done to find out the factors that contribute to the flexural failure in the beams occurring due to strand rupture in the bottommost layer. Factors that should be studied in further experimental testing include the area of prestressing strands in the bottommost layer.

4.4.1.2.2. Observations at the Section of Point Load

At the section of the point load in the test, the strain compatibility analysis (CON-WHT-THR-R) predicted a moment capacity of 848 kip-ft (as shown in Table 4.12) that is less than the total maximum demand (896.3 kip-ft). As explained in Section 3.1.3.7, in this analysis, the effective prestress was determined by subtracting the estimated losses in prestressing steel from the initial prestress found from the original drawings. This was multiplied with the reduced area of cross-section of strands to arrive at the effective prestressing force in the prestressing steel. In Test-2, the first flexural crack was observed on the bottom flange of the beam at about 49.92 ft from the left support at an external load of 170.7 kip. Using this cracking load, an attempt was made to back-calculate the actual effective prestress in the strands. In this calculation, the tensile capacity of concrete was assumed to be $7.5\sqrt{f'c}$. This calculation was done in two ways: using the reduced cross-sectional area and centroid of the strands after accounting for the corrosion damage and using the full cross-sectional area and the original centroid of the strands. This is because the effect of corrosion damage on the effective prestress is not completely understood. Using the reduced cross-sectional area of strands (2.507 in^2) and the new centroid of strands (8.15 in) determined after accounting for the corrosion damage, the effective prestress, and effective prestressing force in the strands were estimated to be 169.5 ksi and 424.9 kip, respectively. In the same calculation, using a composite moment of inertia including the prestressing and non-prestressing steel for calculating the flexural stress at the crack location due to the applied external load, gave an effective prestress and effective prestressing force of 162.4 ksi and 407.1 kip, respectively. If the full area of cross-section and the original centroid of strands were used, the effective prestress and effective prestressing force were calculated to be 148.1 ksi and 371.2 kip, respectively. More research should be done to understand how the corrosion damage and the reduction in cross-sectional area of strands affect the effective prestress in the strands. Using an effective prestress and effective prestressing force in the strands calculated using the reduced strand properties and non-composite section properties as explained above, the ultimate flexural capacity of the cross-section where the point load was applied, calculated using strain compatibility analysis (CON-WHT-THR-R) was 875.7 kip-ft. This capacity is higher than the previously calculated value and also conservative when compared to the maximum demand that the section was able to carry without failure. From the test, it was found that the section did not fail at a moment demand of 896.3 kip-ft. The actual capacity of the beam at this section should have been higher than this, which was never reached during the test because the beam failed due to exceedance of the flexural capacity at a different section.

This section was at an embedment length of 4.58 ft from the right end of the beam which is less than the development length of the bottommost layer strands calculated using ACI 318 (2014) Equation 25.4.8.1 mentioned as Equation 3.1 in the current thesis, which was about 5.5 ft long. So, the strands at this location were not fully developed and had a stress less than that at the nominal flexural strength of the beam (f_{ps}). The stress in the strands at this section was calculated using a linear interpolation between f_{se} and f_{ps}. The accuracy of this calculation depends on the accuracy of the ACI or AASHTO equation for development length calculation. In the literature available, variability was observed in concluding if this equation gives a conservative or an over predicted estimate of the actual development length. For the strands in this beam, it is possible that the actual development length was less than that predicted using ACI or AASHTO equation providing a larger actual flexural capacity than the predicted value.

4.4.2. Comparison for Lesner Bridge I-Beam 9

The test setup, loading protocol, procedure, and the failure modes in these tests on Lesner Bridge I-Beam 9 are described in Section 3.1.4.2. The recommendations of Naito et al. (2011) were only proposed for residual strength calculations of box beams. In this research, these recommendations were also used to estimate the residual capacities of I-Beam sections to verify the validity of these recommendations for I-Beams.

4.4.2.1. Comparison at the Section of Failure in Test-1

In this test, failure occurred with concrete crushing in the deck at the section where the load was applied using an actuator, as discussed in Section 3.1.4.2.1.1. The maximum moment demand to cause the failure (at a distance of 8.83 ft from the left end) including the self-weight moment was 1880 kip-ft. The moment capacity of the beam calculated at the location of the failure, using strain compatibility analysis (CON-WHT-THR-R) described in Section 3.1.3.7.1 with a Whitney stress block model for compressive stress distribution in concrete, is shown in Table 4.16. The capacities at the slice-sections calculated using this analysis, following the recommendations of Naito et al. (2011) and the moment demand along the length of the beam corresponding to the maximum external load in Test-1 and the self-weight moment, are shown in Figure 4.35 to make comparisons at the section of failure.

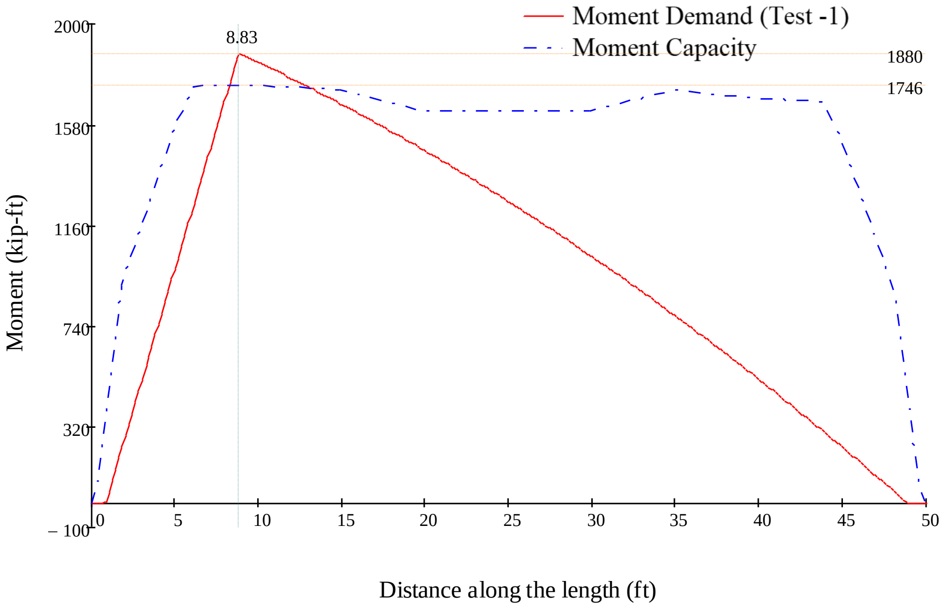


Figure 4.35. Moment Capacity (Strain Compatibility Analysis with CON-WHT-THR-R) vs. Demand Comparison at the section of failure for Test-1 on Lesner Bridge I-Beam-9

It can be observed from the comparison shown in Figure 4.35 that, the moment capacity calculation using CON-WHT-THR-R material parameters of the analysis was conservative in predicting the flexural capacity (1746 kip-ft) at the section of failure. Flexural Capacity at this section was also calculated using CON-HOG-THR-R parameters in the analysis as described in Section 3.1.3.7.4 with a Hognestad model based compressive stress in concrete, and using CON-HOG-EXPT-R parameters as described in Section 3.1.3.7.5 using an experimental stress-strain curve for prestressing steel (shown in Figure 3.14) in combination with Hognestad model for concrete. The estimated moment capacities at the failure section are shown in Table 4.16. The analysis was conservative in estimating the strength at the failure section. Hence, it was observed that the recommendations of Naito et al. (2011) are conservative when used to estimate the residual capacities of corrosion damaged I-Beams.

As explained in Section 2.8, as per the original recommendations of Naito et al. (2011) to account for the corrosion damage, for beams with longitudinal cracking, the prestressing strands which were not reduced to either 0% or 75% of their cross-sectional areas based on the considered corrosion damage, have to be reduced to 95% of their cross-sectional areas. As the calculations performed using these recommendations were found to be conservative, one of the recommendations was modified. Instead of reducing all the remaining strands (strands not reduced to either 0% or 75% of their areas) to 95% of the cross-sectional areas, they were retained at 100% of the cross-sectional areas in the strength calculations. This analysis is referred to in the results shown for the failure section in Table 4.16 as the 'analysis using modified Naito's recommendations'. In this table, undamaged capacities (capacity of the girder without accounting for any corrosion damage) of the cross-sections of the girders are also presented, so that a comparison can be made with the moment capacity of the section after accounting for the corrosion damage.

Furthermore, from the examination of the textured mesh, it was observed that the bottommost layer of the strands, at the location of spall in the mid-span region of the beam, had more effective cover than the effective cover of 2 in. shown on the design drawings. An attempt was made to measure this effective cover on vertices of the textured mesh by using the point picking tool in CloudCompare. From this, the actual effective cover to the bottommost layer of strands in the as-built condition of the girder was measured to be about 3 in. Measurement of this distance on vertices of the textured mesh in two different trials is shown in Figure 4.36. As there was only one spall on the beam with exposed strands, the textured mesh could not be used to verify this at other locations. After Test-1, at a section with a crack due to corrosion damage, concrete was chipped off to examine the location of the bottommost layer of strands. The strands in the four bottom layers were observed to be offset by about 1 in. upwards from their location mentioned in the design drawings. Here also, it was confirmed that the bottommost layer of strands had an effective cover of about 2.7 in. to 3 in. These measurements were made using another known dimension (depth of the bottom flange) as a baseline and is hence, not very accurate. It was confirmed from the measurements on the textured mesh that the depth of the bottom flange matched the bottom flange depth of 6 in. shown in the design cross-section drawing. This discrepancy in the effective cover between the design drawing and the as-built condition could be attributed to a quality control issue during construction. Therefore, strain compatibility analysis (CON-HOG-EXPT-R) was repeated for the failure location by offsetting the strands in the bottom four layers of the cross-section by an inch upwards. In this analysis, the strand offset was neither shown nor considered to determine the corrosion damage associated with the slice-sections (in Section 4.3.2.1). It was only accounted for in the strength calculations. The results of this analysis for the failure location are also included in Table 4.16.

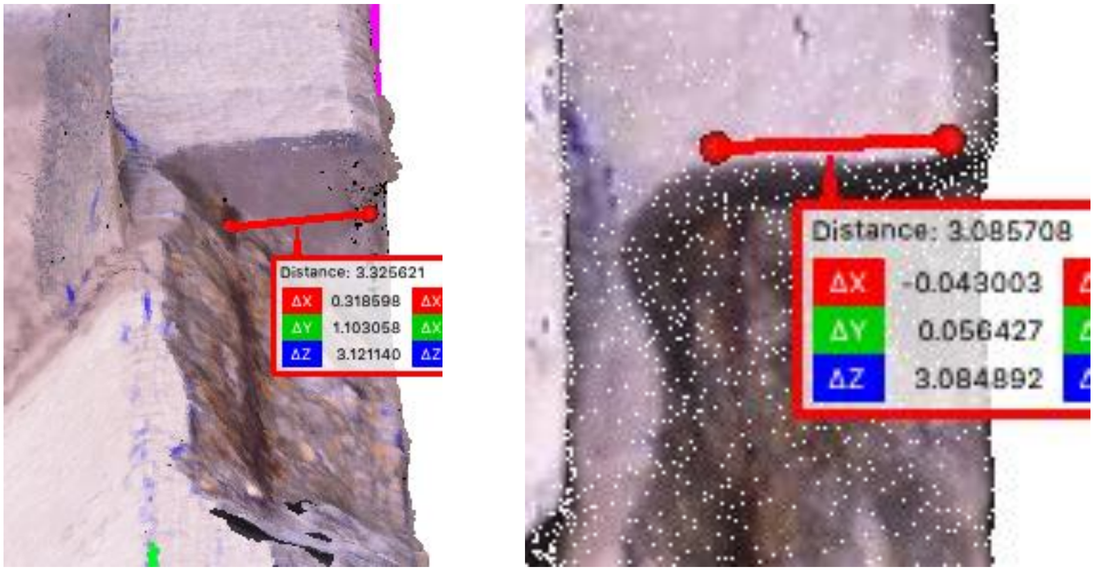


Figure 4.36. Measurement of Actual Effective Cover to the Strands in the Bottommost Layer of I-Beam 9 in as-built condition using CloudCompare

Table 4.16. Comparison of Moment Capacities at Failure Section in Test-1 on I-Beam 9 Calculated Using Strain Compatibility Analysis

Material Parameters in Strain Compatibility Analysis	**Positive Moment Capacity, kip-ft**				**Maximum Moment Demand at the Failure Location, kip-ft**
	CON-WHT-THR-R	**CON-HOG-THR-R**	**CON-HOG-EXPT-R**	**CON-HOG-EXPT-R + Offset of strands by 1 inch upwards**	
Using original recommendations by Naito et al. (2011)	1746	1774	1822	1780	1880
Using modified Naito's recommendations	1820	1849	1898	1854	
Undamaged Strength of the beam (ORG)	1865	1894	1942	1898	

The comparisons from Table 4.16 indicated that using original recommendations of Naito et al. (2011) in strain compatibility analysis gave less capacity than the total moment demand (1880 kip-ft) at the section of failure. By using the modified Naito's recommendations combined with CON-HOG-EXPT-R, the moment capacity calculated (1898 kip-ft) was close to the moment demand from Test-1. However, after accounting for the mismatch between the location of the bottom four strand layers in the as-built condition and the design cross-section drawing, the calculated moment capacity (1854 kip-ft) was still conservative when compared to the demand. This difference could be attributed to assumptions made in the analysis about the stress-strain relationship for the strands or the original recommendations of Naito et al. (2011) used in this analysis. Hence, by using modified recommendations of Naito et al. (2011) for calculation of residual capacity in I-Beams in combination with strain compatibility analysis with CON-HOG-EXPT-R, the moment capacity prediction was made less conservative than using the original recommendations with strain compatibility analysis (CON-WHT-THR-R). Furthermore, it was observed that the I-Beam 9 did not have as much longitudinal cracking before testing as that observed on the box beam-3. Hence, the modification made to the original Naito's recommendations seemed to be justifiable. This should be implemented on other corrosion damaged I-Beams to establish custom modified recommendations for Naito's approach that are capable of predicting the residual capacity of I-Beams without being too conservative. Moment capacity at the failure section without considering any corrosion damage after offsetting the strands (1898 kip-ft) is also close to the moment demand from the test.

4.4.2.2. Comparison at the Section of Failure in Test-2

In this test, a shear failure occurred starting at the section where the load was applied, as discussed in Section 3.1.4.2.2.1. The maximum total moment demand to cause the failure (at a distance of 42.92 ft from the left end) including the self-weight moment was 1910 kip-ft. The capacities at the slice-sections calculated as described in Section 3.1.3.7.1 using strain compatibility analysis with CON-WHT-THR-R, following the recommendations of Naito et al. (2011) and the moment demand along the length of the beam corresponding to the maximum external load in Test-2 and the self-weight moment, are shown in Figure 4.37. The moment capacities of the beam at the location of the point load, calculated using strain compatibility analysis with CON-WHT-THR-R, CON-HOG-THR-R, and CON-HOG-EXPT-R (as explained in Section 3.1.3.7) are summarized in Table 4.17.

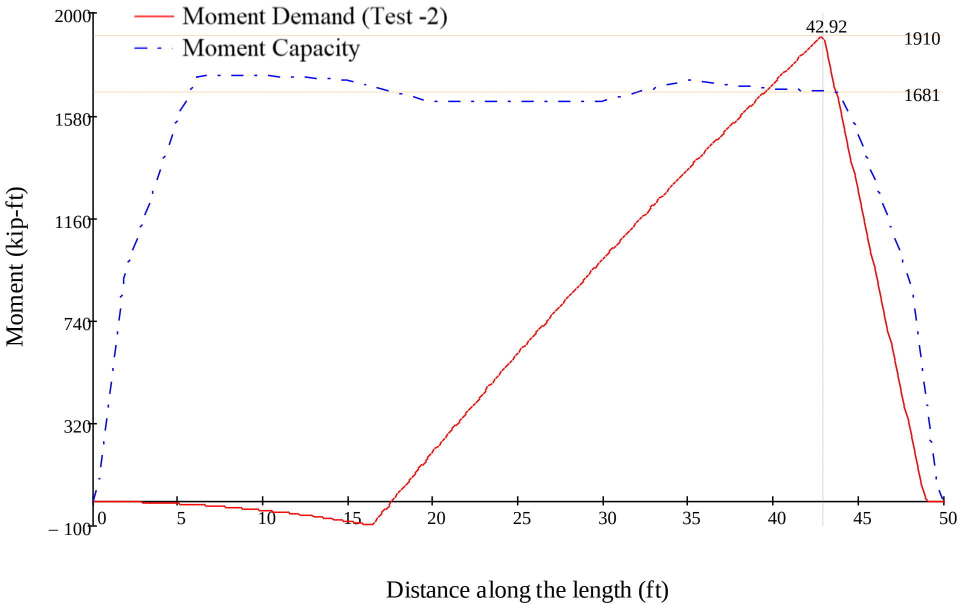


Figure 4.37. Moment Capacity (Strain Compatibility Analysis with CON-WHT-THR-R) vs. Demand Comparison at the section of failure for Test-2 on Lesner Bridge I-Beam-9

From the comparison in Figure 4.37, it was observed that the total moment demand that caused a shear failure (1910 kip-ft) was higher than the ultimate flexural capacity calculated using the original recommendations of Naito et al. (2011) with CON-WHT-THR-R (1681 kip-ft). It was already established that the original recommendations, when used for I-Beams, give a conservative prediction of ultimate moment capacity. In this case, the prediction was observed to be too conservative. By using the modified Naito’s recommendations combined with CON-HOG-EXPT-R, the ultimate moment capacity calculated (1802 kip-ft) was still less than the total moment demand that caused the shear failure. Hence, the damage considered for strength estimation at the section of failure could be conservative. Considering the actual location of the strand layers by offsetting the bottom four layers of strands by an inch reduced this ultimate moment capacity even further, to 1763 kip-ft.

Next, the undamaged moment capacity of the section was calculated. This capacity was calculated assuming that there was no corrosion damage at the section and in the two development lengths inspection window around the section. These results are also included in Table 4.17. On examination of these results, it was noticed that the undamaged moment capacity calculated using strain compatibility analysis with CON-HOG-EXPT-R including the offset in the locations of the strands in the bottom four layers, was 1898 kip-ft, which was still less than the actual shear capacity from the test. Possible reasons for this observation could be the pin support, which is at a distance of 6 ft from the section of the point load, supplying a horizontal restraint to the beam that is capable of creating an arching action rather than a truly simply supported condition. It is also possible that the application of the point load could have provided additional confinement in the compression zone of the beam, resulting in increased compressive strength of the concrete. Additional analysis on this issue was done using Strut and Tie modeling by Al Rufaydah (2020).

Table 4.17. Comparison of Moment Capacities at Failure Section in Test-2 on I-Beam 9 Calculated Using Strain Compatibility Analysis

Description of the Strain Compatibility Analysis	**Positive Moment Capacity (kip-ft)**				**Maximum Moment Demand at the Failure Location, kip-ft**
	CON-WHT-THR-R	**CON-HOG-THR-R**	**CON-HOG-EXPT-R**	**CON-HOG-EXPT-R + Offset of strands by 1 inch upwards**	
Using original recommendations by Naito et al. (2011)	1681	1700	1744	1707	1910
Using modified Naito's recommendations	1729	1757	1802	1763	
Undamaged Strength of the beam (ORG)	1865	1894	1942	1898	

CHAPTER 5 CONCLUSIONS AND FUTURE WORK

Corrosion damage is a major concern for bridges as many bridge failures in the past have been attributed to corrosion damage. The risk associated with corrosion damage caused failures increases as the infrastructure ages. Many bridges across the world built forty to fifty years ago are now in a deteriorated condition and need to be repaired and retrofitted. Bridge inspections are very important for condition assessment and determining the need for repair and retrofitting the corrosion damaged bridges. Close-in visual inspections, mapping the identified corrosion damage, and taking physical measurements of the damage are carried out along with Non-Destructive Evaluation (NDE) to assess the condition of bridges. This process is resource intensive and the safety of the personnel involved in the inspections is also a major concern. Physical access available for close-in inspections is limited, especially in the case of bridges built across water bodies. Reaching lower components of the superstructure of the bridges requires large machinery. Nowadays, a lot of research is being done in using Unmanned Aerial Vehicles (UAVs) for bridge inspections and in using artificial intelligence for the detection of cracks on the images of concrete and steel members. If the process of detecting the damage is automated, the next logical step that would add great value would be, to estimate the effect of the damage detected on the load carrying capacity of the bridges. Using a quantified estimate of the residual capacity of bridges, informed decisions can be made about a requirement for load posting or repair. This would forge a link between the automated visual inspection and residual capacity evaluation of girders. The current research is an exploratory attempt towards this goal.

The main objective of this research is to help resource-intensive bridge inspections and reduce human involvement by semi-automating the processes of damage detection and damage mapping and use this damage to assess the capacity of the bridge girders. The stepwise framework was implemented on actual full-scale prestressed concrete bridge girders obtained from decommissioned bridges in Virginia. At multiple locations along the lengths of the girders, the prestressing strands affected due to the corrosion damage were identified using a semi-automated process. The affected prestressing strands were reduced in cross-sectional area to account for the corrosion damage as per the recommendations of Naito et al. (2011) and were used to calculate the flexural capacity of the bridge girders. The estimated flexural capacities of the girders were compared to the actual capacities of the girders found using destructive tests conducted by Al Rufaydah (2020).

Current practices that require physical access to the girders rely on the images and physical measurements captured during close-in inspections, to assess the effect of the corrosion damage. For this purpose, 3D point clouds offer a promising way of visually representing the girders and the corrosion damage in three dimensions. On 2D images showing only a portion of the girder, it can be difficult to understand the depth dimension of the damage and the importance of this damage in terms of its scale and location on the full girder. 3D point clouds not only facilitate this but also allow measurement of any distance or dimension which can be scaled to real world dimensions. The most important advantage of using 3D point clouds of the bridges as proposed in this research is that close physical access is not required to generate them. LiDAR and Photogrammetry are two possible ways of building the point clouds. LiDAR involves high equipment costs, whereas Photogrammetry facilitates building 3D point clouds using images captured using cameras which are cheaper and have now become commonplace. UAVs can be used to capture the images required to build the 3D point clouds. Instead of entirely relying on UAVs for this purpose, a combination of ground (terrestrial) robots and UAVs can also be used to reduce the time taken to stabilize the UAVs and reduce the problems caused by wind disturbances in the process of image capture. 3D point clouds of the actual prestressed concrete bridge girders mentioned in Table 3.1, were reconstructed from 2D images using Structure from Motion (SfM).

Cracks were detected on the 2D images used for point cloud reconstruction using a Fully Convoluted Network (FCN) with a U-Net architecture. These crack detected images were used for texture mapping the surface meshes on the point clouds of the girders. This facilitated transfer of cracks on to the 3D point clouds. Points belonging to the cracked regions were isolated into a new point cloud. Spalling damage on the point cloud was detected using a thresholding approach on the orientation of the point normals with respect to the reference directions. Dense point clouds and textured mesh models of the girders were scaled to real-world dimensions. Measurement of required dimensions was done on the dense point clouds, and the textured point cloud was super imposed with the cleaned crack and spall point clouds (called a digital state model in this research). Any cracks or spalls that went unidentified in the damage detection were visible on the textured meshes of the girders adding to the accuracy of the approach. Other types of damage such as severed or exposed strands and rust staining on the concrete surface were identified manually by observing the textured mesh models of the girders.

Slices were extracted from the dense point clouds at ten slice-sections along the length of each girder. As per the recommendations of Naito et al. (2011), prestressing strands at a section are affected due to the corrosion damage present in a distance of one development length of the strands, on each side of the section. Mostly, the strands in two bottom layers were affected due to corrosion damage. Therefore, while extracting the slices from the dense point clouds, damage detected in two development regions of the bottommost layer strands (referred to as Inspection Window in this thesis), around the slice-section was projected on to the slices. The slices were overlaid on the cross-section drawings, and the strands affected due to the corrosion damage were identified. They were reduced in cross-sectional area and were used in the calculation of the ultimate flexural capacity of the girders using strain compatibility analysis. In this research, strain compatibility approach was used with different sets of material parameters with prescribed ultimate strains in either concrete or prestressing steel (as mentioned in Table 3.2) to simulate the failure mode. Different constitutive models were used to calculate the residual capacity at the slice-sections. Estimated residual capacities were compared to the actual capacities of the girders found from tests conducted by Al Rufaydah (2020). Comparisons are presented for the failure sections in these tests and the results were analyzed to evaluate the effectiveness of this method. The limitations of the proposed framework are as follows.

- Accuracy of the 2D image-level damage detection affected the quality of the 3D damage maps generated. Bad performance of the Neural Network caused cracks to be excluded from the detection either partially or fully. This problem was partially overcome by using the textured mesh models of the girders, on which undetected cracks were still visible for 3D damage maps instead of using the dense point clouds.
- As the Convolutional Neural Network for crack detection (CNN) was not trained on many images from the current dataset, the detection still has false positives in the detection which were cleaned partly manually on the dense point cloud.
- Cracks that are narrower than the distance between two points in a dense point cloud of a girder could be only seen on the textured mesh model.

- Time taken to implement the framework to estimate the residual capacity of one girder was found to be about 45 hours which is on the high side.

Major conclusions reached from the current research are listed below.

5.1. Conclusions

- The proposed framework facilitates estimating residual strength of a bridge girder without the need for physical access or physical measurement of the dimensions of the damage identified. The point clouds and the textured mesh models can be scaled to the real-world dimensions making use of a known dimension of any bridge component. After scaling, any dimension of the girders can be measured on the 3D dense point clouds using a point cloud processing tool.
- 3D point clouds and textured mesh models were found to be a great visual medium to convey the information about corrosion damage present on the girders. As opposed to representing the damage on 2D images, this approach also conveyed depth perspective, scale, and location of the damage with respect to the full length of the girder.
- The need for human, monetary, and machinery resources required for close-in bridge inspections can be greatly reduced if the framework proposed in this research is implemented.
- Fully Convolutional Network (FCN) with a U-Net architecture was shown to be promising and robust in detecting cracks on the images of the concrete girders that the network was not trained on, despite the presence of varying surroundings and lighting conditions. There were some false positives in the output of 2D level detection which was later cleaned after transferring the cracks to the 3D point clouds. This step requires further improvement with training the network on a variety of images from the current dataset.
- Thresholding on the orientation of the point normals with respect to the reference directions approach was shown to be effective in identifying spalled regions on the point clouds of the girders.
- An effort was made to use the same values for manually defined parameters in each implementation of the algorithms used in this research for each girder. In future work, the values used for manual parameters in the current research can be used as starting points.

Optimal parameters for the density of points in the future datasets can be determined by separately fine-tuning parameters involved in the algorithms used in this research.

- Using design drawings of the girders to be overlaid on top of the slices extracted from the point cloud was not very accurate since the actual locations of the strands varied slightly with the locations from the design drawings. The variations of the as-built conditions from the design drawings could be identified on the textured mesh models of the girders.
- Using textured mesh models of the girders superimposed with the damage clusters is recommended for identifying the strands affected due to corrosion damage because the cracks that the neural network failed to detect were still visible on the texture and can be considered for strength estimation. Hence, it is important to capture well-lit, sharp, and non-blurry images of the beams, especially on the bottom surfaces.
- Comparison of the ultimate flexural capacity of Box Beam-3 estimated using strain compatibility analysis with CON-WHT-THR-R material parameters with the actual shear capacity in Test-1, indicated that Naito's recommendations provided a conservative estimate of the flexural capacity.
- In Test-2, the failure occurred due to the rupture of strands in the bottommost layer at the location of spalling in the middle of the bottom surface with five exposed strands that can be seen in Figure 4.2a. As strain compatibility analysis with CON-WHT-THR-R did not capture this failure mode, STL-HOG-THR-R and STL-HOG-THR-ORG material parameters were used to find the ultimate strain in the bottommost layer strands at the time of fracture. Strain compatibility analysis with STL-HOG-THR-R was found to give a better but not accurate prediction of the ultimate rupture strain as 1.305%. These exposed strands shown in Figure 4.33 were in a heavily corroded 'wire loss' state of corrosion.
- Hence, it is important to find the estimates of average ultimate strains that cause rupture in prestressing strands with different levels of corrosion, using tensile tests on strand samples extracted from corrosion damaged bridge girders.
- Using these estimates of the ultimate strain in the bottommost layer of strands, if the corrosion damage in a bridge beam is found to be similar to that observed at the failure section of Box Beam-3 (shown in Figure 4.33), strain compatibility analysis with STL-HOG-THR-R can give a better approximation of the ultimate flexural capacity.

- Actual bridges that have girders with spalling and exposed strands in a heavily corroded state are likely to fail with strand rupture if loaded to ultimate strains in the range of the corresponding estimates. This would require load posting to avoid the ultimate strains in this range and has to be explored using finite element modeling.
- Recommendations of Naito et al. (2011) were proposed only for box beams. Using these recommendations for I-Beam 9 was found to give conservative estimates of residual strength as can be observed from the comparisons with the actual capacities from Tests 1 and 2 on I-beam 9.
- Using strain compatibility analysis with CON-HOG-EXPT-R parameters with modified Naito's recommendations for I-Beam 9 (strands taken neither to 0% or 75%, were retained at 100% of the cross-sectional area) considering the offset of the strands in the four bottom layers by 1 inch upwards was found to give a closer but still a little conservative estimate of the actual flexural capacity in Test-1. In Test-2, these material parameters were found to give an estimate of the ultimate flexural capacity which was more conservative than the actual shear strength found from the test.
- This framework was found to give good estimates of residual strength. Reduction in resources involved in current visual inspection practices and eliminating the need for physical access, make this approach worthwhile to be explored further to improve the output of each step in the proposed framework which would also result in cost-saving.
- Manual image capture procedure followed in the current research can be done using a UAV or a combination of a ground robot and a UAV to reduce problems associated with UAVs that were introduced in Section 2.2.
- The time required to implement the proposed framework to estimate the residual capacity of a single bridge girder was roughly 45 hours which is slightly on the high side. Using a UAV for image capture would add a significant amount of time to this. However, when compared to the time required to mobilize the personnel, machinery and other resources and the required for traffic closures in the current practices for visual inspection of bridges, the time taken for implementing the current framework would be worthwhile as the need for most of these resources would be eliminated.
- Considerable research is required to implement this framework on an actual full-scale prestressed concrete bridge to explore the usage of UAVs for image capture step, optimize

the processes involved, and reduce the total time required. Use of UAVs for capturing image and videos for bridge inspections are introduced in Section 2.2. If the same images or videos used for manually detecting the damage, were to be used for residual strength estimation, the process would become mostly manual. If such a process with closely captured videos from multiple sides of each girder was used, the affected prestressing strands would have to be identified manually and reduced in cross-sectional area. Identifying exposed strands due to spalling using this more manual approach would be easier than locating the strands affected due to cracks. Furthermore, this would require knowledge of the spatial location of the UAV and the location of the damage with respect to the length of the bridge. Straight line distances between sections can be estimated by knowing the georeferenced GPS locations of the UAV. As mentioned in Section 2.2, it is problematic to obtain GPS locations under a bridge. Even if the location of the damage from the nearest surface is known, it is not easy to understand the spatial relationship of the damage identified in multiple videos of the same location captured from different angles. Identifying all the damage that falls in the inspection window of length equal to two development length (of the prestressing strands) regions around a section can also be difficult.

- Compared to the more manual approach mentioned above, the framework proposed in this research offers some advantages. It allows reconstruction of 3D textured mesh models, scaled to real-world dimensions, which contain rich information about the location of the damage. The damage can be easily located with respect to any required edge or end of the girder or the bridge. Crack lengths and crack widths can be measured. Measurement of any required dimension also adds value by giving information about the as-built condition of the bridge, like in the case of I-Beam 9 where the strands were actually present about one inch higher than the locations indicated on the cross-section drawings. Aspects like this can affect the calculated residual capacity. Future work on the framework proposed in this research can improve the approaches involved and can reduce the time taken for the whole process, making it easier to implement.
- Recommendations of Naito et al. (2011) do not consider lengths and widths of the longitudinal cracks to determine a reduced cross-sectional area of the strands located close to the cracks. Further research has to be done to determine if the degree of corrosion on the

strands is affected by the dimensions of the longitudinal cracks lying directly in line with or close to the strands. If future work shows that the crack dimensions affect the strand corrosion, a tool that enables measurement of the dimensions of the cracks like in the current research would add great value.

5.2. Future Work

Most of these recommendations are intended to improve the result of any step in the proposed framework.

5.2.1. Recommendations for Digital State Models and Semi-Automated Damage Maps

- More work has to be done to determine appropriate overlap between consecutive images and images from consecutive locations to obtain a desirable number of points in the dense point clouds of the girders. Using fewer images with the required overlap (60% to 80%) along with a Medium or High quality parameter on Agisoft Metashape for dense cloud reconstruction can yield larger number of points.
- Work has to be done to explore using UAVs and ground robots for image capture in the framework. Appropriate overlap between images, stabilizing the UAV at image capture location, determining appropriate lighting for dimly lit portions of the bridges, and optimizing and minimizing the time required are some of the aspects to be studied.
- The use of frames from a video captured around the girder in the specified procedure for reconstruction of the point cloud from Structure from Motion can be explored.
- Infrared maps of the girders captured using an Infrared camera can be used for texture mapping on the surface meshes on point clouds to identify locations of delamination.
- The Neural Network used in this research can be trained with more images from the current dataset to improve its performance. The performance should be quantified using labeled images from the current dataset that are not used to train the model. Performances measures like accuracy, precision, recall, Intersection over Union (IoU), etc. should be determined to arrive at the optimal parameters in training and testing processes.
- Instead of using a point-normal based approach for detecting spalls on the point cloud, the same Neural Network (U-Net) that is currently used to detect cracks can be trained to detect multiple types of damage such as spalls with and without exposed strands.

- On the exposed strands, different levels of corrosion could also be detected using a Neural Network.

5.2.2. Recommendations for Residual Strength Estimation

- The process of overlaying the slice image with an image of the cross-section drawing can be automated using template matching or edge matching techniques.
- More research should be done on the factors that contribute to the flexural failure in the beams occurring due to multiple strand rupture. Ultimate strain in the bottommost layer strands and reduced area of the strands after accounting for corrosion damage can be possible factors whose effect should be studied.
- More tension tests on strand samples extracted from corrosion damaged girders are to be conducted to associate each level of strand corrosion with an average value of ultimate strain in the bottommost layer strands.
- More research should be done to understand how the corrosion damage and the reduction in the cross-sectional area of strands affect the effective prestress in the strands. More literature review should be done to determine whether using full cross-sectional properties or reduced cross-sectional properties of strands would give a better estimate of the actual effective prestress in the prestressing steel back-calculated from the cracking load in a flexural test on corrosion damaged prestressed concrete beam.
- More analysis is to be done to assess the redevelopment length of strands near a damaged region. Finite element analysis on the model of Box Beam-3 can be used to determine which locations of the point load near the spalling will cause the beam to fail in strand rupture at the spall location.
- Further research has to be done to determine if the degree of corrosion on the strands is affected by the dimensions of the longitudinal cracks lying directly in line with or close to the strands. Also, degree of corrosion on the unexposed strands located close to other types of corrosion damage like rust staining on the concrete surface can also be studied to determine if reduction in cross-sectional area is required to get better estimates of residual strength.

REFERENCES

Agisoft Helpdesk. (2019). General Image Capture Tips for Agisoft Metashape. Retrieved from https://agisoft.freshdesk.com/support/solutions/articles/31000149337-general-tips

Agisoft Metashape Professional (Version 1.6.2) (Software). (2020). Retrieved from https://www.agisoft.com/downloads/installer/

Al Rufaydah, A. (2020). *Shear Strength Assessment of Corrosion Damaged Prestressed Concrete Bridge Girders*. Virginia Polytechnic Institute and State University.

Alfailakawi, A. (2019). *Experimental and Analytical Evaluation of Residual Capacity and Repair Methods for Corrosion-Damaged Prestressed Concrete Bridge Girders*.

American Association of State Highway and Transportation Officials (AASHTO). (2017). *AASHTO LRFD Bridge Design Specifications*. Washington DC.

American Concrete Institute (ACI). (2014). *Building Code Requirement for Structural Concrete (ACI 318-14)*. Farmington Hills, MI.

ASCE. (2017). *Report Card on America's Infrastructure*. Retrieved from https://www.infrastructurereportcard.org/wp-content/uploads/2017/01/Bridges-Final.pdf

Bang, S., Park, S., Kim, H., & Kim, H. (2019). Encoder–decoder network for pixel-level road crack detection in black-box images. *Computer-Aided Civil and Infrastructure Engineering*, *34*(8), 713–727. https://doi.org/10.1111/mice.12440

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Cha, Y. J., Choi, W., Suh, G., Mahmoudkhani, S., & Büyüköztürk, O. (2018). Autonomous Structural Visual Inspection Using Region-Based Deep Learning for Detecting Multiple Damage Types. *Computer-Aided Civil and Infrastructure Engineering*, *33*(9), 731–747. https://doi.org/10.1111/mice.12334

Chang, P. C., Flatau, A., & Liu, S. C. (2003). *Review Paper : Health Monitoring of Civil Infrastructure*. *2*(3), 257–267.

Class Notes for CS231n: Convolutional Neural Networks for Visual Recognition. (n.d.). Retrieved from http://cs231n.github.io/convolutional-networks/

Dorafshan, S., & Maguire, M. (2018). Bridge Inspection: Human Performance, Unmanned Aerial Systems and Automation. In *Journal of Civil Structural Health Monitoring* (Vol. 8). https://doi.org/10.1007/s13349-018-0285-4

Dorafshan, S., Maguire, M., Hoffer, N. V, & Coopmans, C. (2017). Fatigue Crack Detection Using Unmanned Aerial Systems in Under-Bridge Inspection. In *Research Report: Idaho*

Transportation Department. https://doi.org/10.1061/(ASCE)BE.1943-5592.0001291.

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

Dorafshan, S., Thomas, R. J., & Maguire, M. (2018b). Fatigue Crack Detection Using Unmanned Aerial Systems in Fracture Critical Inspection of Steel Bridges. *Journal of Bridge Engineering*, *23*(10), 1–15. https://doi.org/10.1061/(ASCE)BE.1943-5592.0001291

Dung, C. V., & Anh, L. D. (2019). Autonomous concrete crack detection using deep fully convolutional neural network. *Automation in Construction*, *99*(October 2018), 52–58. https://doi.org/10.1016/j.autcon.2018.11.028

Eisenbach, M., Stricker, R., Seichter, D., Amende, K., Debes, K., Sesselmann, M., … Gross, H.-M. (2017). How to Get Pavement Distress Detection Ready for Deep Learning? A Systematic Approach. *International Joint Conference on Neural Networks (IJCNN)*, 2039–2047.

Erkal, B. G., & Hajjar, J. F. (2014). *Laser-Based Structural Sensing and Surface Damage Detection*. Boston, Massachusetts.

Erkal, B. G., & Hajjar, J. F. (2017). Laser-based surface damage detection and quantification using predicted surface properties. *Automation in Construction*, *83*(August), 285–302. https://doi.org/10.1016/j.autcon.2017.08.004

FHWA. (2016). *Deficient Bridges by Superstructure Material 2016*. Retrieved from https://www.fhwa.dot.gov/bridge/nbi/no10/mat16.cfm#b

FHWA. (2018). *Bridge Condition by Highway System 2018*. Retrieved from https://www.fhwa.dot.gov/bridge/nbi/no10/condition18.cfm

Helsel, M., & Salomon, A. L. (2015). 3D + Time Reconstruction: Designing Optimal Camera Parameters. In *Virginia Polytechnic Institute and State University*.

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science Engineering*, *9*(3), 90–95.

I-Art 3D. (2019). What is “Depth Map”? Retrieved from http://www.i-art3d.com/Eng/About_Depth.htm

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

Kalfarisi, R., Wu, Z. Y., & Soh, K. (2020). Crack Detection and Segmentation Using Deep Learning with 3D Reality Mesh Model for Quantitative Assessment and Integrated

Visualization. *Journal of Computing in Civil Engineering*, *34*(3), 4020010. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000890

Leal da Silva, W. R., & Lucena, D. S. De. (2018). *Concrete Cracks Detection Based on Deep Learning Image Classification †*. 1–6. https://doi.org/10.3390/ICEM18

Liu, Y., Yao, J., Lu, X., Xie, R., & Li, L. (2019). DeepCrack: A Deep Hierarchical Feature Learning Architecture for Crack Segmentation. *Neurocomputing*, *338*, 139–153. https://doi.org/10.1016/j.neucom.2019.01.036

Liu, Z., Cao, Y., Wang, Y., & Wang, W. (2019). Computer vision-based concrete crack detection using U-net fully convolutional networks. *Automation in Construction*, *104*(January), 129–139. https://doi.org/10.1016/j.autcon.2019.04.005

Maguire, M., Dorafshan, S., & Thomas, R. J. (2018). SDNET2018: A concrete crack image dataset for machine learning applications. https://doi.org/https://doi.org/10.15142/T3TD19

Naito, C., Jones, L., & Hodgson, I. (2011). Development of flexural strength rating procedures for adjacent prestressed concrete box girder bridges. *Journal of Bridge Engineering*, *16*(5), 662–670. https://doi.org/10.1061/(ASCE)BE.1943-5592.0000186

Naito, C., Sause, R., Hodgson, I., Pessiki, S., & Desai, C. (2006). *Forensic Evaluation of Prestressed Box Beams from the Lake View Drive over I- Forensic Evaluation of Prestressed Box Beams from the Lake View Drive over I-70 Bridge*. Bethlehem, PA.

Oliphant, T. E. (2006). *A guide to NumPy* (Vol. 1). Trelgol Publishing USA.

Open3D Point Cloud Tutorial. (2020). Voxel Downsampling. Retrieved from www.open3d.org website: http://www.open3d.org/docs/release/tutorial/Basic/pointcloud.html#Voxel-downsampling

Orsa, R. J., Eatherton, M. R., & Moen, C. D. (2011). *Using Close-Range Photogrammetry to Characterize Initial Imperfections of Cold Formed Steel Members*.

Özgenel, Ç. F. (2019). Concrete Crack Images for Classification. https://doi.org/http://dx.doi.org/10.17632/5y9wdsg2zt.2#file-c0d86f9f-852e-4d00-bf45-9a0e24e3b932

Özgenel, F., & Gönenç Sorguç, A. (2018). Performance comparison of pretrained convolutional neural networks on crack detection in buildings. *ISARC 2018 - 35th International Symposium on Automation and Robotics in Construction and International AEC/FM Hackathon: The Future of Building Things*, (Isarc). https://doi.org/10.22260/isarc2018/0094

Pauly, M., Gross, M., & Kobbelt, L. P. (2002). Efficient simplification of point-sampled surfaces. *Proceedings of the IEEE Visualization Conference*, (Section 4), 163–170. https://doi.org/10.1109/visual.2002.1183771

Python Software Foundation. (2020). bisect — Array bisection algorithm. Retrieved from Python

Software Foundation website: https://docs.python.org/3/library/bisect.html

Rabbani, T., van den Heuvel, F. A., & Vosselman, G. (2006). Segmentation of point clouds using smoothness constraint. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences - Commission V Symposium "Image Engineering and Vision Metrology," 36*(5), 248–253. Retrieved from http://www.isprs.org/proceedings/XXXVI/part5/paper/RABB_639.pdf

Richard, A., & Gall, J. (2017). *A Bag-of-Words Equivalent Recurrent Neural Network for Action Recognition*.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 9351*, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

Rusu, R. B., Marton, Z. C., Blodow, N., Dolha, M., & Beetz, M. (2008). Towards 3D Point cloud based object maps for household environments. *Robotics and Autonomous Systems, 56*(11), 927–941. https://doi.org/10.1016/j.robot.2008.08.005

Salomon, A. L. (2016a). *Digital State Models for Infrastructure Condition Assessment and Structural Testing*. Virginia Tech.

Salomon, A. L. (2016b). *OJOS Noncontact Visual Sensing Instruction Manual*.

Salomon, A. L., Straub, B., Kriz, A., Stilwel, D. J., Hebdon, M. H., & Moen, C. D. (2017). Bridge Geometry Measurements with an Unmanned Aerial Vehicle Flown in Local Wind Disturbances. *Submitted to Transportation Research Board 96th Annual Meeting Compendium of Papers*, 10. https://doi.org/10.5151/cidi2017-060

scikit-learn developers. (2019). Clustering using scikit-learn. Retrieved February 3, 2020, from https://scikit-learn.org/stable/modules/clustering.html#dbscan

Shakarji, C. M. (1998). Least-Squares Fitting Algorithms of the NIST Algorithm Testing System. *Journal of Research of the National Institute of Standards and Technology, 103*(6), 633–641. https://doi.org/10.6028/jres.103.043

Shelhamer, E., Long, J., & Darrell, T. (2017). Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(4), 640–651.

Shi, Y., Cui, L., Qi, Z., Meng, F., & Chen, Z. (2016). Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems, 17*(12), 3434–3445.

Smith, A. G. (2020). *As-Built Structural Models from Laser Scan Data*. Virginia Polytechnic Institute and State University.

Taha, A. A., & Hanbury, A. (2015). Metrics for evaluating 3D medical image segmentation : analysis , selection , and tool. *BMC Medical Imaging*. https://doi.org/10.1186/s12880-015-0068-x

The GIMP Development Team. (n.d.). GNU IMAGE MANIPULATION PROGRAM (GIMP) Version 2.10.14. Retrieved from https://www.gimp.org

Torok, M. M., Fard, M. G., & Kochersberger, K. B. (2012). Post-disaster Robotic Building Assessment: Automated 3D Crack Detection from Image-based Reconstructions. *Computing in Civil Engineering (2012)*.

Torok, M. M., Golparvar-Fard, M., & Kochersberger, K. B. (2014). Image-based automated 3D crack detection for post-disaster building assessment. *Journal of Computing in Civil Engineering*, *28*(5), 1–13. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000334

van der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science Engineering*, *13*(2), 22–30. https://doi.org/10.1109/MCSE.2011.37

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., … Contributors, S. 1. 0. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/https://doi.org/10.1038/s41592-019-0686-2

Wallisch, P. (2014). Principal Components Analysis. In *MATLAB for Neuroscientists* (Second, pp. 305–315). https://doi.org/10.1016/B978-0-12-383836-0.00017-5

Yang, F., Zhang, L., Yu, S., Prokhorov, D., Mei, X., & Ling, H. (2019). Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection. *ArXiv Preprint ArXiv:1901.06340*.

Yang, L., Li, B., Li, W., Liu, Z., Yang, G., & Xiao, J. (2017). Deep Concrete Inspection Using Unmanned Aerial Vehicle Towards CSSC Database. *International Conference on Intelligent Robots and Systems (IROS)*, (61528303).

Yang, X., Li, H., Yu, Y., Luo, X., Huang, T., & Yang, X. (2018). Automatic Pixel-Level Crack Detection and Measurement Using Fully Convolutional Network. *Computer-Aided Civil and Infrastructure Engineering*, *33*, 1090–1109. https://doi.org/10.1111/mice.12412

Yeum, C. M., & Dyke, S. J. (2015). Vision-Based Automated Crack Detection for Bridge Inspection. *Computer-Aided Civil and Infrastructure Engineering*, *30*, 759–770. https://doi.org/10.1111/mice.12141

Zhang, A., Wang, K. C. P., Fei, Y., Liu, Y., Chen, C., Yang, G., … Qiu, S. (2019). Automated Pixel-Level Pavement Crack Detection on 3D Asphalt Surfaces with a Recurrent Neural Network. *Computer-Aided Civil and Infrastructure Engineering*, *34*(3), 213–229. https://doi.org/10.1111/mice.12409

Zhang, L., Yang, F., Zhang, Y. D., & Zhu, Y. J. (2016). Road crack detection using deep convolutional neural network. *2016 IEEE International Conference on Image Processing (ICIP)*, 3708–3712.

Zhou, Q.-Y., Park, J., & Koltun, V. (2018). {Open3D}: {A} Modern Library for {3D} Data Processing. *ArXiv:1801.09847*.

Zou, Q., Cao, Y., Li, Q., Mao, Q., & Wang, S. (2012). CrackTree: Automatic crack detection from pavement images. *Pattern Recognition Letters*, *33*(3), 227–238.

APPENDICES

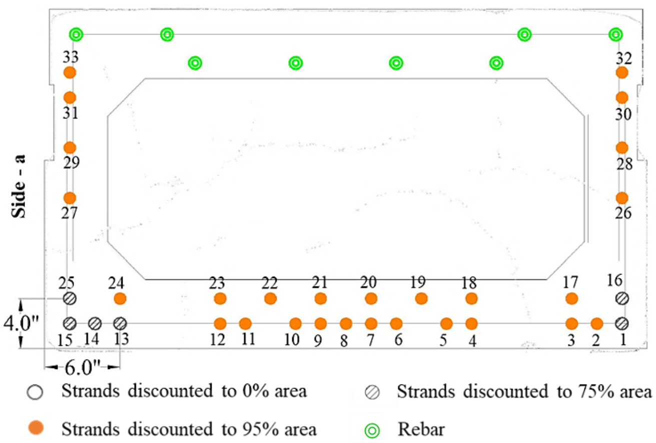
APPENDIX - A

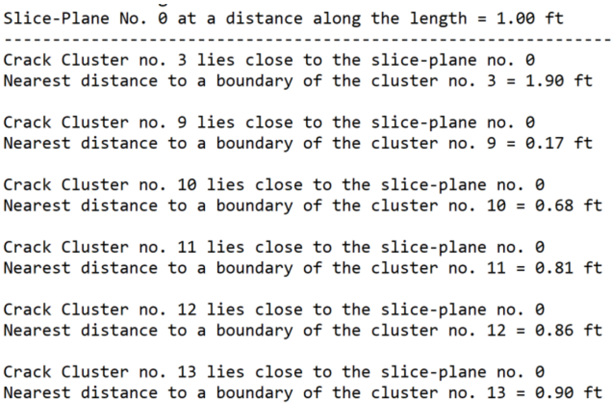
ESTIMATION OF RESIDUAL CAPACITY OF BOX BEAM-3 AT VARIOUS SLICE-SECTIONS

This Appendix deals with the determination of corrosion damage inside the Inspection Window for each slice-section of Box Beam-3 to arrive at a reduced area of cross-section to be used for strength calculation. Residual strength is calculated using the reduced cross-sectional area of the strands as explained in Section 3.1.3. As explained in Section 3.1.3.2, slices were extracted at 22 slice-sections along the length of the beam. The first and the last slices were extracted at a distance of 1 ft from the corresponding ends of the beam. The spacing of intermediate slices was chosen to be 2.5 ft. The spacing of the extracted slices is demonstrated in Figure 3.9. In the following sub-sections, the overlay of each slice on top of the design cross-section drawing is shown along with automatically generated the text file output that shows the list of damage clusters to be considered to calculate strength at each slice-section. These constitute the tools for strength estimation at the slice-section. These damage clusters are shown superimposed on a portion of the textured mesh of the beam which lies inside the Inspection Window (IW) of size equal to two development lengths of the bottommost layer strands. Strands affected due to each damage cluster are illustrated for each slice-section. At some slice-sections, there were some undetected cracks, especially on the bottom surface which affect the strength of the beam. These cracks were still visible on the textured mesh and were accounted for in strength calculation. These undetected cracks, if any, are highlighted using bounding boxes on the textured mesh shown for each slice-section.

The distance along the length of the beam at which each slice-section was extracted is shown and the tools for strength estimation as explained above are presented in the following sub-sections of the Appendix. Inspection Window for each slice-section is represented as IW. Crack clusters are designated as ‘C’, followed by the identification number of the cracks (for example, C9 means Crack Cluster No. 9). Similarly, spall clusters are represented using the letter ‘S’, followed by a number. Side-a and side-b are marked to show the sides of the beam as represented in Figure 4.2a.

Slice No. 0 (At y = 1 ft from the left end of the beam)


Side - a
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
4.0"
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
6.0"
○ Strands discounted to 0% area
▨ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar



```
Slice-Plane No. 0 at a distance along the length = 1.00 ft
--------------------------------------------------------------
Crack Cluster no. 3 lies close to the slice-plane no. 0
Nearest distance to a boundary of the cluster no. 3 = 1.90 ft

Crack Cluster no. 9 lies close to the slice-plane no. 0
Nearest distance to a boundary of the cluster no. 9 = 0.17 ft

Crack Cluster no. 10 lies close to the slice-plane no. 0
Nearest distance to a boundary of the cluster no. 10 = 0.68 ft

Crack Cluster no. 11 lies close to the slice-plane no. 0
Nearest distance to a boundary of the cluster no. 11 = 0.81 ft

Crack Cluster no. 12 lies close to the slice-plane no. 0
Nearest distance to a boundary of the cluster no. 12 = 0.86 ft

Crack Cluster no. 13 lies close to the slice-plane no. 0
Nearest distance to a boundary of the cluster no. 13 = 0.90 ft
```

a) Overlay of Slice No. 0 with the Cross-Section Drawing

b) Portion of the Text File Showing the Damage Clusters for Slice No. 0

Figure A.1. Tools for Strength Estimation at Slice-0 on Aden Road Bridge Box Beam-3

Table A.1. Effect of the Damage Clusters on the Prestressing Strands Inside the IW

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C3, Undetected Crack (visible on texture)	Side - a	13,14, 15, 25	Unexposed	75%
C9, C10	Side - b	1, 16	Unexposed	75%
C11, C12, C13	End of the beam	Cracks at the end assumed not affecting strength	Unexposed	-

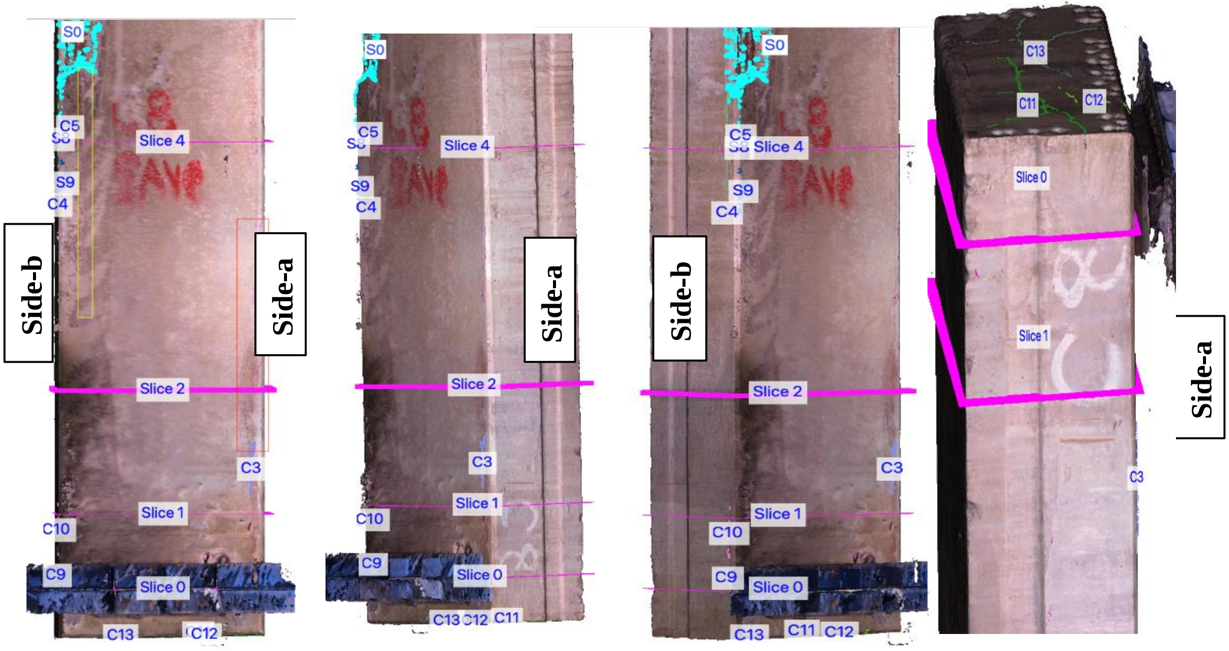
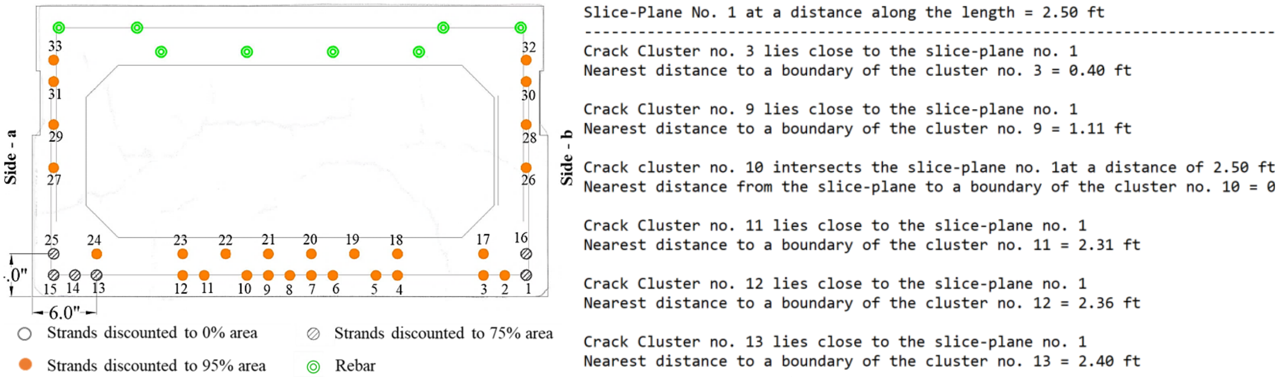


Figure A.2. Damage Clusters in the IW considered for Strength Estimation at Slice-0

Slice No. 1 (At y = 2.5 ft from the left end of the beam)


Side - a
Side - b
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
.0"
6.0"
○ Strands discounted to 0% area
⊘ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

a) Overlay of Slice No.1 with the Cross-Section Drawing

```
Slice-Plane No. 1 at a distance along the length = 2.50 ft
---------------------------------------------------------------------------
Crack Cluster no. 3 lies close to the slice-plane no. 1
Nearest distance to a boundary of the cluster no. 3 = 0.40 ft

Crack Cluster no. 9 lies close to the slice-plane no. 1
Nearest distance to a boundary of the cluster no. 9 = 1.11 ft

Crack cluster no. 10 intersects the slice-plane no. 1at a distance of 2.50 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 10 = 0

Crack Cluster no. 11 lies close to the slice-plane no. 1
Nearest distance to a boundary of the cluster no. 11 = 2.31 ft

Crack Cluster no. 12 lies close to the slice-plane no. 1
Nearest distance to a boundary of the cluster no. 12 = 2.36 ft

Crack Cluster no. 13 lies close to the slice-plane no. 1
Nearest distance to a boundary of the cluster no. 13 = 2.40 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 1

Figure A.3. Tools for Strength Estimation at Slice-1 on Aden Road Bridge Box Beam-3

Table A.2. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice-1

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C3, Undetected Crack (visible on texture)	Side - a	13,14, 15, 25	Unexposed	75%
C9, C10	Side - b	1, 16	Unexposed	75%
C11, C12, C13	End of the beam	Cracks at the end assumed not affecting strength	Unexposed	-

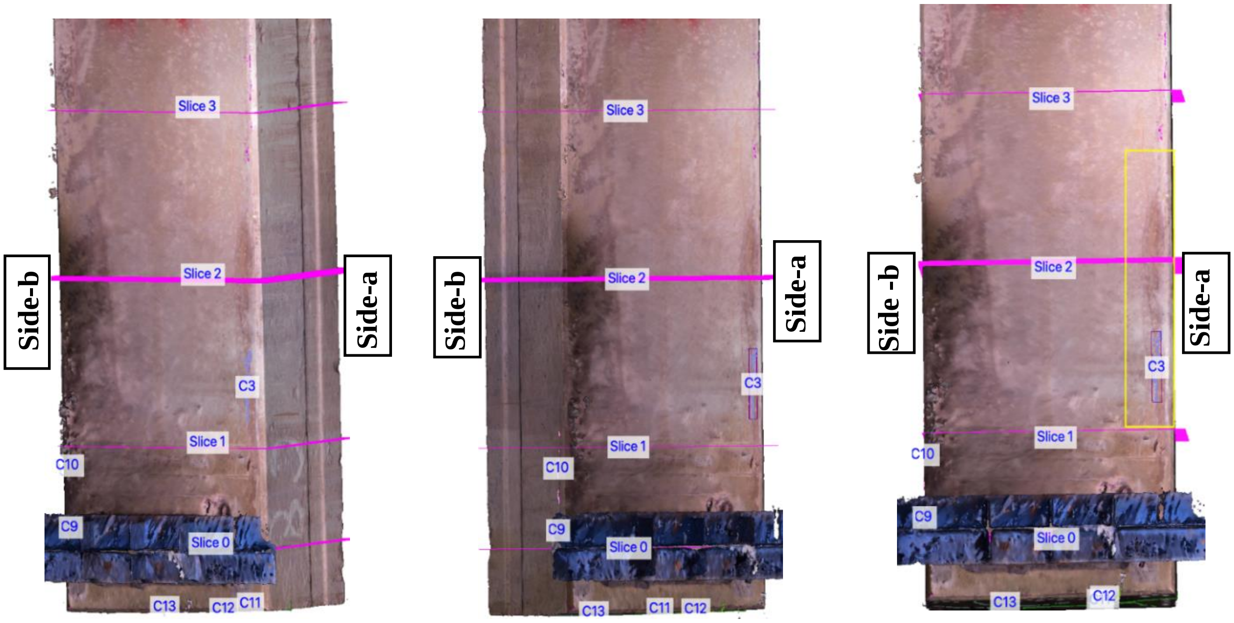
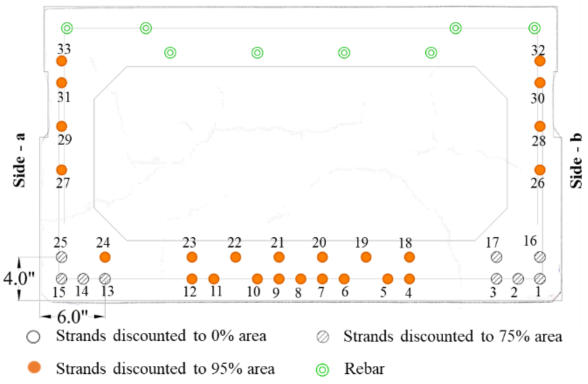


Figure A.4. Damage Clusters in the IW considered for Strength Estimation at Slice-1

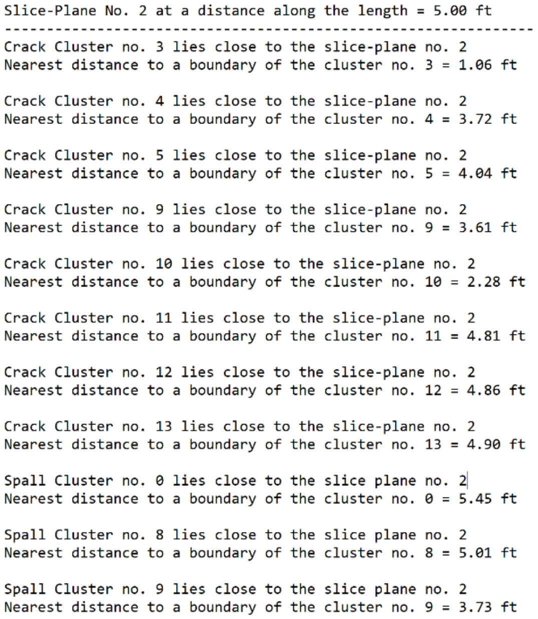
Slice No. 2 (At y = 5 ft from the left end of the beam)

This slice is the closest to the failure section in Test-1 on Box Beam-3. So, this corrosion damage and the reduced area of strands also applies to the section of failure in Test-1.



Side - a
Side - b
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
4.0"
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
6.0"
○ Strands discounted to 0% area
▨ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

a) Overlay of Slice No. 2 with the Cross-Section Drawing



```
Slice-Plane No. 2 at a distance along the length = 5.00 ft
--------------------------------------------------------------
Crack Cluster no. 3 lies close to the slice-plane no. 2
Nearest distance to a boundary of the cluster no. 3 = 1.06 ft

Crack Cluster no. 4 lies close to the slice-plane no. 2
Nearest distance to a boundary of the cluster no. 4 = 3.72 ft

Crack Cluster no. 5 lies close to the slice-plane no. 2
Nearest distance to a boundary of the cluster no. 5 = 4.04 ft

Crack Cluster no. 9 lies close to the slice-plane no. 2
Nearest distance to a boundary of the cluster no. 9 = 3.61 ft

Crack Cluster no. 10 lies close to the slice-plane no. 2
Nearest distance to a boundary of the cluster no. 10 = 2.28 ft

Crack Cluster no. 11 lies close to the slice-plane no. 2
Nearest distance to a boundary of the cluster no. 11 = 4.81 ft

Crack Cluster no. 12 lies close to the slice-plane no. 2
Nearest distance to a boundary of the cluster no. 12 = 4.86 ft

Crack Cluster no. 13 lies close to the slice-plane no. 2
Nearest distance to a boundary of the cluster no. 13 = 4.90 ft

Spall Cluster no. 0 lies close to the slice plane no. 2
Nearest distance to a boundary of the cluster no. 0 = 5.45 ft

Spall Cluster no. 8 lies close to the slice plane no. 2
Nearest distance to a boundary of the cluster no. 8 = 5.01 ft

Spall Cluster no. 9 lies close to the slice plane no. 2
Nearest distance to a boundary of the cluster no. 9 = 3.73 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 2

Figure A.5. Tools for Strength Estimation at Slice-2 on Aden Road Bridge Box Beam-3

Table A.3. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice 2

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C3, Undetected Crack (visible on texture)	Side - a	13, 14, 15, 25	Unexposed	75%
C4, C5, C9, C10	Side - b	1, 16	Unexposed	75%
Undetected Crack (visible on texture)	Side - b	2, 3, 17	Unexposed	75%
C11, C12, C13	End of the beam	Cracks at the end assumed to not affect strength	Unexposed	-
S0	Side - b	None affected	-	-
S8, S9	Noise	-	-	-
-	-	Other unexposed strands unaffected due to cracks		95%

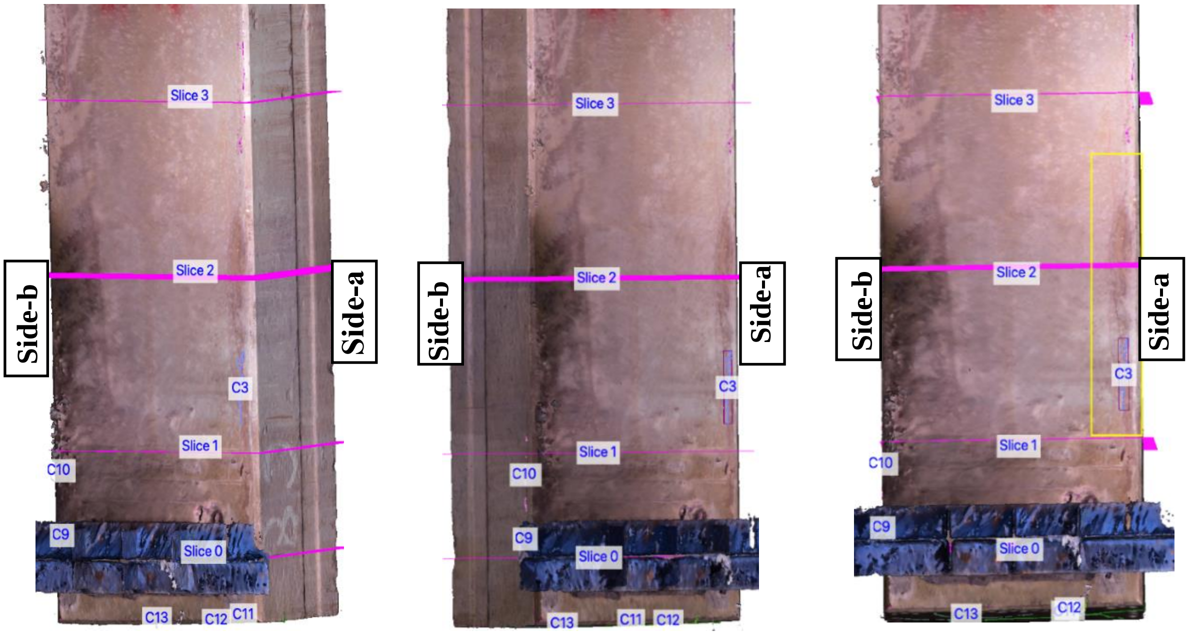
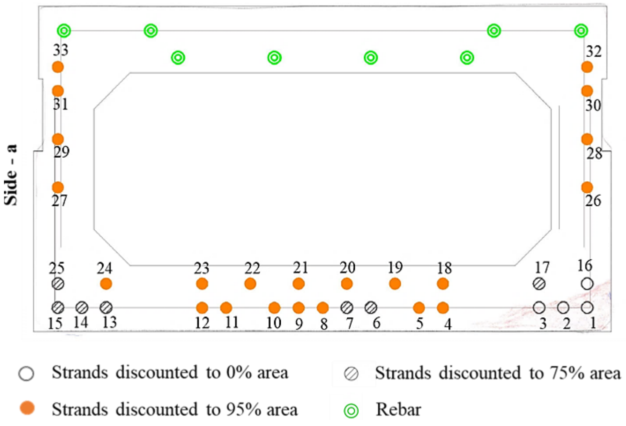


Figure A.6. Damage Clusters in the Inspection Window to be considered for Strength Estimation at Slice-2 of Box Beam-3

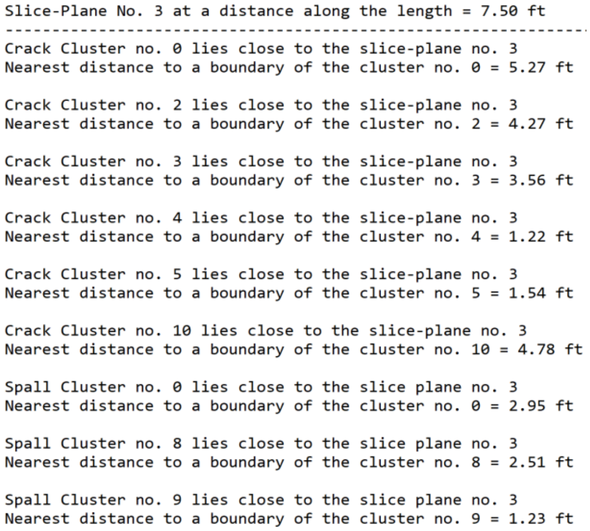
Table A.4. Calculation of Reduced Area of Strands for Residual Strength Estimation at Failure Section in Test-1 on Box Beam-3

Row	Distance from the bottom to each layer (in)	Number of strands in each layer	No. of strands discounted to 0% area	No. of strands discounted to 75% area	No. of strands discounted to 95% area	Area of Strands in Each Layer (in^2)
1	2	15	0	6	9	1.501
2	4	10	0	3	7	1.024
3	12	2	0	0	2	0.219
4	16	2	0	0	2	0.219
5	20	2	0	0	2	0.219
6	22	2	0	0	2	0.219
TOTAL		33				3.398

Slice No. 3 (At y = 7.5 ft from the left end of the beam)


Side - a
Side - b
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
○ Strands discounted to 0% area
▨ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

a) Overlay of Slice No. 3 with the Cross-Section Drawing



```
Slice-Plane No. 3 at a distance along the length = 7.50 ft
--------------------------------------------------------------
Crack Cluster no. 0 lies close to the slice-plane no. 3
Nearest distance to a boundary of the cluster no. 0 = 5.27 ft

Crack Cluster no. 2 lies close to the slice-plane no. 3
Nearest distance to a boundary of the cluster no. 2 = 4.27 ft

Crack Cluster no. 3 lies close to the slice-plane no. 3
Nearest distance to a boundary of the cluster no. 3 = 3.56 ft

Crack Cluster no. 4 lies close to the slice-plane no. 3
Nearest distance to a boundary of the cluster no. 4 = 1.22 ft

Crack Cluster no. 5 lies close to the slice-plane no. 3
Nearest distance to a boundary of the cluster no. 5 = 1.54 ft

Crack Cluster no. 10 lies close to the slice-plane no. 3
Nearest distance to a boundary of the cluster no. 10 = 4.78 ft

Spall Cluster no. 0 lies close to the slice plane no. 3
Nearest distance to a boundary of the cluster no. 0 = 2.95 ft

Spall Cluster no. 8 lies close to the slice plane no. 3
Nearest distance to a boundary of the cluster no. 8 = 2.51 ft

Spall Cluster no. 9 lies close to the slice plane no. 3
Nearest distance to a boundary of the cluster no. 9 = 1.23 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 3

Figure A.7. Tools for Strength Estimation at Slice-3 on Aden Road Bridge Box Beam-3

Table A.5. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 3

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C3, Undetected Crack (visible on texture)	Side - a	13,14, 15, 25	Unexposed	75%
C0	Middle	6, 7	Unexposed	75%
Undetected Crack (visible on texture)	Side - b	17	Unexposed	75%
C4, C5, C10	Side - b	1, 16	Exposed	0%
S0	Side - b	1, 2, 3, 16		
C2, S8, S9	Noise	-	-	-

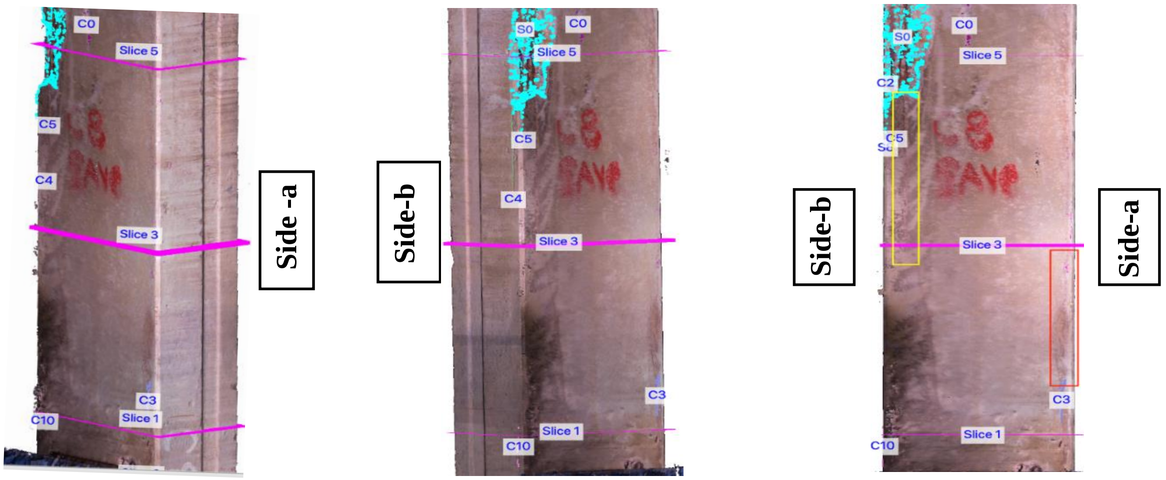
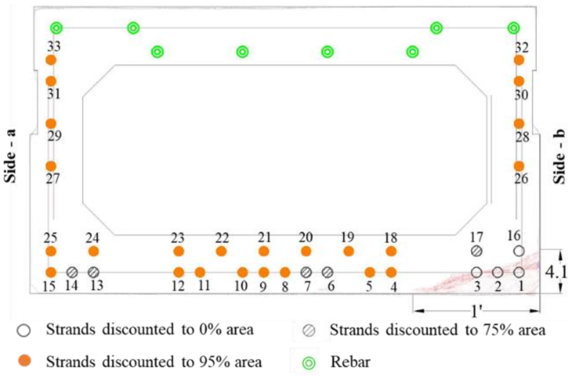
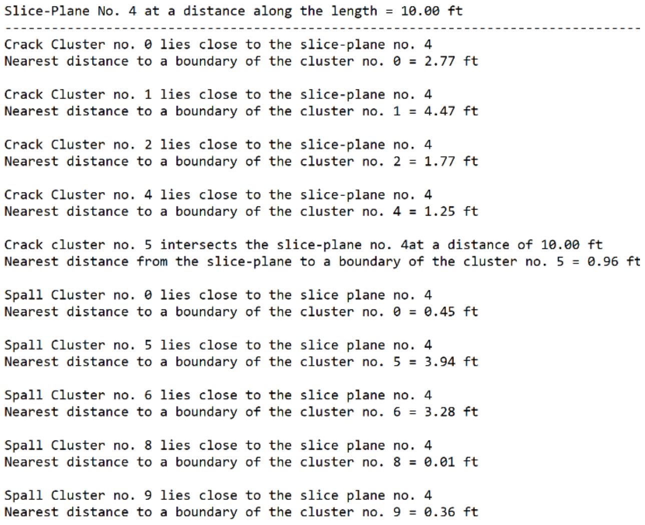


Figure A.8. Damage Clusters in the Inspection Window to be considered for Strength Estimation at Slice-3 of Box Beam-3

Slice No. 4 (At y = 10 ft from the left end of the beam)


Side - a
Side - b
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15 14 13
12 11
10 9 8 7 6
5 4
3 2 1
4.1
1'
○ Strands discounted to 0% area
▨ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

a) Overlay of Slice No. 4 with the Cross-Section Drawing



```
Slice-Plane No. 4 at a distance along the length = 10.00 ft
-------------------------------------------------------------------------------------
Crack Cluster no. 0 lies close to the slice-plane no. 4
Nearest distance to a boundary of the cluster no. 0 = 2.77 ft

Crack Cluster no. 1 lies close to the slice-plane no. 4
Nearest distance to a boundary of the cluster no. 1 = 4.47 ft

Crack Cluster no. 2 lies close to the slice-plane no. 4
Nearest distance to a boundary of the cluster no. 2 = 1.77 ft

Crack Cluster no. 4 lies close to the slice-plane no. 4
Nearest distance to a boundary of the cluster no. 4 = 1.25 ft

Crack cluster no. 5 intersects the slice-plane no. 4at a distance of 10.00 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 5 = 0.96 ft

Spall Cluster no. 0 lies close to the slice plane no. 4
Nearest distance to a boundary of the cluster no. 0 = 0.45 ft

Spall Cluster no. 5 lies close to the slice plane no. 4
Nearest distance to a boundary of the cluster no. 5 = 3.94 ft

Spall Cluster no. 6 lies close to the slice plane no. 4
Nearest distance to a boundary of the cluster no. 6 = 3.28 ft

Spall Cluster no. 8 lies close to the slice plane no. 4
Nearest distance to a boundary of the cluster no. 8 = 0.01 ft

Spall Cluster no. 9 lies close to the slice plane no. 4
Nearest distance to a boundary of the cluster no. 9 = 0.36 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 4

Figure A.9. Tools for Strength Estimation at Slice-4 on Aden Road Bridge Box Beam-3

Table A.6. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 4

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
Undetected Crack (visible on texture)	Side - a	13,14	Unexposed	75%
C0, C1, S5, S6	Middle	6, 7	Unexposed	75%
Undetected Crack (visible on texture)	Side - b	17	Unexposed	75%
C4, C5	Side - b	1, 16	Exposed	0%
S0	Side - b	1, 2, 3, 16		
C2, S8, S9	Noise	-	-	-

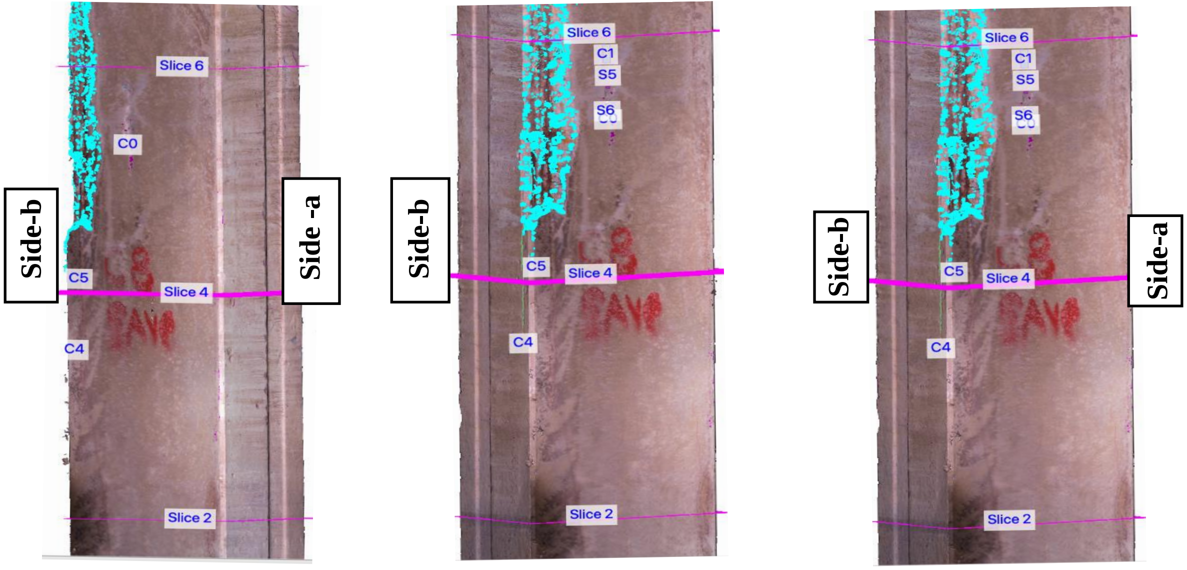
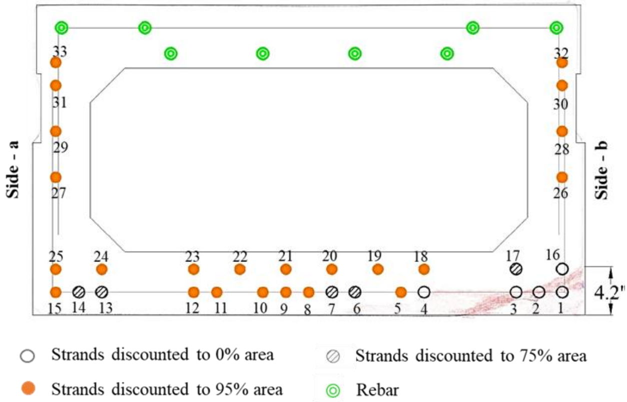
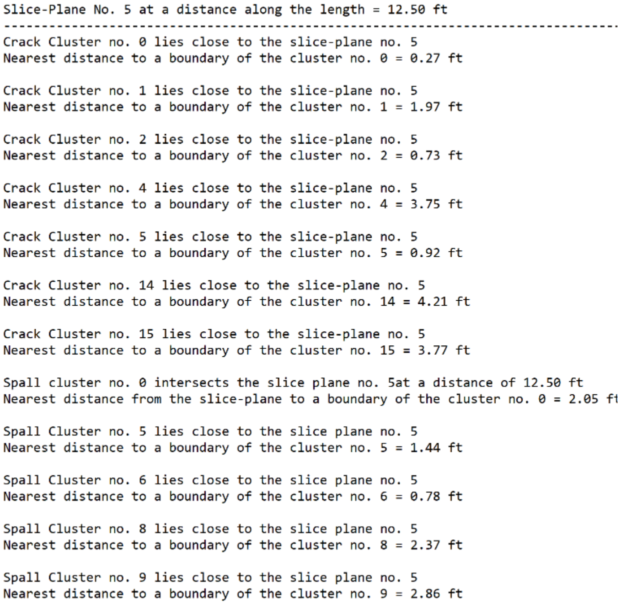


Figure A.10. Damage Clusters in the IW to be considered for Strength Estimation at Slice-4

Slice No. 5 (At y = 12.5 ft from the left end of the beam)


Side - a
Side - b
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
4.2"
○ Strands discounted to 0% area
▨ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

a) Overlay of Slice No. 5 with the Cross-Section Drawing



```
Slice-Plane No. 5 at a distance along the length = 12.50 ft
--------------------------------------------------------------------------------
Crack Cluster no. 0 lies close to the slice-plane no. 5
Nearest distance to a boundary of the cluster no. 0 = 0.27 ft

Crack Cluster no. 1 lies close to the slice-plane no. 5
Nearest distance to a boundary of the cluster no. 1 = 1.97 ft

Crack Cluster no. 2 lies close to the slice-plane no. 5
Nearest distance to a boundary of the cluster no. 2 = 0.73 ft

Crack Cluster no. 4 lies close to the slice-plane no. 5
Nearest distance to a boundary of the cluster no. 4 = 3.75 ft

Crack Cluster no. 5 lies close to the slice-plane no. 5
Nearest distance to a boundary of the cluster no. 5 = 0.92 ft

Crack Cluster no. 14 lies close to the slice-plane no. 5
Nearest distance to a boundary of the cluster no. 14 = 4.21 ft

Crack Cluster no. 15 lies close to the slice-plane no. 5
Nearest distance to a boundary of the cluster no. 15 = 3.77 ft

Spall cluster no. 0 intersects the slice plane no. 5at a distance of 12.50 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 0 = 2.05 ft

Spall Cluster no. 5 lies close to the slice plane no. 5
Nearest distance to a boundary of the cluster no. 5 = 1.44 ft

Spall Cluster no. 6 lies close to the slice plane no. 5
Nearest distance to a boundary of the cluster no. 6 = 0.78 ft

Spall Cluster no. 8 lies close to the slice plane no. 5
Nearest distance to a boundary of the cluster no. 8 = 2.37 ft

Spall Cluster no. 9 lies close to the slice plane no. 5
Nearest distance to a boundary of the cluster no. 9 = 2.86 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 5

Figure A.11. Tools for Strength Estimation at Slice-5 on Aden Road Bridge Box Beam-3

Table A.7. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 5

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
Undetected Crack (visible on texture)	Side - a	13,14	Unexposed	75%
C0, C1, S5, S6, C14	Middle	6, 7	Unexposed	75%
Undetected Crack (visible on texture)	Side - b	17	Unexposed	75%
C4, C5	Side - b	1, 16	Exposed	0%
S0	Side - b	1, 2, 3, 4, 16		
C2, S8, S9, C15	Noise	-	-	-

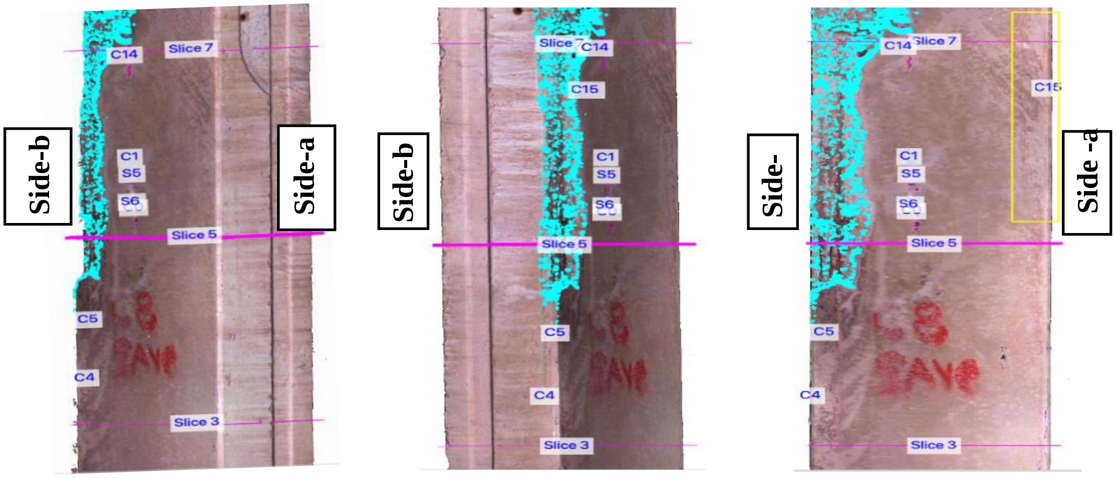
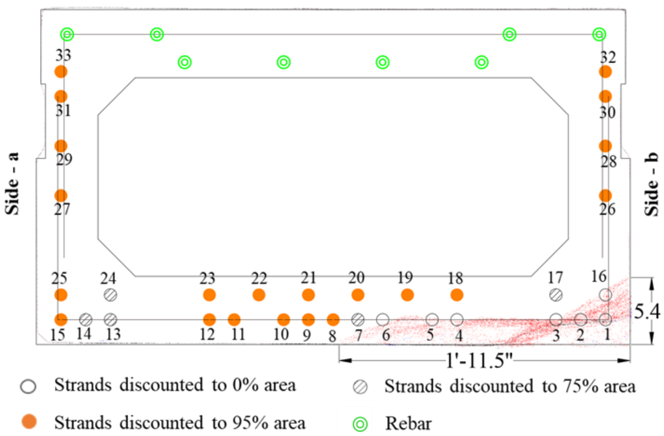
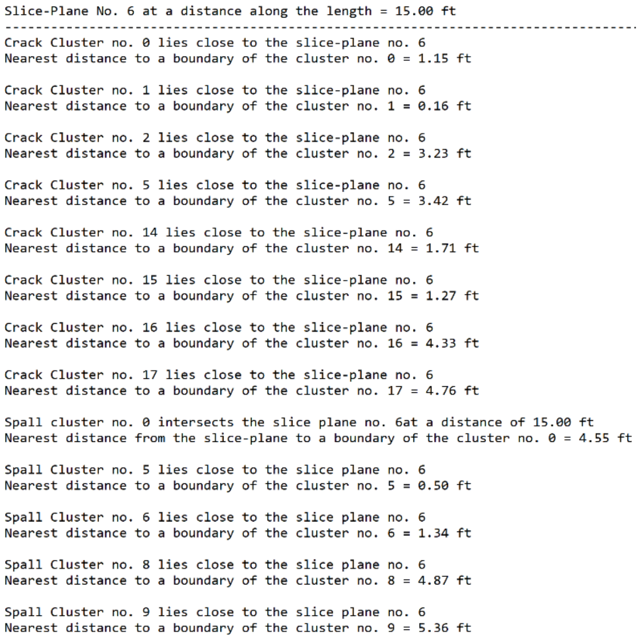


Figure A.12. Damage Clusters in the IW to be considered for Strength Estimation at Slice-5

Slice No. 6 (At y = 15 ft from the left end of the beam)


Side - a
Side - b
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
5.4
1'-11.5"
Strands discounted to 0% area
Strands discounted to 75% area
Strands discounted to 95% area
Rebar

a) Overlay of Slice No. 6 with the Cross-Section Drawing



```
Slice-Plane No. 6 at a distance along the length = 15.00 ft
------------------------------------------------------------------------------------------
Crack Cluster no. 0 lies close to the slice-plane no. 6
Nearest distance to a boundary of the cluster no. 0 = 1.15 ft

Crack Cluster no. 1 lies close to the slice-plane no. 6
Nearest distance to a boundary of the cluster no. 1 = 0.16 ft

Crack Cluster no. 2 lies close to the slice-plane no. 6
Nearest distance to a boundary of the cluster no. 2 = 3.23 ft

Crack Cluster no. 5 lies close to the slice-plane no. 6
Nearest distance to a boundary of the cluster no. 5 = 3.42 ft

Crack Cluster no. 14 lies close to the slice-plane no. 6
Nearest distance to a boundary of the cluster no. 14 = 1.71 ft

Crack Cluster no. 15 lies close to the slice-plane no. 6
Nearest distance to a boundary of the cluster no. 15 = 1.27 ft

Crack Cluster no. 16 lies close to the slice-plane no. 6
Nearest distance to a boundary of the cluster no. 16 = 4.33 ft

Crack Cluster no. 17 lies close to the slice-plane no. 6
Nearest distance to a boundary of the cluster no. 17 = 4.76 ft

Spall cluster no. 0 intersects the slice plane no. 6at a distance of 15.00 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 0 = 4.55 ft

Spall Cluster no. 5 lies close to the slice plane no. 6
Nearest distance to a boundary of the cluster no. 5 = 0.50 ft

Spall Cluster no. 6 lies close to the slice plane no. 6
Nearest distance to a boundary of the cluster no. 6 = 1.34 ft

Spall Cluster no. 8 lies close to the slice plane no. 6
Nearest distance to a boundary of the cluster no. 8 = 4.87 ft

Spall Cluster no. 9 lies close to the slice plane no. 6
Nearest distance to a boundary of the cluster no. 9 = 5.36 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 6

Figure A.13. Tools for Strength Estimation at Slice-6 on Aden Road Bridge Box Beam-3

Table A.8. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 6

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C16, C17	Side - a	13,14	Unexposed	75%
C0, C1, S5, S6, C14	Middle	7	Unexposed	75%
Undetected Crack (from texture)	Side - b	17	Unexposed	75%
C5	Side - b	1, 16	Exposed	0%
S0	Side - b	1, 2, 3, 4, 5, 6 ,16		
C2, S8, S9, C15	Noise	-	-	-

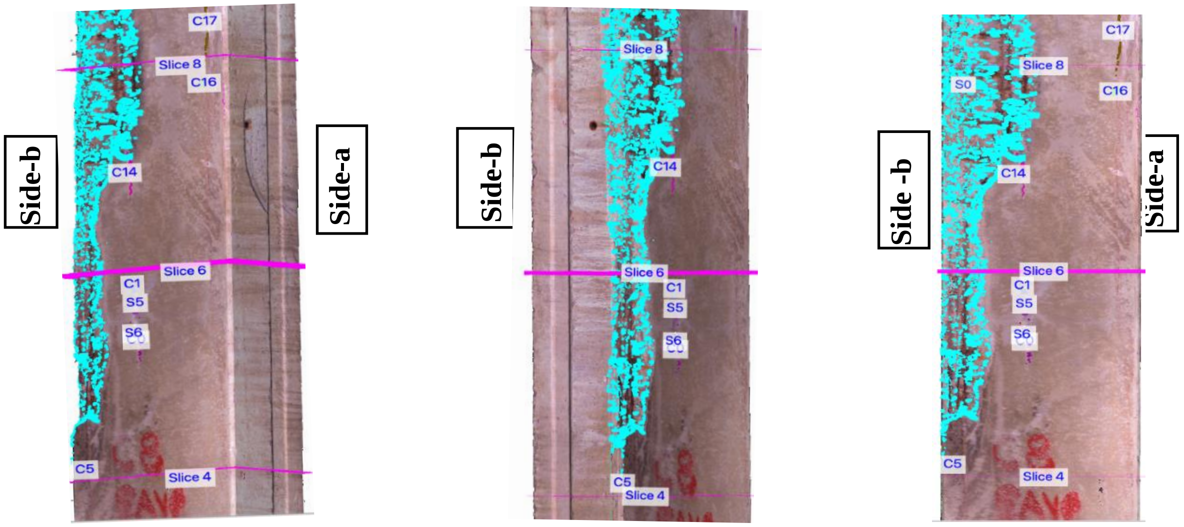
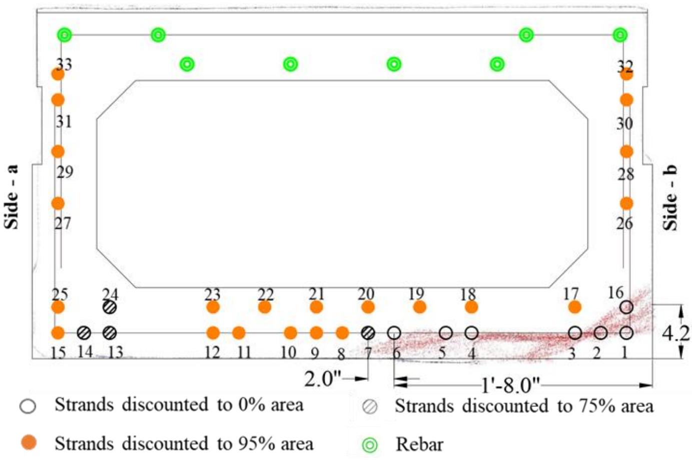


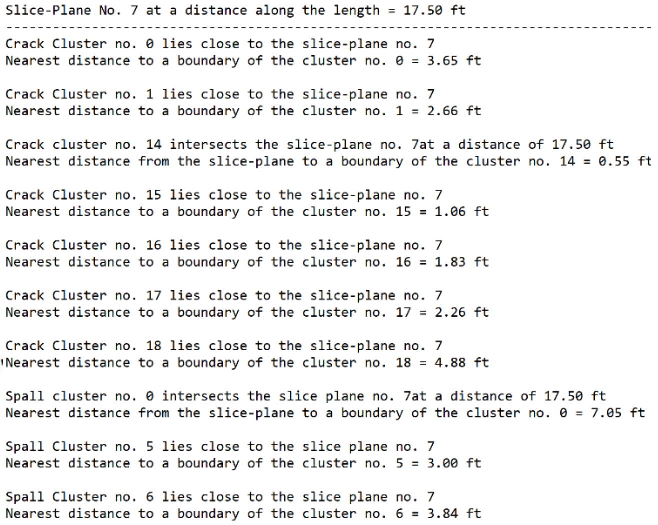
Figure A.14. Damage Clusters in the IW to be considered for Strength Estimation at Slice-6

Slice No. 7 (At y = 17.5 ft from the left end)



Side - a
Side - b
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
4.2"
2.0"
1'-8.0"
○ Strands discounted to 0% area
▨ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

a) Overlay of Slice No. 7 with the Cross-Section Drawing



```
Slice-Plane No. 7 at a distance along the length = 17.50 ft
-----------------------------------------------------------------------------------------
Crack Cluster no. 0 lies close to the slice-plane no. 7
Nearest distance to a boundary of the cluster no. 0 = 3.65 ft

Crack Cluster no. 1 lies close to the slice-plane no. 7
Nearest distance to a boundary of the cluster no. 1 = 2.66 ft

Crack cluster no. 14 intersects the slice-plane no. 7at a distance of 17.50 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 14 = 0.55 ft

Crack Cluster no. 15 lies close to the slice-plane no. 7
Nearest distance to a boundary of the cluster no. 15 = 1.06 ft

Crack Cluster no. 16 lies close to the slice-plane no. 7
Nearest distance to a boundary of the cluster no. 16 = 1.83 ft

Crack Cluster no. 17 lies close to the slice-plane no. 7
Nearest distance to a boundary of the cluster no. 17 = 2.26 ft

Crack Cluster no. 18 lies close to the slice-plane no. 7
Nearest distance to a boundary of the cluster no. 18 = 4.88 ft

Spall cluster no. 0 intersects the slice plane no. 7at a distance of 17.50 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 0 = 7.05 ft

Spall Cluster no. 5 lies close to the slice plane no. 7
Nearest distance to a boundary of the cluster no. 5 = 3.00 ft

Spall Cluster no. 6 lies close to the slice plane no. 7
Nearest distance to a boundary of the cluster no. 6 = 3.84 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 7

Figure A.15. Tools for Strength Estimation at Slice-7 on Aden Road Bridge Box Beam-3

Table A.9. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 7

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C16, C17, C18	Side - a	13,14, 24	Unexposed	75%
C0, C1, C14, S5, S6	Middle	7	Unexposed	75%
S0	Side - b	1, 2, 3, 4, 5, 6 ,16	Exposed	0%
C15	Noise	-	-	-

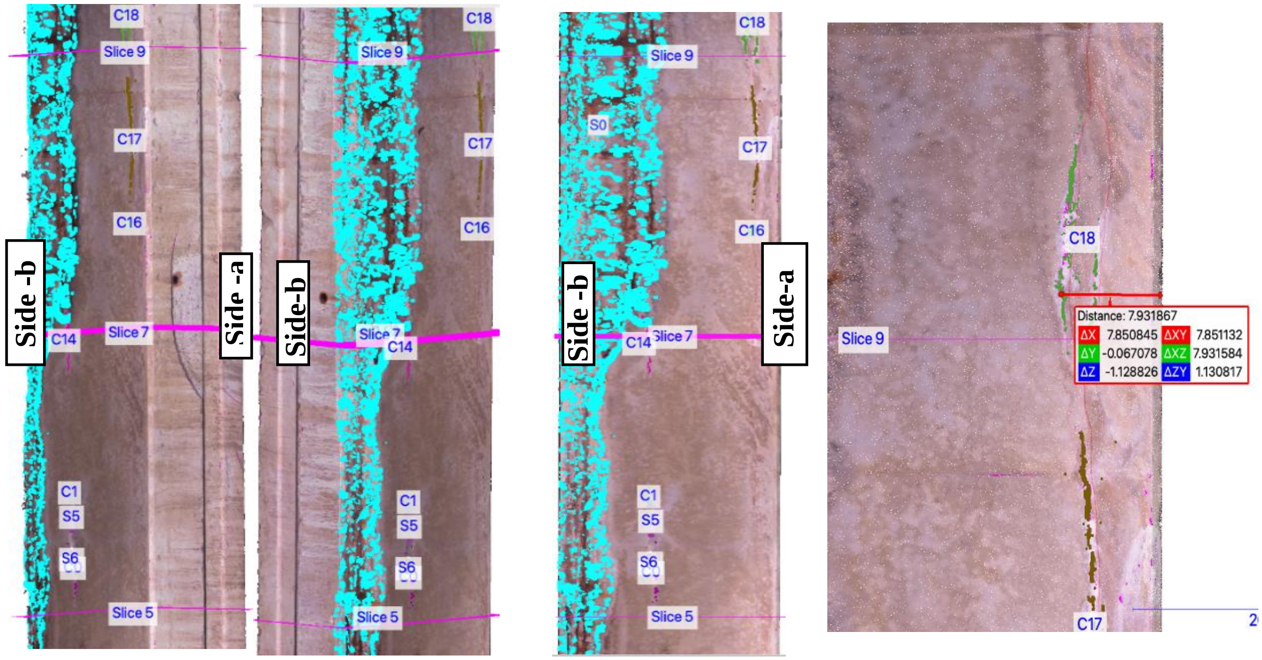
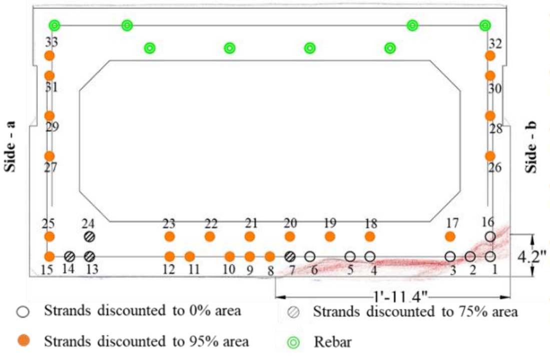
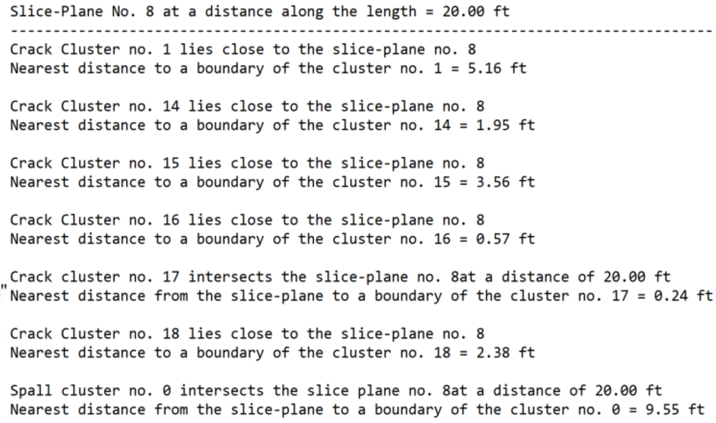


Figure A.16. Damage Clusters in the IW to be considered for Strength Estimation at Slice-7

Slice No. 8 (At y = 20 ft from the left end)



a) Overlay of Slice No. 8 with the Cross-Section Drawing



```
Slice-Plane No. 8 at a distance along the length = 20.00 ft
----------------------------------------------------------------------------------
Crack Cluster no. 1 lies close to the slice-plane no. 8
Nearest distance to a boundary of the cluster no. 1 = 5.16 ft

Crack Cluster no. 14 lies close to the slice-plane no. 8
Nearest distance to a boundary of the cluster no. 14 = 1.95 ft

Crack Cluster no. 15 lies close to the slice-plane no. 8
Nearest distance to a boundary of the cluster no. 15 = 3.56 ft

Crack Cluster no. 16 lies close to the slice-plane no. 8
Nearest distance to a boundary of the cluster no. 16 = 0.57 ft

Crack cluster no. 17 intersects the slice-plane no. 8at a distance of 20.00 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 17 = 0.24 ft

Crack Cluster no. 18 lies close to the slice-plane no. 8
Nearest distance to a boundary of the cluster no. 18 = 2.38 ft

Spall cluster no. 0 intersects the slice plane no. 8at a distance of 20.00 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 0 = 9.55 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 8

Figure A.17. Tools for Strength Estimation at Slice-8 on Aden Road Bridge Box Beam-3

Table A.10. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 8

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C16, C17, C18	Side - a	13,14, 24	Unexposed	75%
C1, C14, S5, S6	Middle	7	Unexposed	75%
S0	Side - b	1, 2, 3, 4, 5, 6 ,16	Exposed	0%
C15	Noise	-	-	-

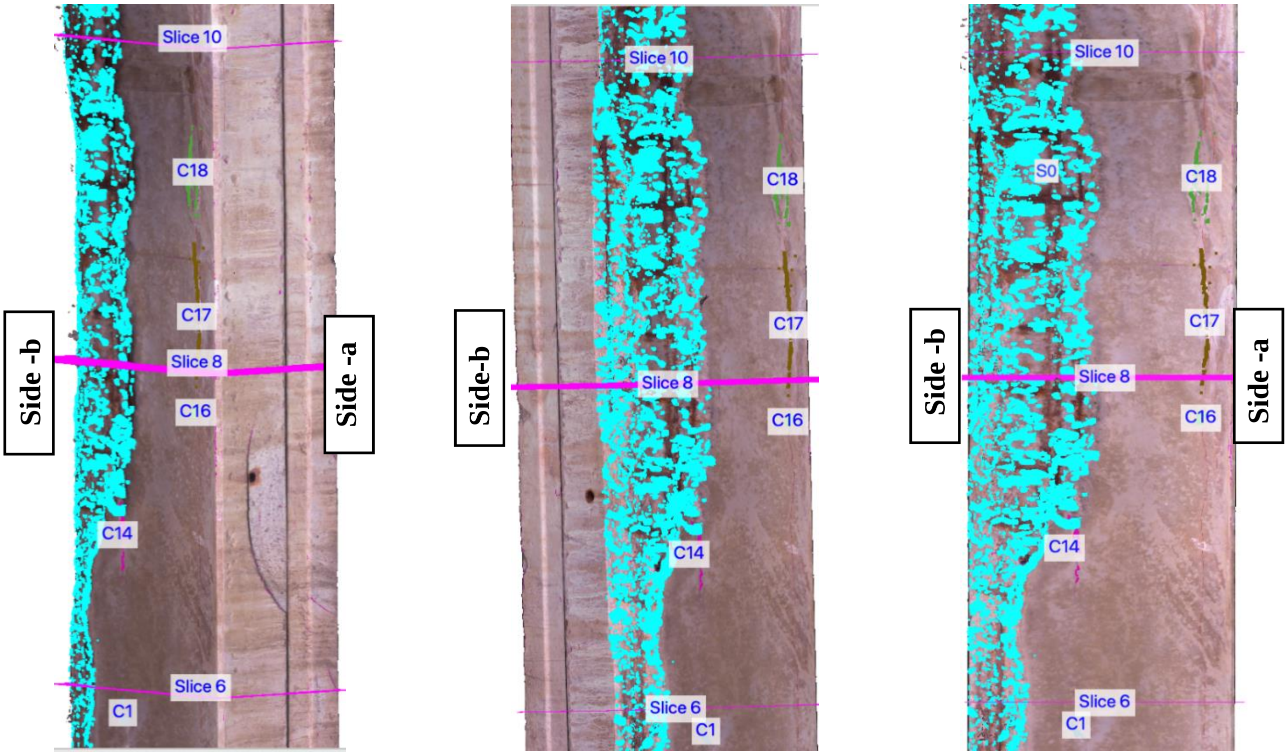
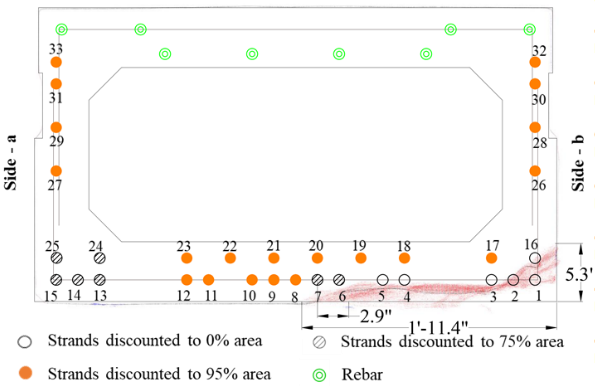
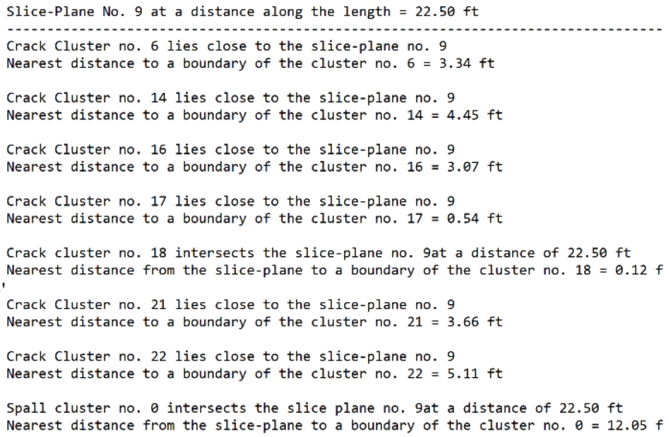


Figure A.18. Damage Clusters in the IW to be considered for Strength Estimation at Slice-8

Slice No. 9 (At y = 22.5 ft from the left end)



a) Overlay of Slice No. 9 with the Cross-Section Drawing



```
Slice-Plane No. 9 at a distance along the length = 22.50 ft
--------------------------------------------------------------------------------
Crack Cluster no. 6 lies close to the slice-plane no. 9
Nearest distance to a boundary of the cluster no. 6 = 3.34 ft

Crack Cluster no. 14 lies close to the slice-plane no. 9
Nearest distance to a boundary of the cluster no. 14 = 4.45 ft

Crack Cluster no. 16 lies close to the slice-plane no. 9
Nearest distance to a boundary of the cluster no. 16 = 3.07 ft

Crack Cluster no. 17 lies close to the slice-plane no. 9
Nearest distance to a boundary of the cluster no. 17 = 0.54 ft

Crack cluster no. 18 intersects the slice-plane no. 9at a distance of 22.50 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 18 = 0.12 f

Crack Cluster no. 21 lies close to the slice-plane no. 9
Nearest distance to a boundary of the cluster no. 21 = 3.66 ft

Crack Cluster no. 22 lies close to the slice-plane no. 9
Nearest distance to a boundary of the cluster no. 22 = 5.11 ft

Spall cluster no. 0 intersects the slice plane no. 9at a distance of 22.50 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 0 = 12.05 f
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 9

Figure A.19. Tools for Strength Estimation at Slice-9 on Aden Road Bridge Box Beam-3

Table A.11. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 9

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C6	Side - a	15, 25	Unexposed	75%
C16, C17, C18	Side - a	13,14, 24	Unexposed	75%
C14	Middle	6, 7	Unexposed	75%
C21	Side - b	1, 16	Exposed	
S0	Side - b	1, 2, 3, 4, 5, 16	Exposed	0%
C22	Noise	-	-	-

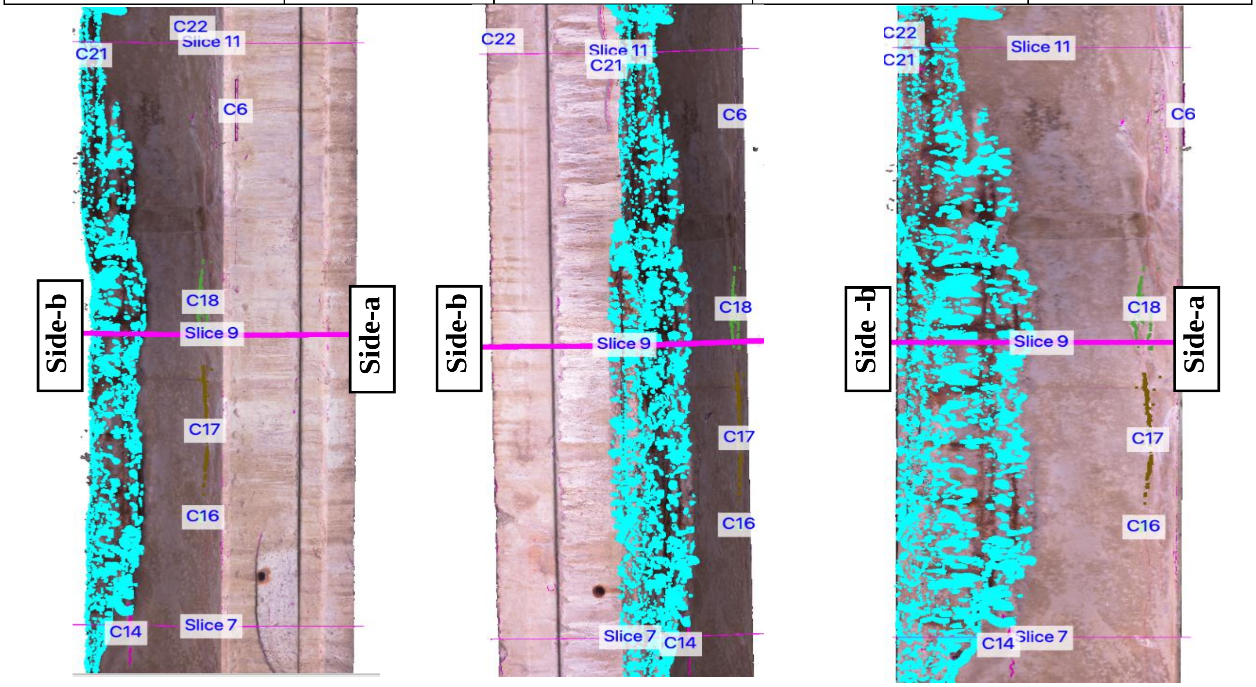
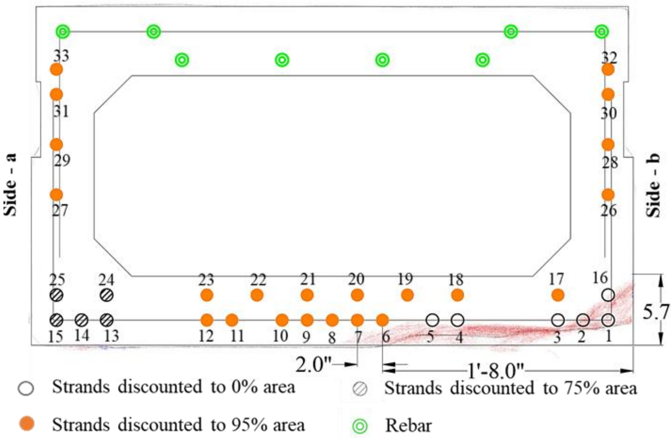
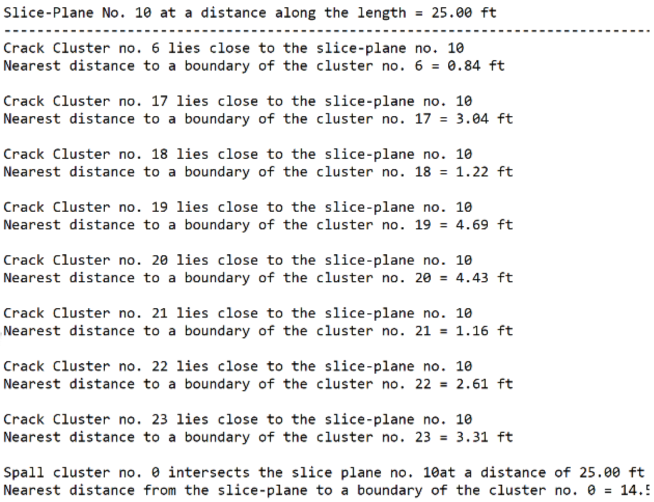


Figure A.20. Damage Clusters in the IW to be considered for Strength Estimation at Slice-9

Slice No. 10 (At y = 25 ft from the left end)


Side - a
Side - b
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
5.7'
2.0"
1'-8.0"
○ Strands discounted to 0% area
▨ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

a) Overlay of Slice No. 10 with the Cross-Section Drawing



```
Slice-Plane No. 10 at a distance along the length = 25.00 ft
----------------------------------------------------------------------------------
Crack Cluster no. 6 lies close to the slice-plane no. 10
Nearest distance to a boundary of the cluster no. 6 = 0.84 ft

Crack Cluster no. 17 lies close to the slice-plane no. 10
Nearest distance to a boundary of the cluster no. 17 = 3.04 ft

Crack Cluster no. 18 lies close to the slice-plane no. 10
Nearest distance to a boundary of the cluster no. 18 = 1.22 ft

Crack Cluster no. 19 lies close to the slice-plane no. 10
Nearest distance to a boundary of the cluster no. 19 = 4.69 ft

Crack Cluster no. 20 lies close to the slice-plane no. 10
Nearest distance to a boundary of the cluster no. 20 = 4.43 ft

Crack Cluster no. 21 lies close to the slice-plane no. 10
Nearest distance to a boundary of the cluster no. 21 = 1.16 ft

Crack Cluster no. 22 lies close to the slice-plane no. 10
Nearest distance to a boundary of the cluster no. 22 = 2.61 ft

Crack Cluster no. 23 lies close to the slice-plane no. 10
Nearest distance to a boundary of the cluster no. 23 = 3.31 ft

Spall cluster no. 0 intersects the slice plane no. 10at a distance of 25.00 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 0 = 14.5
```

b) Portion of the Text File Showing the Damage Clusters for Slice No.10

Figure A.21. Tools for Strength Estimation at Slice-10 on Aden Road Bridge Box Beam-3

Table A.12. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 10

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C6, C20	Side - a	15, 25	Unexposed	75%
C17, C18, C19	Side - a	13,14, 24	Unexposed	75%
C21	Side - b	1, 16	Exposed	0%
S0	Side - b	1, 2, 3, 4, 5, 16	Exposed	
C22, C23	Noise	-	-	-

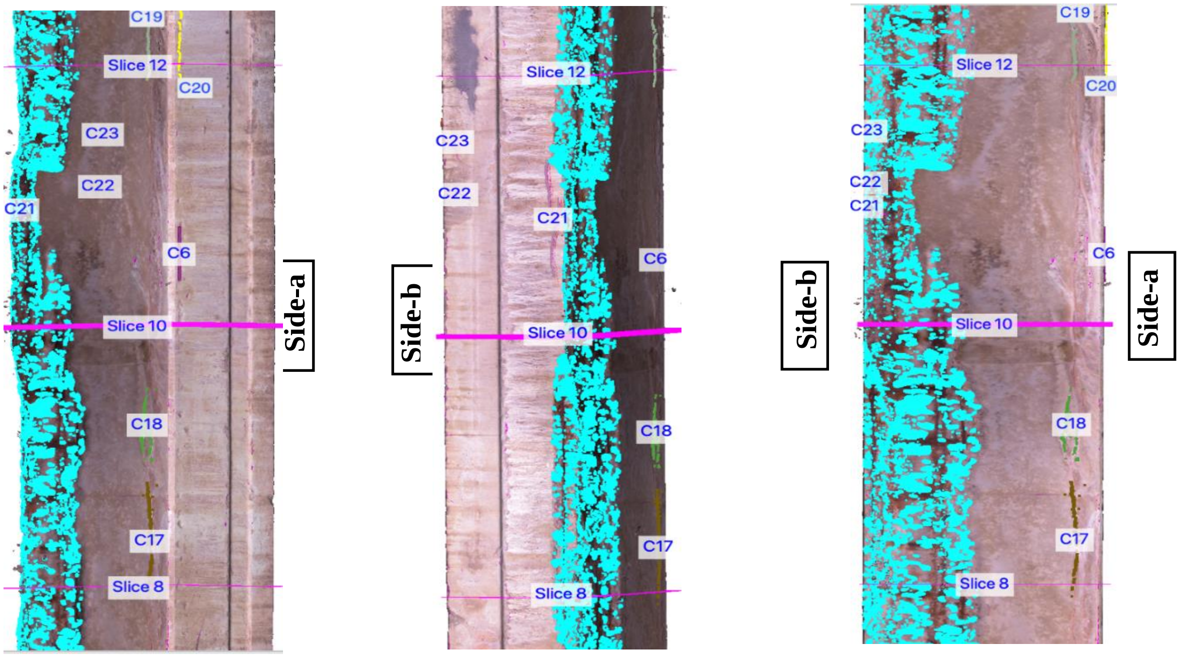
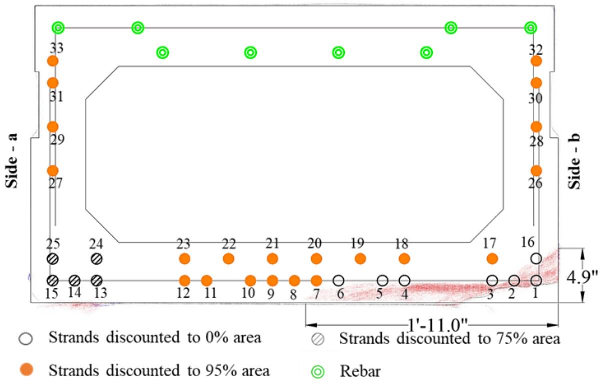
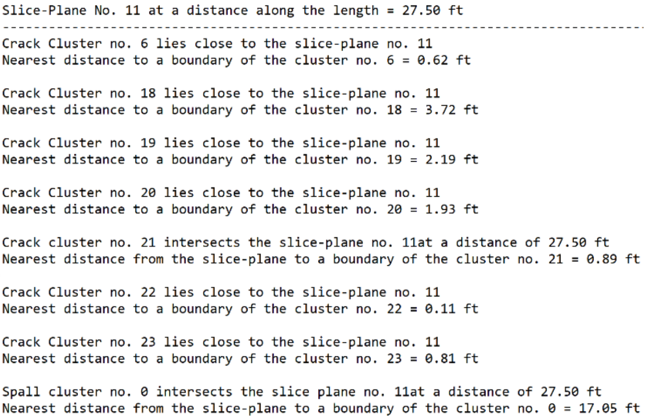


Figure A.22. Damage Clusters in the IW to be considered for Strength Estimation at Slice-10

Slice No. 11 (At y = 27.5 ft from the left end)


33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
Side - a
Side - b
4.9"
1'-11.0"
○ Strands discounted to 0% area
⊘ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

a) Overlay of Slice No. 11 with the Cross-Section Drawing



```
Slice-Plane No. 11 at a distance along the length = 27.50 ft
--------------------------------------------------------------------------------
Crack Cluster no. 6 lies close to the slice-plane no. 11
Nearest distance to a boundary of the cluster no. 6 = 0.62 ft

Crack Cluster no. 18 lies close to the slice-plane no. 11
Nearest distance to a boundary of the cluster no. 18 = 3.72 ft

Crack Cluster no. 19 lies close to the slice-plane no. 11
Nearest distance to a boundary of the cluster no. 19 = 2.19 ft

Crack Cluster no. 20 lies close to the slice-plane no. 11
Nearest distance to a boundary of the cluster no. 20 = 1.93 ft

Crack cluster no. 21 intersects the slice-plane no. 11at a distance of 27.50 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 21 = 0.89 ft

Crack Cluster no. 22 lies close to the slice-plane no. 11
Nearest distance to a boundary of the cluster no. 22 = 0.11 ft

Crack Cluster no. 23 lies close to the slice-plane no. 11
Nearest distance to a boundary of the cluster no. 23 = 0.81 ft

Spall cluster no. 0 intersects the slice plane no. 11at a distance of 27.50 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 0 = 17.05 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 11

Figure A.23. Tools for Strength Estimation at Slice-11 on Aden Road Bridge Box Beam-3

Table A.13. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 11

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C6, C20	Side - a	15, 25	Unexposed	75%
C18, C19	Side - a	13,14, 24	Unexposed	75%
C21	Side - b	1, 16	Exposed	0%
S0	Side - b	1, 2, 3, 4, 5, 6, 16	Exposed	
C22, C23	Noise	-	-	-

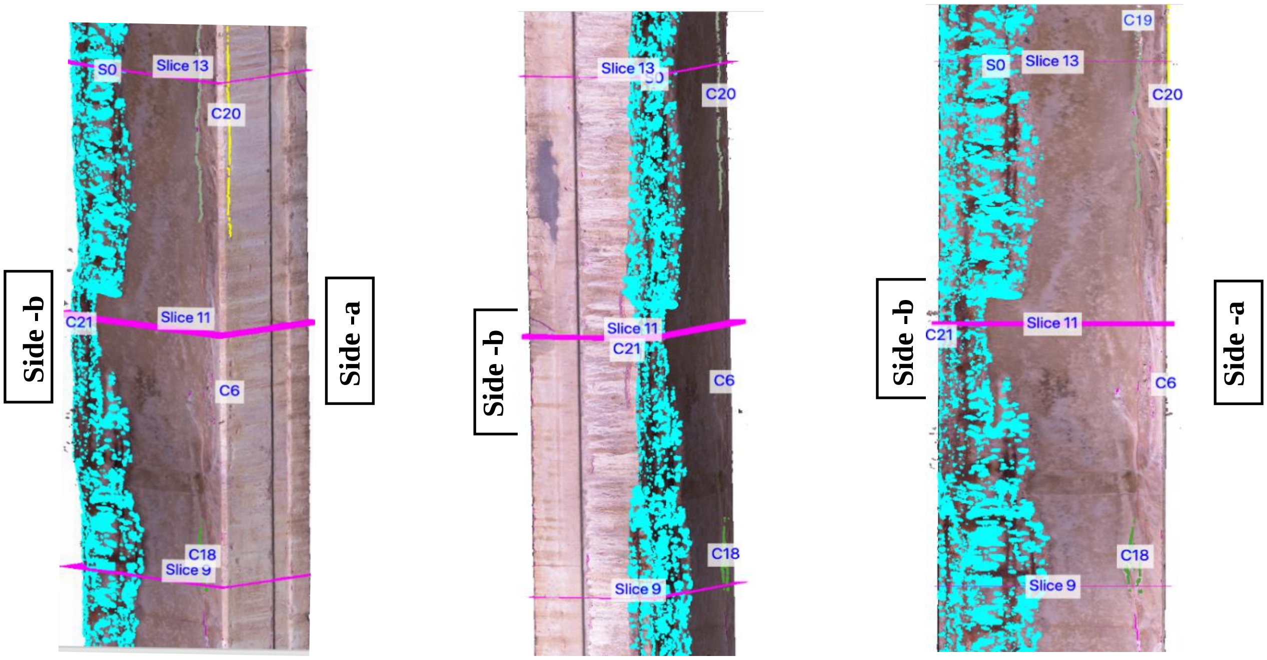
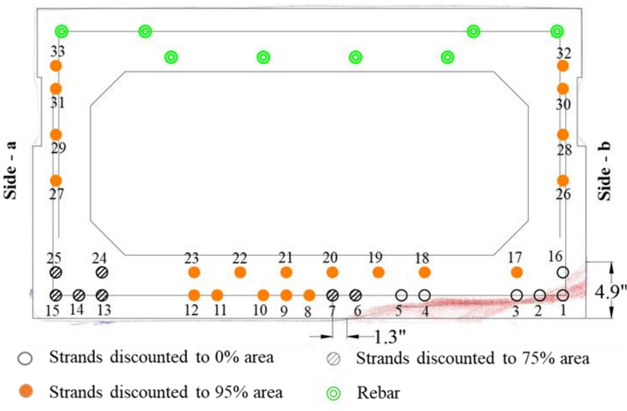
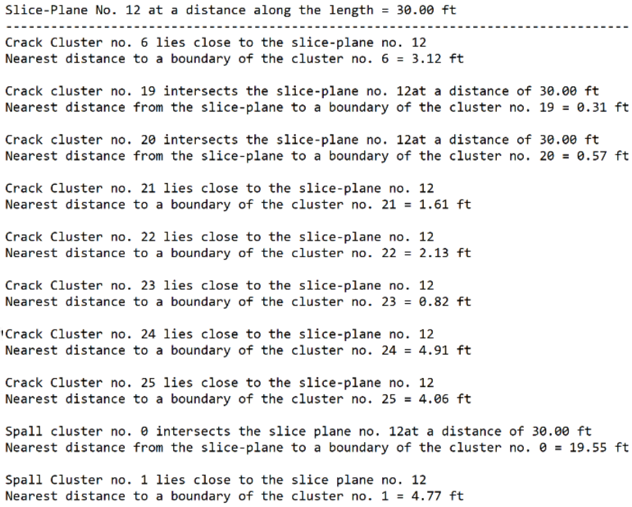


Figure A.24. Damage Clusters in IW considered for Strength Estimation at Slice-11 of Box Beam-3

Slice No. 12 (At y = 30 ft from the left end)


Side - a
Side - b
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
4.9"
1.3"
○ Strands discounted to 0% area
▨ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

a) Overlay of Slice No. 12 with the Cross-Section Drawing



```
Slice-Plane No. 12 at a distance along the length = 30.00 ft
--------------------------------------------------------------------------------
Crack Cluster no. 6 lies close to the slice-plane no. 12
Nearest distance to a boundary of the cluster no. 6 = 3.12 ft

Crack cluster no. 19 intersects the slice-plane no. 12at a distance of 30.00 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 19 = 0.31 ft

Crack cluster no. 20 intersects the slice-plane no. 12at a distance of 30.00 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 20 = 0.57 ft

Crack Cluster no. 21 lies close to the slice-plane no. 12
Nearest distance to a boundary of the cluster no. 21 = 1.61 ft

Crack Cluster no. 22 lies close to the slice-plane no. 12
Nearest distance to a boundary of the cluster no. 22 = 2.13 ft

Crack Cluster no. 23 lies close to the slice-plane no. 12
Nearest distance to a boundary of the cluster no. 23 = 0.82 ft

Crack Cluster no. 24 lies close to the slice-plane no. 12
Nearest distance to a boundary of the cluster no. 24 = 4.91 ft

Crack Cluster no. 25 lies close to the slice-plane no. 12
Nearest distance to a boundary of the cluster no. 25 = 4.06 ft

Spall cluster no. 0 intersects the slice plane no. 12at a distance of 30.00 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 0 = 19.55 ft

Spall Cluster no. 1 lies close to the slice plane no. 12
Nearest distance to a boundary of the cluster no. 1 = 4.77 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 12

Figure A.25. Tools for Strength Estimation at Slice-12 on Aden Road Bridge Box Beam-3

Table A.14. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 12

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C6, C20, C24	Side - a	15, 25	Unexposed	75%
C19	Side - a	13,14, 24	Unexposed	75%
C25	Middle	6, 7	Unexposed	75%
C21	Side - b	1, 16	Exposed	0%
S0	Side - b	1, 2, 3, 4, 5, 16		
C22, C23, S1	Noise	-	-	-

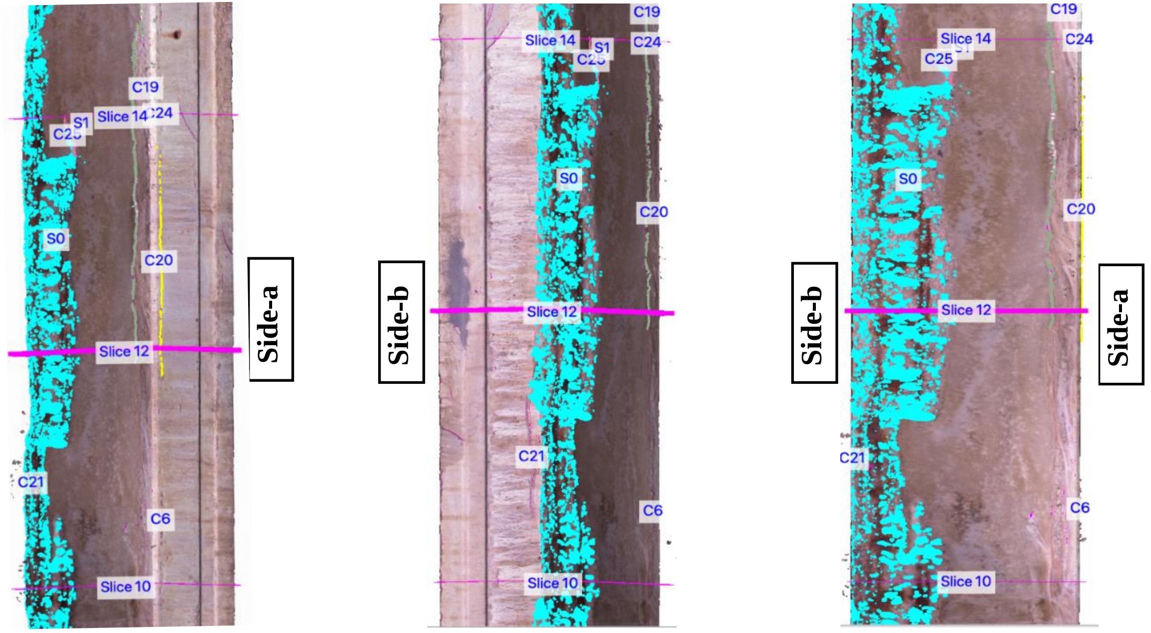
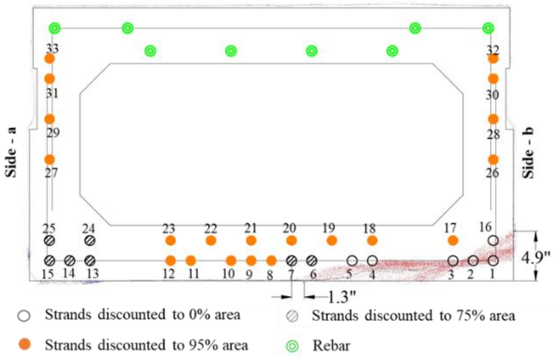
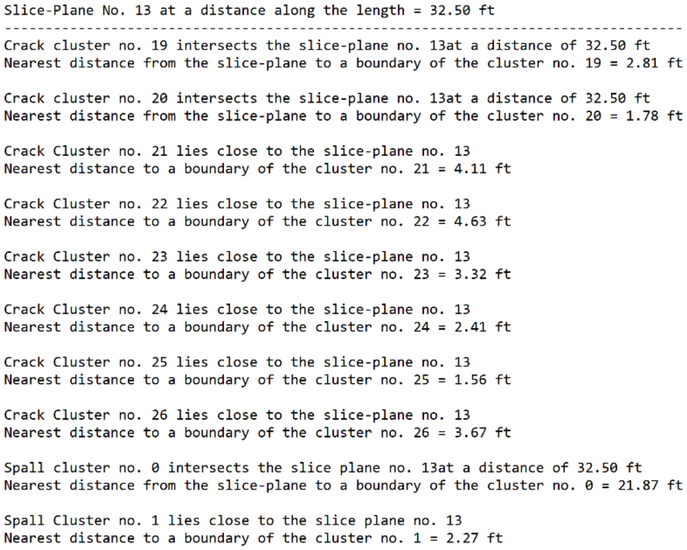


Figure A.26. Damage Clusters in IW considered for Strength Estimation at Slice-12 of Box Beam-3

Slice No. 13 (At y = 32.5 ft from the left end)


33
32
31
30
29
28
27
26
Side - a
Side - b
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
4.9"
1.3"
○ Strands discounted to 0% area
▨ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

a) Overlay of Slice No. 13 with the Cross-Section Drawing



```
Slice-Plane No. 13 at a distance along the length = 32.50 ft
--------------------------------------------------------------------------------
Crack cluster no. 19 intersects the slice-plane no. 13at a distance of 32.50 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 19 = 2.81 ft

Crack cluster no. 20 intersects the slice-plane no. 13at a distance of 32.50 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 20 = 1.78 ft

Crack Cluster no. 21 lies close to the slice-plane no. 13
Nearest distance to a boundary of the cluster no. 21 = 4.11 ft

Crack Cluster no. 22 lies close to the slice-plane no. 13
Nearest distance to a boundary of the cluster no. 22 = 4.63 ft

Crack Cluster no. 23 lies close to the slice-plane no. 13
Nearest distance to a boundary of the cluster no. 23 = 3.32 ft

Crack Cluster no. 24 lies close to the slice-plane no. 13
Nearest distance to a boundary of the cluster no. 24 = 2.41 ft

Crack Cluster no. 25 lies close to the slice-plane no. 13
Nearest distance to a boundary of the cluster no. 25 = 1.56 ft

Crack Cluster no. 26 lies close to the slice-plane no. 13
Nearest distance to a boundary of the cluster no. 26 = 3.67 ft

Spall cluster no. 0 intersects the slice plane no. 13at a distance of 32.50 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 0 = 21.87 ft

Spall Cluster no. 1 lies close to the slice plane no. 13
Nearest distance to a boundary of the cluster no. 1 = 2.27 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 13

Figure A.27. Tools for Strength Estimation at Slice-13 on Aden Road Bridge Box Beam-3

Table A.15. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 13

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C20, C24	Side - a	15, 25	Unexposed	75%
C19	Side - a	13,14, 24	Unexposed	75%
C25	Middle	6, 7	Unexposed	75%
C21, C26	Side - b	1, 16	Exposed	0%
S0	Side - b	1, 2, 3, 4, 5, 16		
C22, C23, S1	Noise	-	-	-

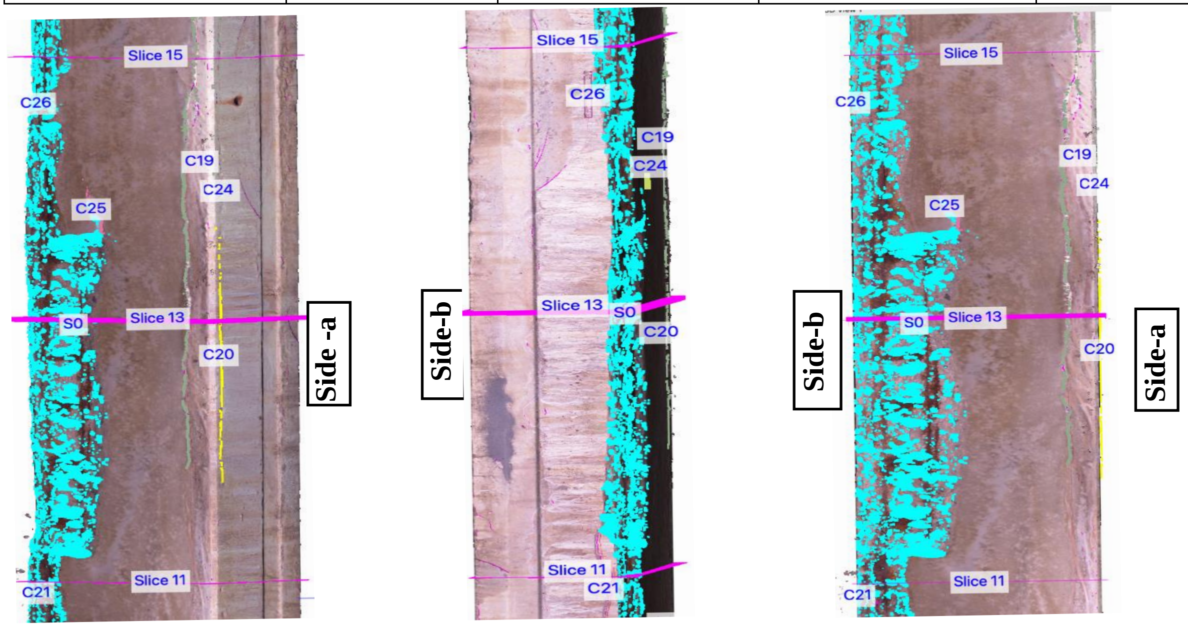
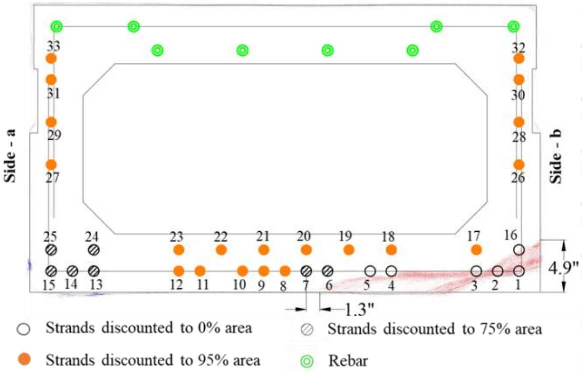
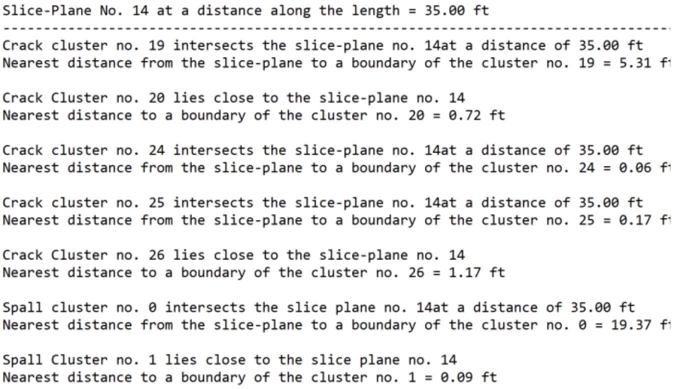


Figure A.28. Damage Clusters in IW considered for Strength Estimation at Slice-13 of Box Beam-3

Slice No. 14 (At y = 35 ft from the left end)


Side - a
Side - b
33
31
29
27
32
30
28
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
4.9"
1.3"
○ Strands discounted to 0% area
▨ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

a) Overlay of Slice No. 14 with the Cross-Section Drawing



```
Slice-Plane No. 14 at a distance along the length = 35.00 ft
------------------------------------------------------------------------------------------
Crack cluster no. 19 intersects the slice-plane no. 14at a distance of 35.00 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 19 = 5.31 f

Crack Cluster no. 20 lies close to the slice-plane no. 14
Nearest distance to a boundary of the cluster no. 20 = 0.72 ft

Crack cluster no. 24 intersects the slice-plane no. 14at a distance of 35.00 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 24 = 0.06 f

Crack cluster no. 25 intersects the slice-plane no. 14at a distance of 35.00 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 25 = 0.17 f

Crack Cluster no. 26 lies close to the slice-plane no. 14
Nearest distance to a boundary of the cluster no. 26 = 1.17 ft

Spall cluster no. 0 intersects the slice plane no. 14at a distance of 35.00 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 0 = 19.37 f

Spall Cluster no. 1 lies close to the slice plane no. 14
Nearest distance to a boundary of the cluster no. 1 = 0.09 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 14

Figure A.29. Tools for Strength Estimation at Slice-14 on Aden Road Bridge Box Beam-3

Table A.16. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 14

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C20, C24	Side - a	15, 25	Unexposed	75%
C19	Side - a	13,14, 24	Unexposed	75%
C25	Middle	6, 7	Unexposed	75%
C26	Side - b	1, 16	Exposed	0%
S0	Side - b	1, 2, 3, 4, 5, 16		
S1	Noise	-	-	-

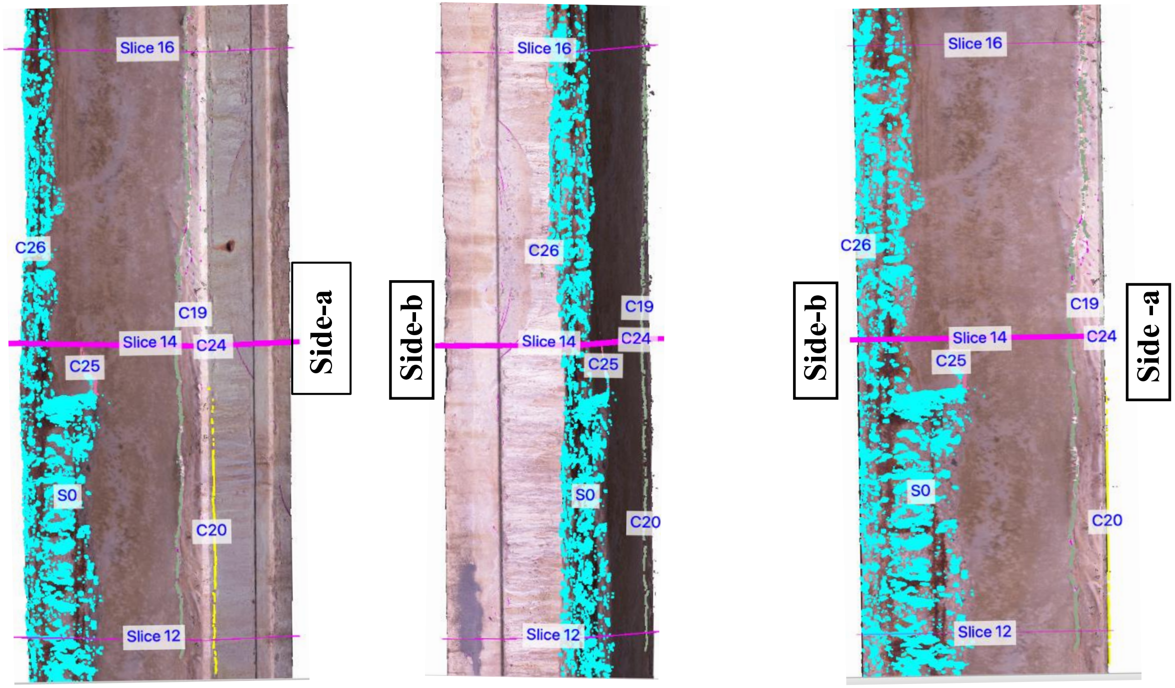
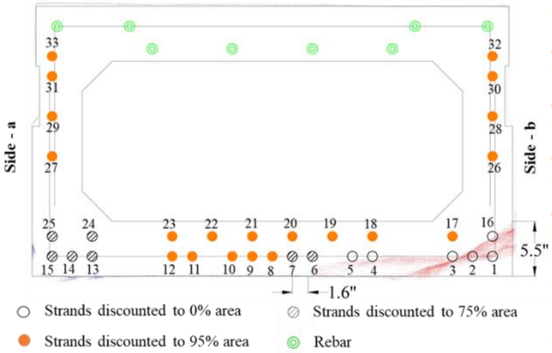
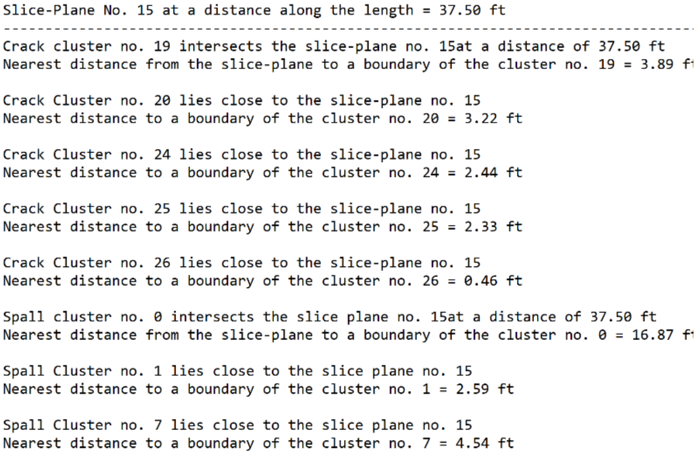


Figure A.30. Damage Clusters in IW considered for Strength Estimation at Slice-14 of Box Beam-3

Slice No. 15 (At y = 37.5 ft from the left end)



a) Overlay of Slice No. 15 with the Cross-Section Drawing



```
Slice-Plane No. 15 at a distance along the length = 37.50 ft
--------------------------------------------------------------------------------
Crack cluster no. 19 intersects the slice-plane no. 15at a distance of 37.50 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 19 = 3.89 f

Crack Cluster no. 20 lies close to the slice-plane no. 15
Nearest distance to a boundary of the cluster no. 20 = 3.22 ft

Crack Cluster no. 24 lies close to the slice-plane no. 15
Nearest distance to a boundary of the cluster no. 24 = 2.44 ft

Crack Cluster no. 25 lies close to the slice-plane no. 15
Nearest distance to a boundary of the cluster no. 25 = 2.33 ft

Crack Cluster no. 26 lies close to the slice-plane no. 15
Nearest distance to a boundary of the cluster no. 26 = 0.46 ft

Spall cluster no. 0 intersects the slice plane no. 15at a distance of 37.50 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 0 = 16.87 f

Spall Cluster no. 1 lies close to the slice plane no. 15
Nearest distance to a boundary of the cluster no. 1 = 2.59 ft

Spall Cluster no. 7 lies close to the slice plane no. 15
Nearest distance to a boundary of the cluster no. 7 = 4.54 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 15

Figure A.31. Tools for Strength Estimation at Slice-15 on Aden Road Bridge Box Beam-3

Table A.17. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 15

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C20, C24	Side - a	15, 25	Unexposed	75%
C19	Side - a	13,14, 24, 15, 25	Unexposed	75%
C25	Middle	6, 7	Unexposed	75%
C26	Side - b	1, 16	Exposed	0%
S0	Side - b	1, 2, 3, 4, 5, 16		
S1, S7	Noise	-	-	-

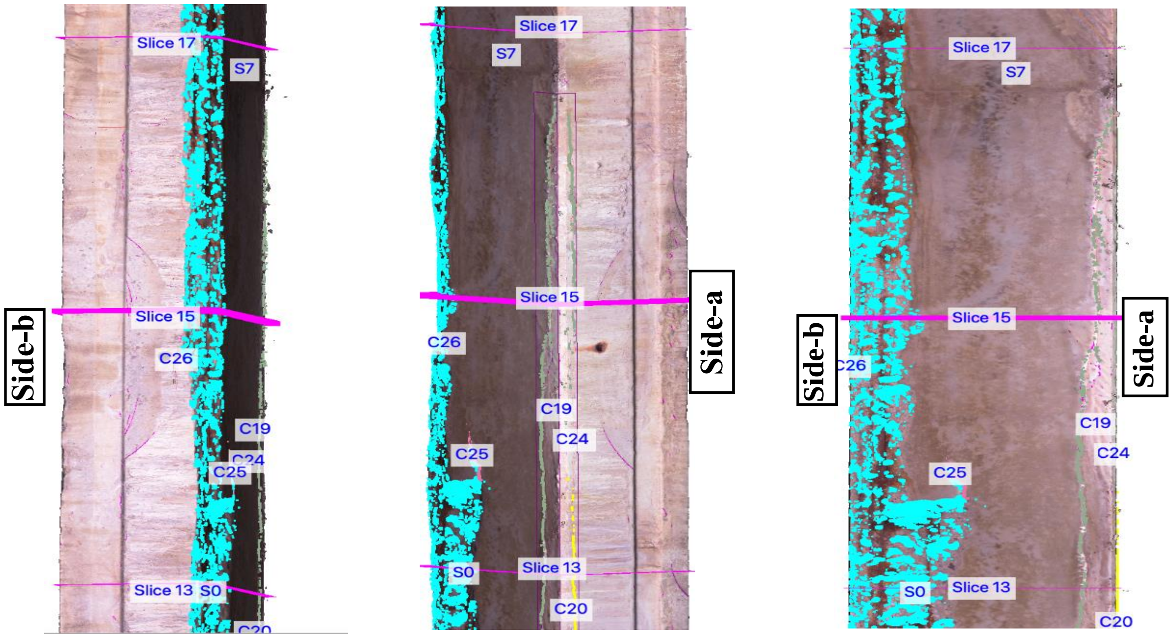
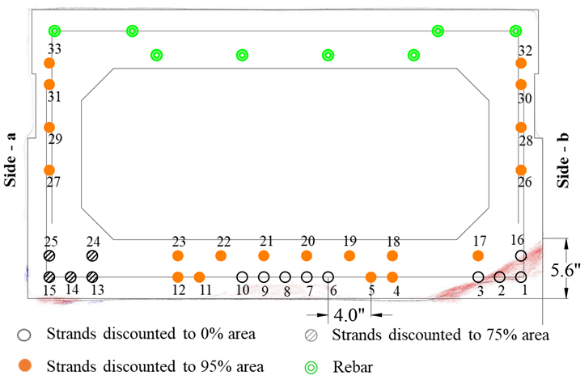
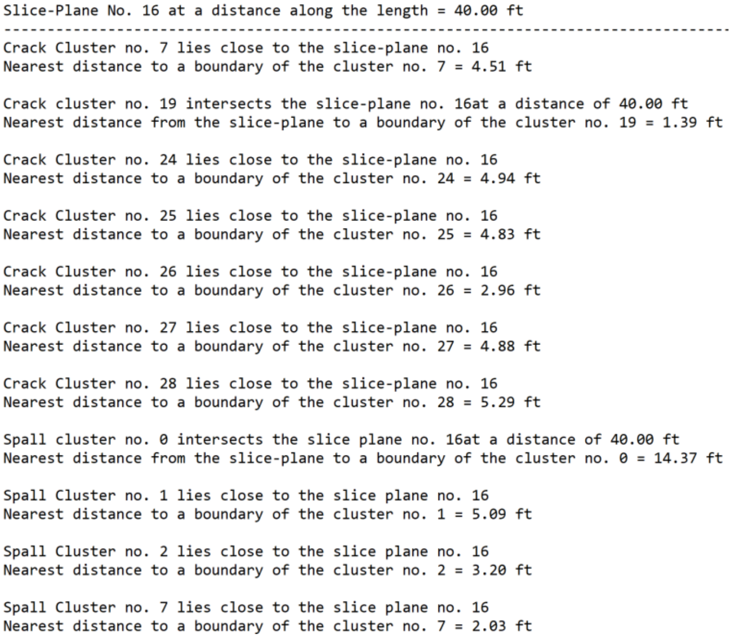


Figure A.32. Damage Clusters in IW considered for Strength Estimation at Slice-15 of Box Beam-3

Slice No. 16 (At y = 40 ft from the left end)


Side - a
Side - b
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
5.6"
4.0"
○ Strands discounted to 0% area
▨ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

a) Overlay of Slice No. 16 with the Cross-Section Drawing



```
Slice-Plane No. 16 at a distance along the length = 40.00 ft
------------------------------------------------------------------------------------
Crack Cluster no. 7 lies close to the slice-plane no. 16
Nearest distance to a boundary of the cluster no. 7 = 4.51 ft

Crack cluster no. 19 intersects the slice-plane no. 16at a distance of 40.00 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 19 = 1.39 ft

Crack Cluster no. 24 lies close to the slice-plane no. 16
Nearest distance to a boundary of the cluster no. 24 = 4.94 ft

Crack Cluster no. 25 lies close to the slice-plane no. 16
Nearest distance to a boundary of the cluster no. 25 = 4.83 ft

Crack Cluster no. 26 lies close to the slice-plane no. 16
Nearest distance to a boundary of the cluster no. 26 = 2.96 ft

Crack Cluster no. 27 lies close to the slice-plane no. 16
Nearest distance to a boundary of the cluster no. 27 = 4.88 ft

Crack Cluster no. 28 lies close to the slice-plane no. 16
Nearest distance to a boundary of the cluster no. 28 = 5.29 ft

Spall cluster no. 0 intersects the slice plane no. 16at a distance of 40.00 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 0 = 14.37 ft

Spall Cluster no. 1 lies close to the slice plane no. 16
Nearest distance to a boundary of the cluster no. 1 = 5.09 ft

Spall Cluster no. 2 lies close to the slice plane no. 16
Nearest distance to a boundary of the cluster no. 2 = 3.20 ft

Spall Cluster no. 7 lies close to the slice plane no. 16
Nearest distance to a boundary of the cluster no. 7 = 2.03 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 16

Figure A.33. Tools for Strength Estimation at Slice-16 on Aden Road Bridge Box Beam-3

Table A.18. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 16

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C7, C19, C24	Side - a	13,14, 24, 15, 25	Unexposed	75%
C27	Middle + Side-a	13, 24	Unexposed	
C26, C28	Side - b	1, 16	Exposed	0%
S0	Side - b	1, 2, 3,16		
C25	Middle	6, 7	Exposed (Strand 10 is slightly exposed)	0%
S2	Middle	6, 7, 8, 9, 10		
S1, S7	Noise	-	-	-

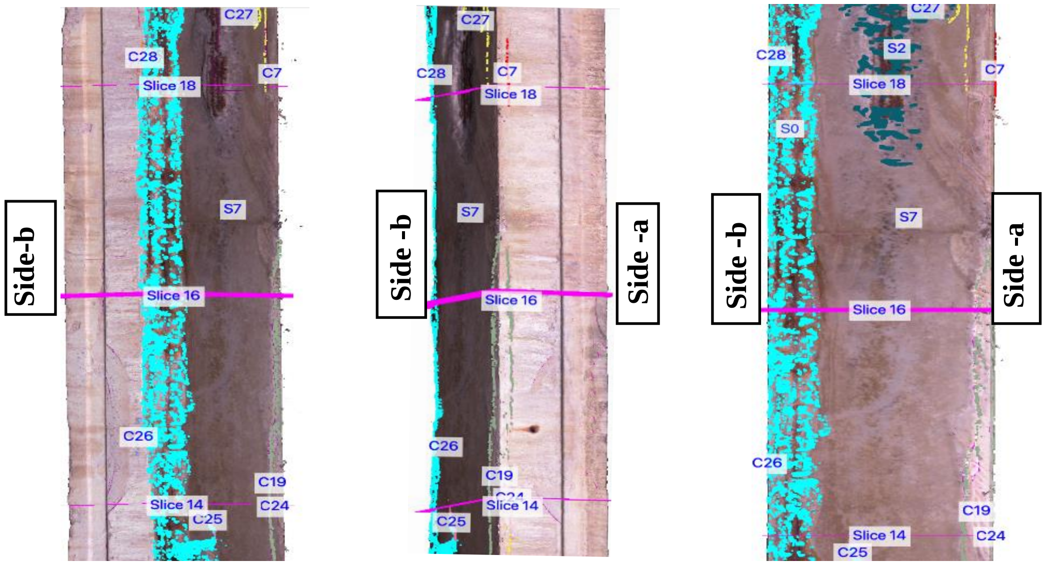
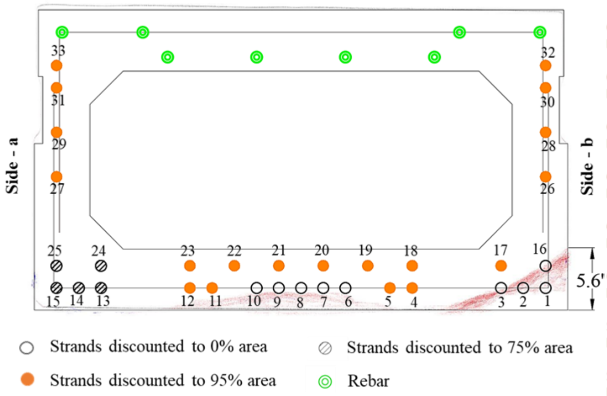


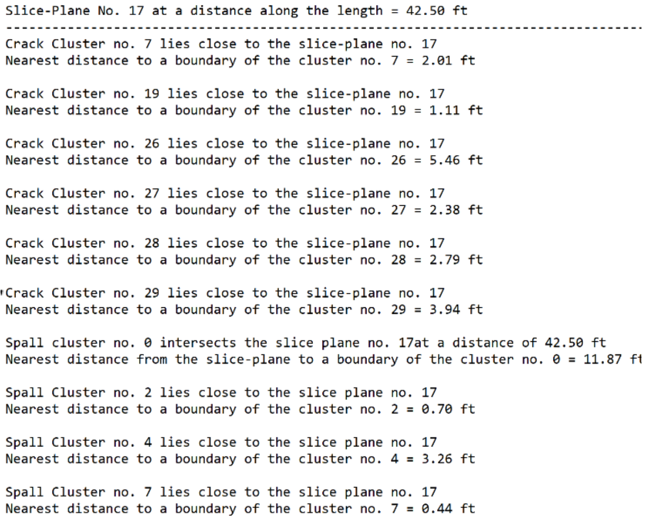
Figure A.34. Damage Clusters in IW considered for Strength Estimation at Slice-16 of Box Beam-3

Slice No. 17 (At y = 42.5 ft from the left end)



Side - a
Side - b
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15 14 13
12 11
10 9 8 7 6
5 4
3 2 1
5.6'
○ Strands discounted to 0% area
▨ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

a) Overlay of Slice No. 17 with the Cross-Section Drawing



```
Slice-Plane No. 17 at a distance along the length = 42.50 ft
--------------------------------------------------------------------------------
Crack Cluster no. 7 lies close to the slice-plane no. 17
Nearest distance to a boundary of the cluster no. 7 = 2.01 ft

Crack Cluster no. 19 lies close to the slice-plane no. 17
Nearest distance to a boundary of the cluster no. 19 = 1.11 ft

Crack Cluster no. 26 lies close to the slice-plane no. 17
Nearest distance to a boundary of the cluster no. 26 = 5.46 ft

Crack Cluster no. 27 lies close to the slice-plane no. 17
Nearest distance to a boundary of the cluster no. 27 = 2.38 ft

Crack Cluster no. 28 lies close to the slice-plane no. 17
Nearest distance to a boundary of the cluster no. 28 = 2.79 ft

Crack Cluster no. 29 lies close to the slice-plane no. 17
Nearest distance to a boundary of the cluster no. 29 = 3.94 ft

Spall cluster no. 0 intersects the slice plane no. 17at a distance of 42.50 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 0 = 11.87 ft

Spall Cluster no. 2 lies close to the slice plane no. 17
Nearest distance to a boundary of the cluster no. 2 = 0.70 ft

Spall Cluster no. 4 lies close to the slice plane no. 17
Nearest distance to a boundary of the cluster no. 4 = 3.26 ft

Spall Cluster no. 7 lies close to the slice plane no. 17
Nearest distance to a boundary of the cluster no. 7 = 0.44 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 17

Figure A.35. Tools for Strength Estimation at Slice-17 on Aden Road Bridge Box Beam-3

Table A.19. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 17

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C7, C19	Side - a	13,14, 24, 15, 25	Unexposed	75%
C27	Middle + Side -a	13, 24	Unexposed	
C26, C28, C29	Side - b	1, 16	Exposed	0%
S0	Side - b	1, 2, 3,16		
S2	Middle	6, 7, 8, 9, 10	Exposed (Strand 10 is slightly exposed)	0%
S7	Noise	-	-	-
S4	Actually, a part of Spall Cluster 2			

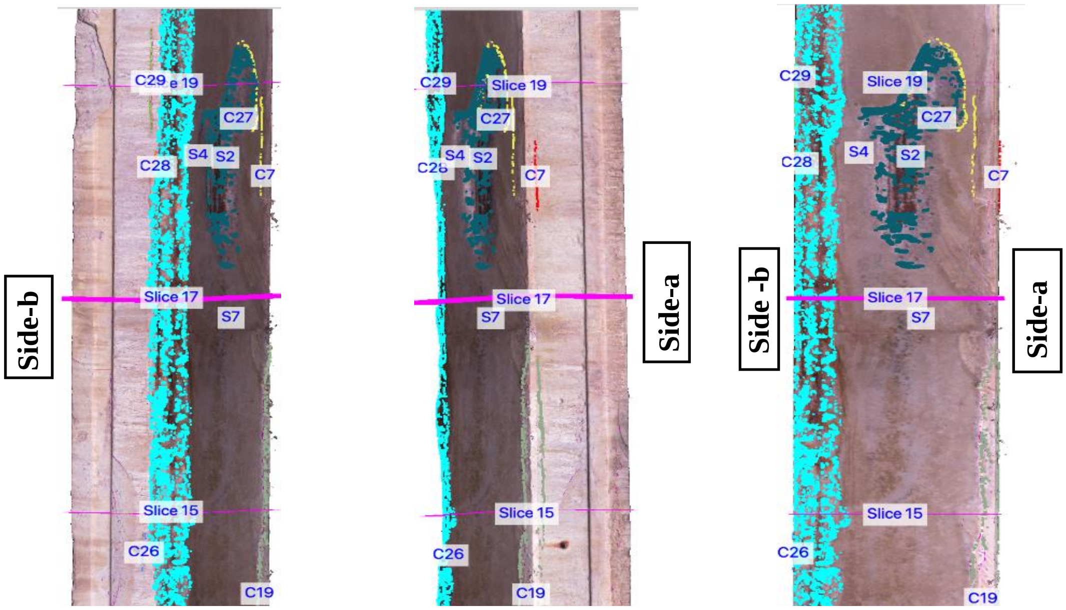
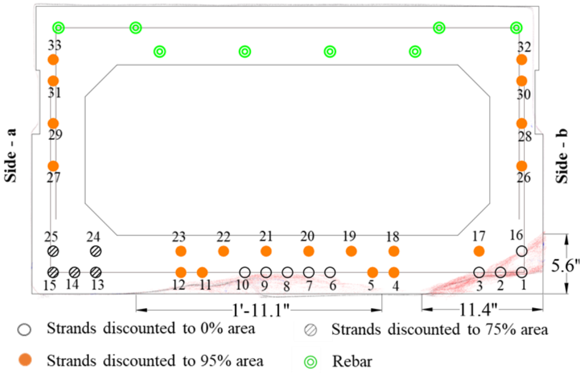
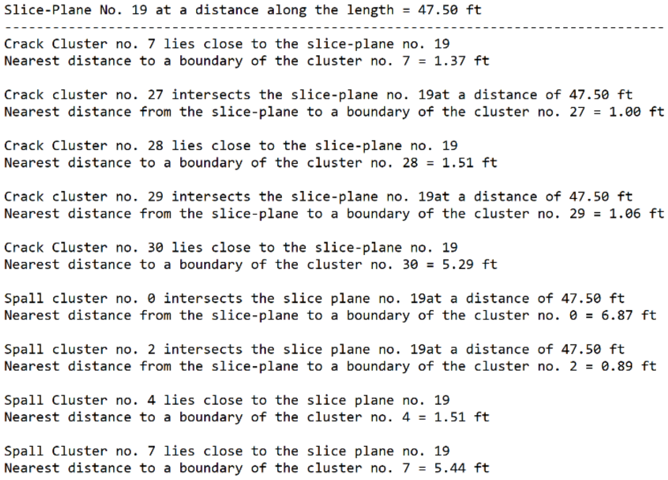


Figure A.36. Damage Clusters in IW considered for Strength Estimation at Slice-17 of Box Beam-3

Slice No. 19 (At y = 47.5 ft from the left end)



a) Overlay of Slice No. 19 with the Cross-Section Drawing



```
Slice-Plane No. 19 at a distance along the length = 47.50 ft
------------------------------------------------------------------------------
Crack Cluster no. 7 lies close to the slice-plane no. 19
Nearest distance to a boundary of the cluster no. 7 = 1.37 ft

Crack cluster no. 27 intersects the slice-plane no. 19at a distance of 47.50 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 27 = 1.00 ft

Crack Cluster no. 28 lies close to the slice-plane no. 19
Nearest distance to a boundary of the cluster no. 28 = 1.51 ft

Crack cluster no. 29 intersects the slice-plane no. 19at a distance of 47.50 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 29 = 1.06 ft

Crack Cluster no. 30 lies close to the slice-plane no. 19
Nearest distance to a boundary of the cluster no. 30 = 5.29 ft

Spall cluster no. 0 intersects the slice plane no. 19at a distance of 47.50 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 0 = 6.87 ft

Spall cluster no. 2 intersects the slice plane no. 19at a distance of 47.50 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 2 = 0.89 ft

Spall Cluster no. 4 lies close to the slice plane no. 19
Nearest distance to a boundary of the cluster no. 4 = 1.51 ft

Spall Cluster no. 7 lies close to the slice plane no. 19
Nearest distance to a boundary of the cluster no. 7 = 5.44 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 19

Figure A.37. Tools for Strength Estimation at Slice-19 on Aden Road Bridge Box Beam-3

Table A.20. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 19

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C7	Side - a	15, 25	Unexposed	75%
C27	Middle + Side -a	13, 14, 24	Unexposed	
C28, C29	Side - b	1, 16	Exposed	0%
S0	Side - b	1, 2, 3,16		
S2	Middle	6, 7, 8, 9, 10	Exposed (Strand 10 is slightly exposed)	0%
S7, C30	Noise	-	-	-
S4	Actually, a part of Spall Cluster 2			

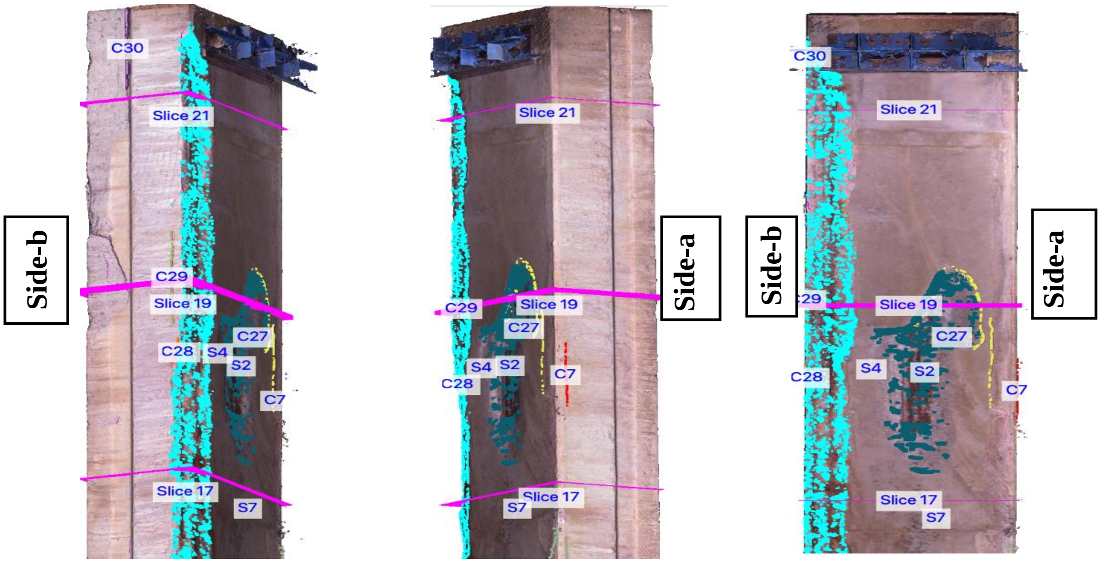
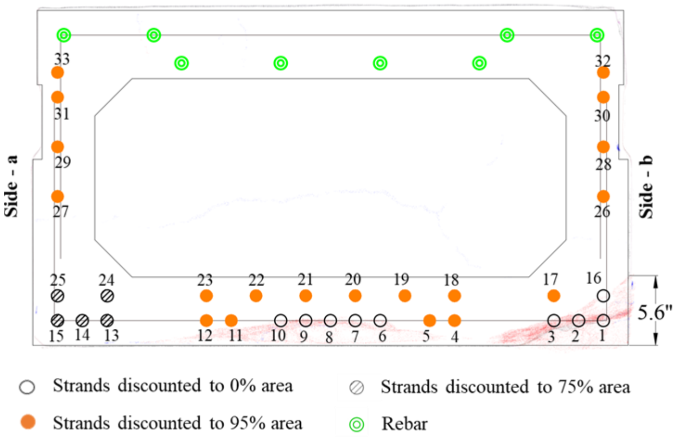
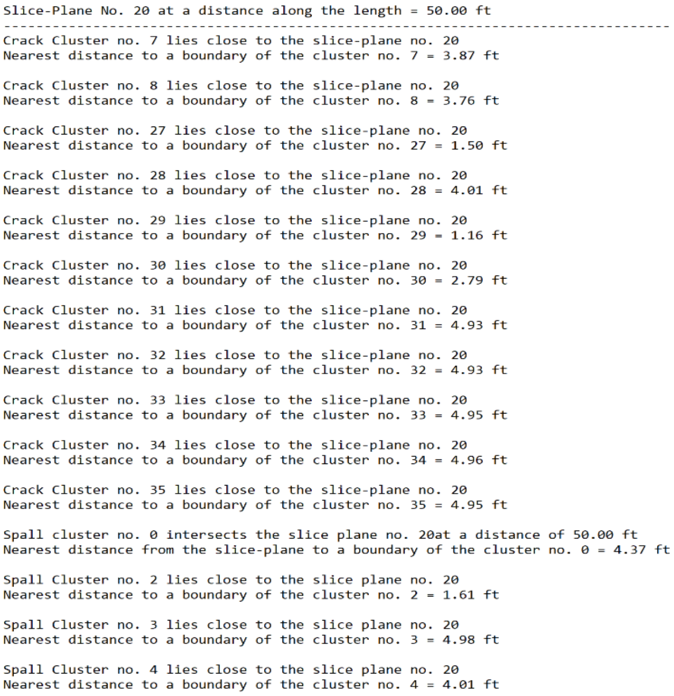


Figure A.38. Damage Clusters in IW considered for Strength Estimation at Slice-19 of Box Beam-3

Slice No. 20 (At y = 50 ft from the left end)


Side - a
Side - b
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
5.6"
○ Strands discounted to 0% area
▨ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

a) Overlay of Slice No. 20 with the Cross-Section Drawing



```
Slice-Plane No. 20 at a distance along the length = 50.00 ft
--------------------------------------------------------------------------------
Crack Cluster no. 7 lies close to the slice-plane no. 20
Nearest distance to a boundary of the cluster no. 7 = 3.87 ft

Crack Cluster no. 8 lies close to the slice-plane no. 20
Nearest distance to a boundary of the cluster no. 8 = 3.76 ft

Crack Cluster no. 27 lies close to the slice-plane no. 20
Nearest distance to a boundary of the cluster no. 27 = 1.50 ft

Crack Cluster no. 28 lies close to the slice-plane no. 20
Nearest distance to a boundary of the cluster no. 28 = 4.01 ft

Crack Cluster no. 29 lies close to the slice-plane no. 20
Nearest distance to a boundary of the cluster no. 29 = 1.16 ft

Crack Cluster no. 30 lies close to the slice-plane no. 20
Nearest distance to a boundary of the cluster no. 30 = 2.79 ft

Crack Cluster no. 31 lies close to the slice-plane no. 20
Nearest distance to a boundary of the cluster no. 31 = 4.93 ft

Crack Cluster no. 32 lies close to the slice-plane no. 20
Nearest distance to a boundary of the cluster no. 32 = 4.93 ft

Crack Cluster no. 33 lies close to the slice-plane no. 20
Nearest distance to a boundary of the cluster no. 33 = 4.95 ft

Crack Cluster no. 34 lies close to the slice-plane no. 20
Nearest distance to a boundary of the cluster no. 34 = 4.96 ft

Crack Cluster no. 35 lies close to the slice-plane no. 20
Nearest distance to a boundary of the cluster no. 35 = 4.95 ft

Spall cluster no. 0 intersects the slice plane no. 20at a distance of 50.00 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 0 = 4.37 ft

Spall Cluster no. 2 lies close to the slice plane no. 20
Nearest distance to a boundary of the cluster no. 2 = 1.61 ft

Spall Cluster no. 3 lies close to the slice plane no. 20
Nearest distance to a boundary of the cluster no. 3 = 4.98 ft

Spall Cluster no. 4 lies close to the slice plane no. 20
Nearest distance to a boundary of the cluster no. 4 = 4.01 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 20

Figure A.39. Tools for Strength Estimation at Slice-20 on Aden Road Bridge Box Beam-3

Table A.21. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 20

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C7	Side - a	15, 25	Unexposed	75%
C27	Middle + Side -a	13, 14, 24	Unexposed	
C8, C28, C29	Side - b	1, 16	Exposed	0%
S0	Side - b	1, 2, 3,16		
S2	Middle	6, 7, 8, 9, 10	Exposed (Strand 10 is slightly exposed)	0%
S7, C30, S3	Noise	-	-	-
S4	A part of Spall Cluster 2			
C31, C32, C33, C34, C35	End of the beam	Assumed that the cracks at the end do not affect the residual capacity		

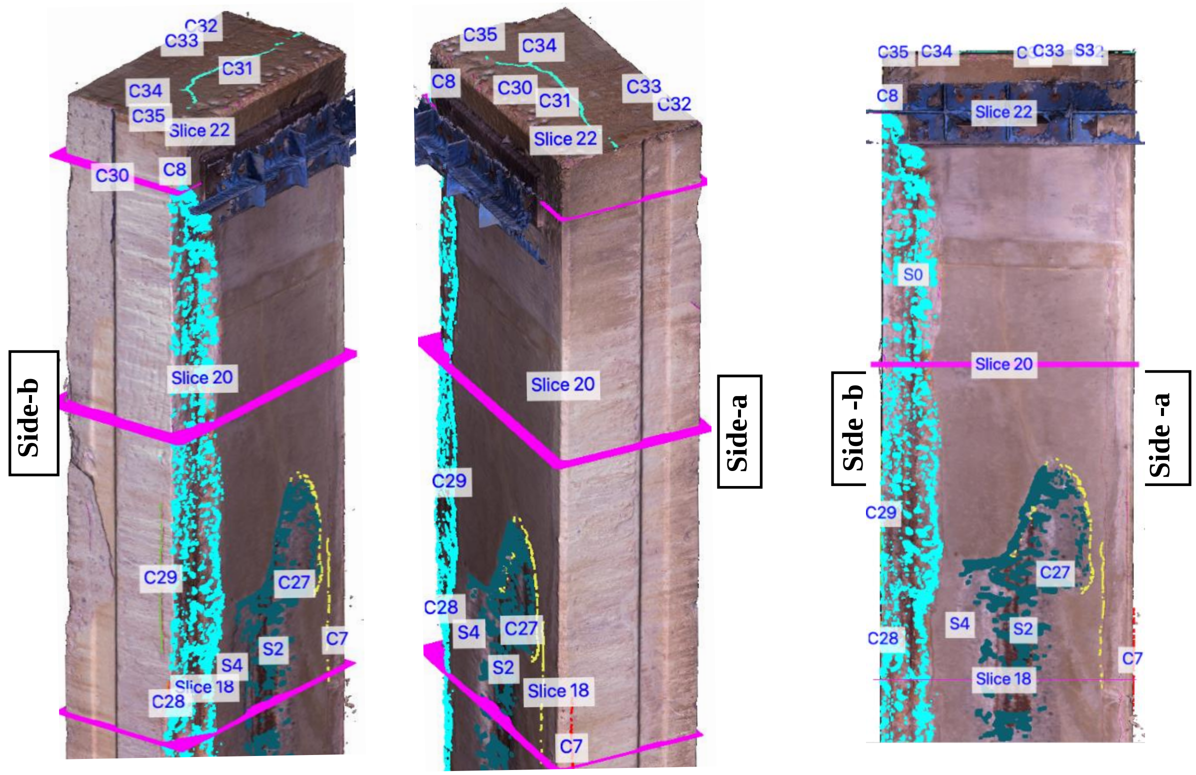
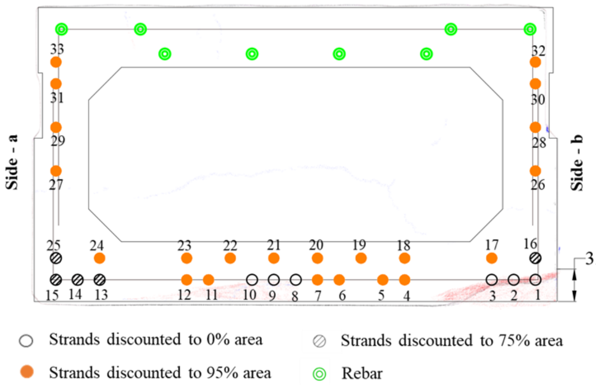


Figure A.40. Damage Clusters in IW considered for Strength Estimation at Slice-20 of Box Beam-3

Slice No. 21 (At y = 52.5 ft from the left end)



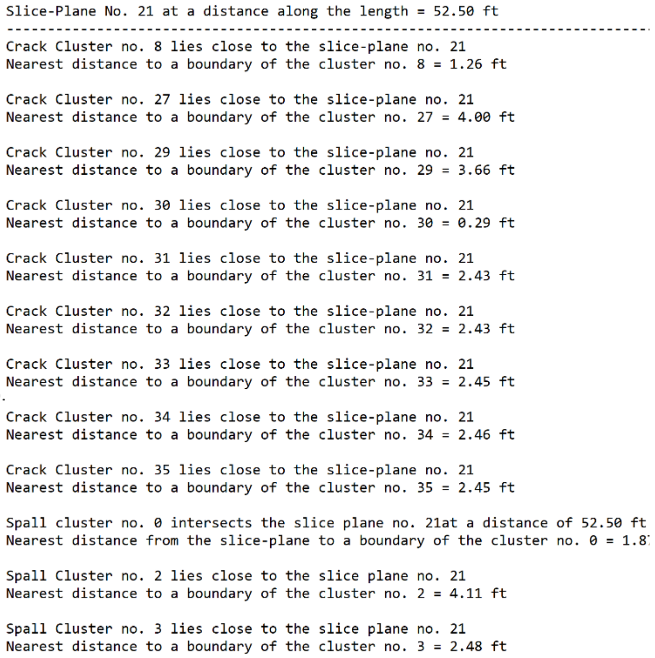
○ Strands discounted to 0% area

⊘ Strands discounted to 75% area

● Strands discounted to 95% area

◎ Rebar

a) Overlay of Slice No. 21 with the Cross-Section Drawing



```
Slice-Plane No. 21 at a distance along the length = 52.50 ft
---------------------------------------------------------------------------
Crack Cluster no. 8 lies close to the slice-plane no. 21
Nearest distance to a boundary of the cluster no. 8 = 1.26 ft

Crack Cluster no. 27 lies close to the slice-plane no. 21
Nearest distance to a boundary of the cluster no. 27 = 4.00 ft

Crack Cluster no. 29 lies close to the slice-plane no. 21
Nearest distance to a boundary of the cluster no. 29 = 3.66 ft

Crack Cluster no. 30 lies close to the slice-plane no. 21
Nearest distance to a boundary of the cluster no. 30 = 0.29 ft

Crack Cluster no. 31 lies close to the slice-plane no. 21
Nearest distance to a boundary of the cluster no. 31 = 2.43 ft

Crack Cluster no. 32 lies close to the slice-plane no. 21
Nearest distance to a boundary of the cluster no. 32 = 2.43 ft

Crack Cluster no. 33 lies close to the slice-plane no. 21
Nearest distance to a boundary of the cluster no. 33 = 2.45 ft

Crack Cluster no. 34 lies close to the slice-plane no. 21
Nearest distance to a boundary of the cluster no. 34 = 2.46 ft

Crack Cluster no. 35 lies close to the slice-plane no. 21
Nearest distance to a boundary of the cluster no. 35 = 2.45 ft

Spall cluster no. 0 intersects the slice plane no. 21at a distance of 52.50 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 0 = 1.8

Spall Cluster no. 2 lies close to the slice plane no. 21
Nearest distance to a boundary of the cluster no. 2 = 4.11 ft

Spall Cluster no. 3 lies close to the slice plane no. 21
Nearest distance to a boundary of the cluster no. 3 = 2.48 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 21

Figure A.41. Tools for Strength Estimation at Slice-21 on Aden Road Bridge Box Beam-3

Table A.22. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 21

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C27, Undetected cracks	Middle + Side -a	13, 14,15, 25	Unexposed	75%
C8, C29	Side - b	16	Unexposed	75%
S0	Side - b	1, 2, 3	Exposed	0%
S2	Middle	8, 9, 10	Exposed (Strand 10 is slightly exposed)	0%
C30, S3	Noise	-	-	-
C31, C32, C33, C34, C35	End of the beam	It is assumed that cracks at the end do not affect the residual capacity		

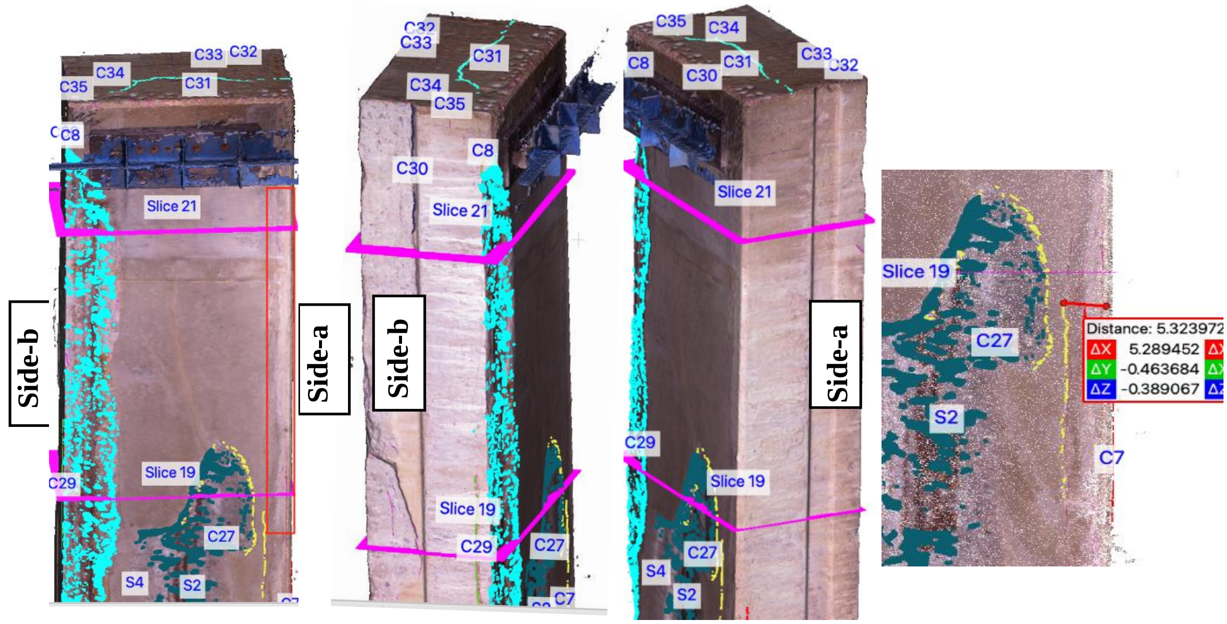
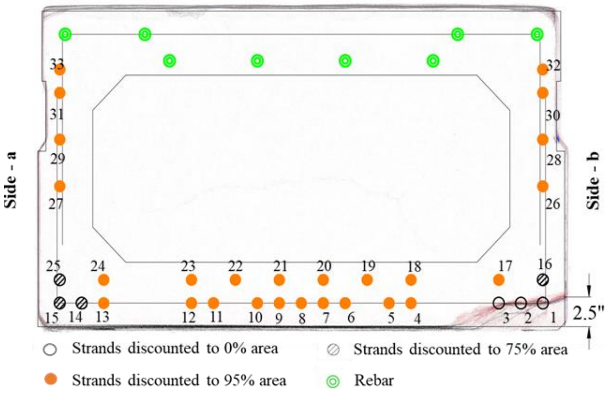
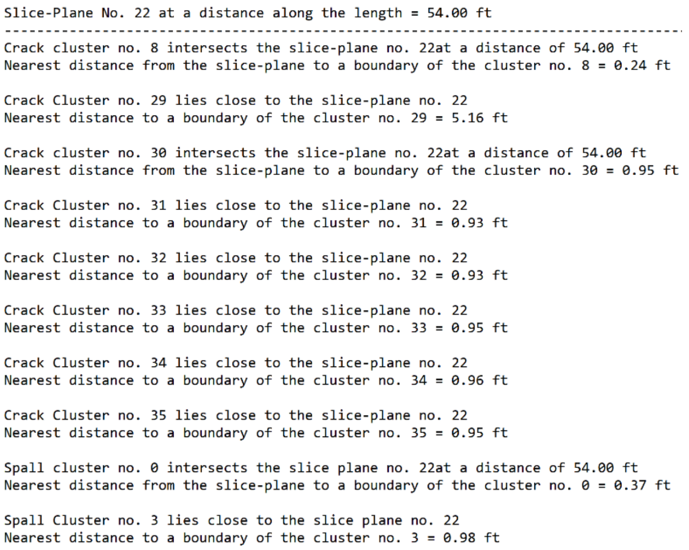


Figure A.42. Damage Clusters in IW considered for Strength Estimation at Slice-21 of Box Beam-3

Slice No. 22 (At y = 54 ft from the left end)


Side - a
Side - b
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
2.5"
○ Strands discounted to 0% area
▨ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

a) Overlay of Slice No. 22 with the Cross-Section Drawing


```
Slice-Plane No. 22 at a distance along the length = 54.00 ft
-----------------------------------------------------------------------------------
Crack cluster no. 8 intersects the slice-plane no. 22at a distance of 54.00 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 8 = 0.24 ft

Crack Cluster no. 29 lies close to the slice-plane no. 22
Nearest distance to a boundary of the cluster no. 29 = 5.16 ft

Crack cluster no. 30 intersects the slice-plane no. 22at a distance of 54.00 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 30 = 0.95 ft

Crack Cluster no. 31 lies close to the slice-plane no. 22
Nearest distance to a boundary of the cluster no. 31 = 0.93 ft

Crack Cluster no. 32 lies close to the slice-plane no. 22
Nearest distance to a boundary of the cluster no. 32 = 0.93 ft

Crack Cluster no. 33 lies close to the slice-plane no. 22
Nearest distance to a boundary of the cluster no. 33 = 0.95 ft

Crack Cluster no. 34 lies close to the slice-plane no. 22
Nearest distance to a boundary of the cluster no. 34 = 0.96 ft

Crack Cluster no. 35 lies close to the slice-plane no. 22
Nearest distance to a boundary of the cluster no. 35 = 0.95 ft

Spall cluster no. 0 intersects the slice plane no. 22at a distance of 54.00 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 0 = 0.37 ft

Spall Cluster no. 3 lies close to the slice plane no. 22
Nearest distance to a boundary of the cluster no. 3 = 0.98 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 22

Figure A.43. Tools for Strength Estimation at Slice-22 on Aden Road Bridge Box Beam-3

Table A.23. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 22

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
Undetected cracking	Middle + Side -a	14,15, 25	Unexposed	75%
C8, C29	Side - b	16	Unexposed	75%
S0	Side - b	1, 2, 3	Exposed	0%
C30, S3	Noise	-	-	-
C31, C32, C33, C34, C35	End of the beam	It is assumed that cracks at the end do not affect the residual capacity		

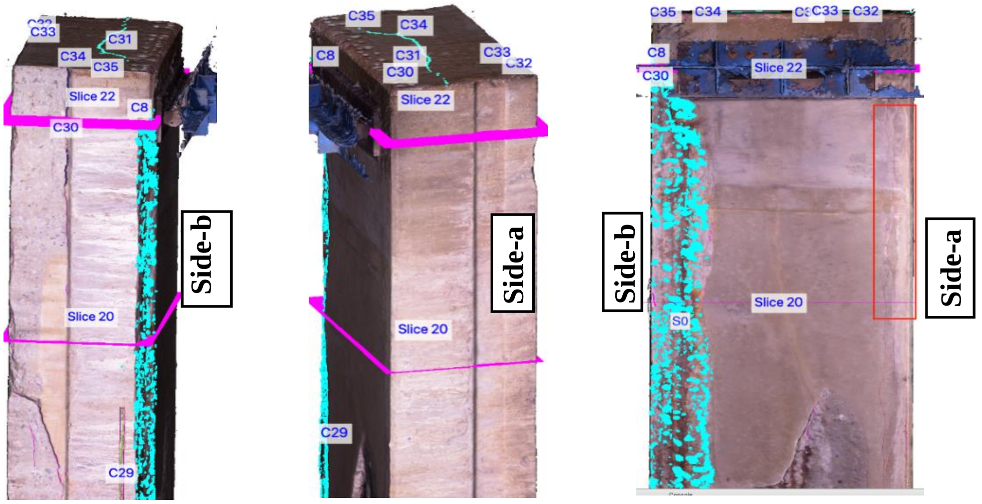


Figure A.44. Damage Clusters in IW considered for Strength Estimation at Slice-22 of Box Beam-3

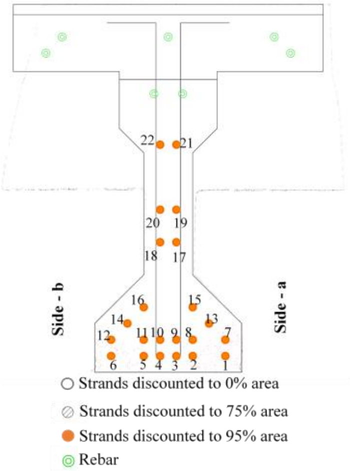
APPENDIX – B

ESTIMATION OF RESIDUAL CAPACITY OF I-BEAM 9 AT VARIOUS SLICE-SECTIONS

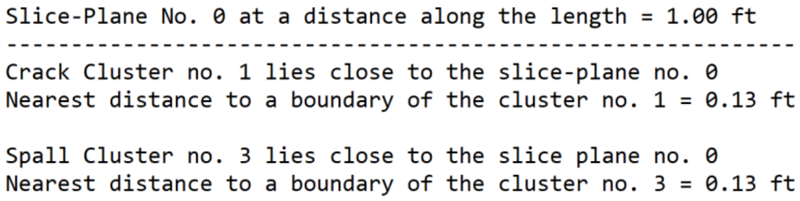
This Appendix deals with the determination of corrosion damage inside the Inspection Window for each slice-section of I-Beam 9 to arrive at a reduced area of cross-section to be used for strength calculation. As explained in Section 3.1.3.2, slices were extracted at 10 slice-sections along the length of the beam. The first and the last slices were extracted at a distance of 1 ft from the corresponding ends of the beam. The spacing of intermediate slices was chosen to be 5 ft. In the following sub-sections, the overlay of each slice on top of the design cross-section drawing is shown along with automatically generated the text file output that shows the list of damage clusters to be considered to calculate strength at each slice-section. These constitute the tools for strength estimation at the slice-section. These damage clusters are shown superimposed on a portion of the textured mesh of the beam which lies inside the Inspection Window (IW) of size equal to two development lengths of the bottommost layer strands. Strands affected due to each damage cluster are shown for each slice-section. At some slice-sections, there were some undetected cracks, especially on the bottom surface which affect the strength of the beam. These cracks were still visible on the textured mesh and were accounted for in strength calculation. These undetected cracks, if any, are highlighted using bounding boxes on the textured mesh shown for each slice-section.

The distance along the length of the beam at which each slice-section was extracted is shown and the tools for strength estimation as explained above are presented in the following sub-sections of the Appendix. Inspection Window for each slice-section is represented as IW. Crack clusters are designated as ‘C’, followed by the identification number of the cracks (for example, C9 means Crack Cluster No. 9). Similarly, spall clusters are represented using the letter ‘S’, followed by a number. Side-a and side-b are marked to show the sides of the beam as represented in Figure 4.2b.

Slice No. 0 (At y = 1 ft from the left end of the beam)


22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
Side - b
Side - a
○ Strands discounted to 0% area
▨ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

a) Overlay of Slice No. 0 with the Cross-Section Drawing



```
Slice-Plane No. 0 at a distance along the length = 1.00 ft
----------------------------------------------------------
Crack Cluster no. 1 lies close to the slice-plane no. 0
Nearest distance to a boundary of the cluster no. 1 = 0.13 ft

Spall Cluster no. 3 lies close to the slice plane no. 0
Nearest distance to a boundary of the cluster no. 3 = 0.13 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 0

Figure B.1. Tools for Strength Estimation at Slice-0 of Lesner Bridge I-Beam 9

Table B. 1. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 0

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C1	Side - b	None. Crack in the end diaphragm	Unexposed	-
S3	At the end of the beam	None. Effect on the strength due to end damage clusters was already accounted for	Unexposed	-
-	-	Other strands unaffected due to cracks		95%

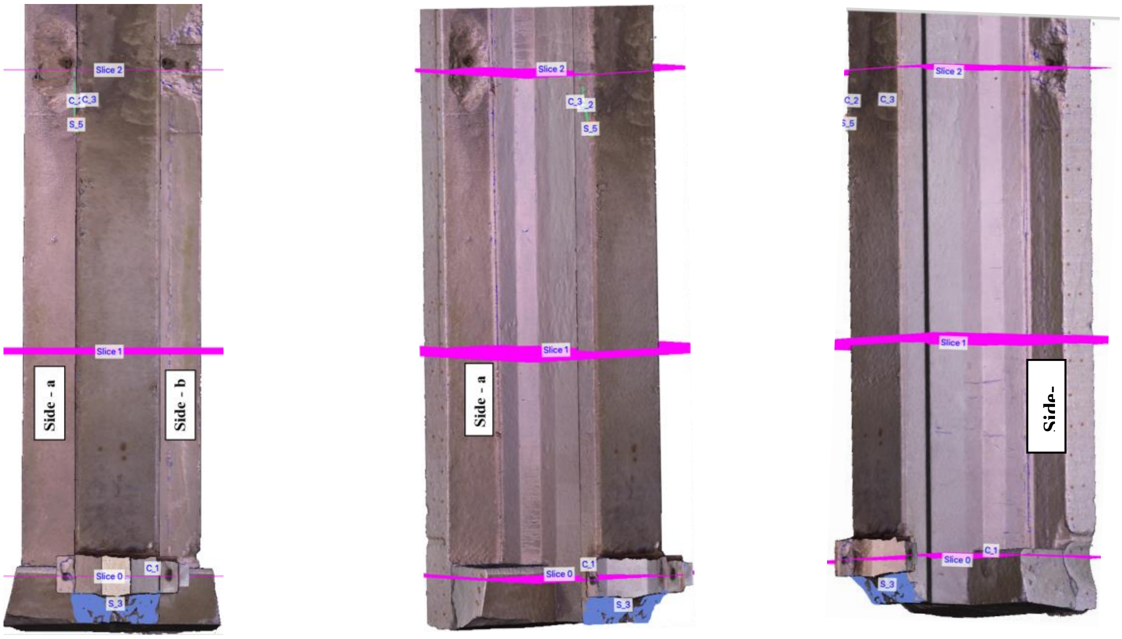
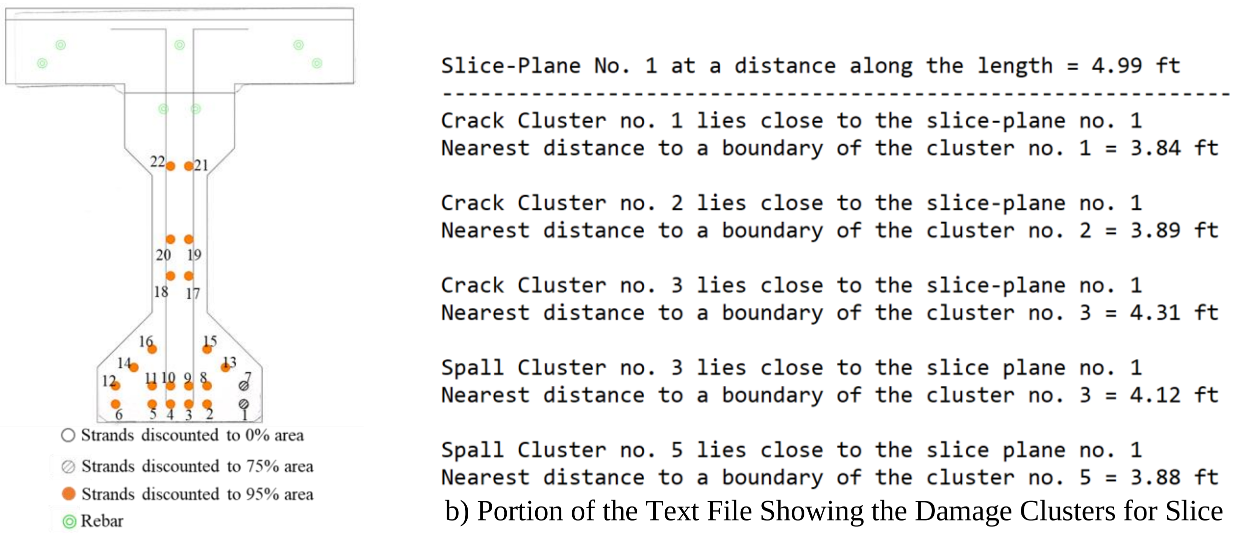


Figure B.2. Damage Clusters in the IW considered for Strength Estimation at Slice-0 and Slice-1

Slice No. 1 (At y = 5 ft from the left end of the beam)


22
21
20
19
18
17
16
15
14
13
12
11 10 9 8
7
6
5 4 3 2
1
○ Strands discounted to 0% area
⊘ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

```
Slice-Plane No. 1 at a distance along the length = 4.99 ft
------------------------------------------------------------
Crack Cluster no. 1 lies close to the slice-plane no. 1
Nearest distance to a boundary of the cluster no. 1 = 3.84 ft

Crack Cluster no. 2 lies close to the slice-plane no. 1
Nearest distance to a boundary of the cluster no. 2 = 3.89 ft

Crack Cluster no. 3 lies close to the slice-plane no. 1
Nearest distance to a boundary of the cluster no. 3 = 4.31 ft

Spall Cluster no. 3 lies close to the slice plane no. 1
Nearest distance to a boundary of the cluster no. 3 = 4.12 ft

Spall Cluster no. 5 lies close to the slice plane no. 1
Nearest distance to a boundary of the cluster no. 5 = 3.88 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 1

a) Overlay of Slice No. 1 with the Cross-Section Drawing

Figure B.3. Tools for Strength Estimation at Slice-1 of Lesner Bridge I-Beam 9

Table B. 2. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 1

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C1	Side - b	None. Crack in the end diaphragm	-	-
C2, C3	Side-a	1, 7	Unexposed	75%
S5	Side-a	None. Too small	-	-
S3	End of the beam	None. End damage clusters was already accounted for	Unexposed	-
Undetected crack at Slice 2	Side-a	1, 7. Strand 2 is unaffected as it is located at 3in. from crack	Unexposed	75%
-	-	Other strands unaffected due to cracks		95%

Damage Clusters in the Inspection Window shown in *Figure B.2* also apply for strength estimation at Slice-1. Undetected crack at Slice 2 is shown below in *Figure B. 4*.

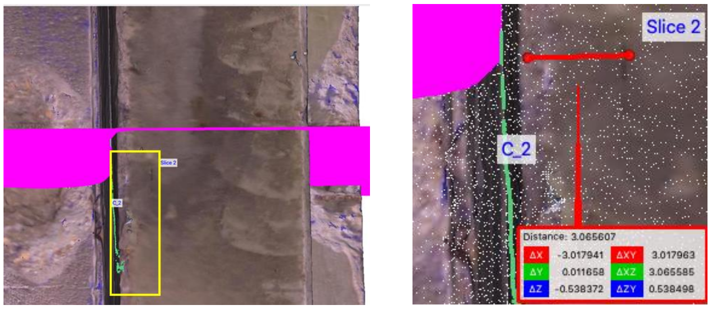
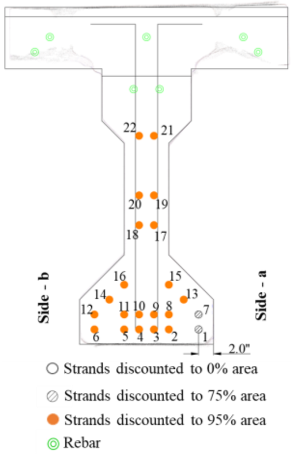
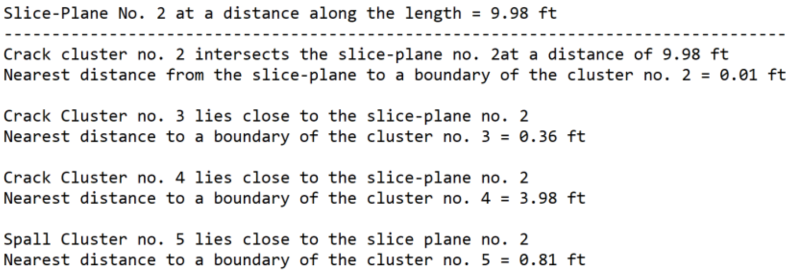


Figure B. 4. Undetected Crack Affecting Strength at Slice-1 and Slice-2

Slice No. 2 (At y = 9.98 ft from the left end of the beam)


22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
2.0"
Side - b
Side - a
○ Strands discounted to 0% area
⊘ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

a) Overlay of Slice No. 2 with the Cross-Section Drawing



```
Slice-Plane No. 2 at a distance along the length = 9.98 ft
----------------------------------------------------------------------------------
Crack cluster no. 2 intersects the slice-plane no. 2at a distance of 9.98 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 2 = 0.01 ft

Crack Cluster no. 3 lies close to the slice-plane no. 2
Nearest distance to a boundary of the cluster no. 3 = 0.36 ft

Crack Cluster no. 4 lies close to the slice-plane no. 2
Nearest distance to a boundary of the cluster no. 4 = 3.98 ft

Spall Cluster no. 5 lies close to the slice plane no. 2
Nearest distance to a boundary of the cluster no. 5 = 0.81 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 2

Figure B. 5. Tools for Strength Estimation at Slice-2 of Lesner Bridge I-Beam 9

Table B. 3. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 2

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C2, C3, C4	Side-a	1, 7	Unexposed	75%
S5	Side-a	None	-	-
Undetected crack near Slice 2	Side-a	1, 7. Strand 2 is unaffected as it is located at 3in. from crack	Unexposed	75%
Undetected crack near Slice 3	Side-a	1,7		
-	-	Other strands unaffected due to cracks		95%

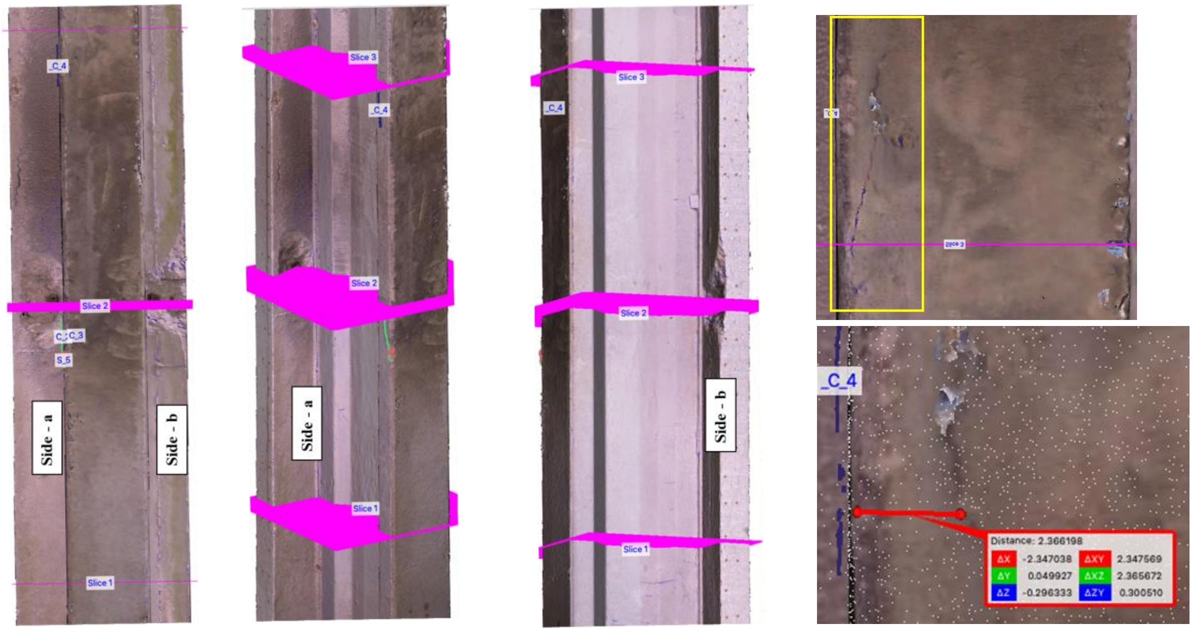
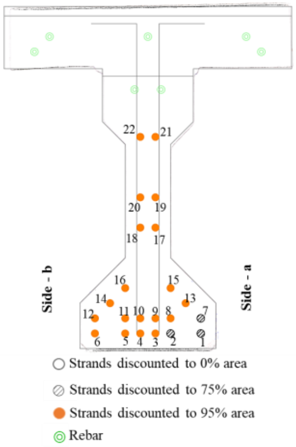
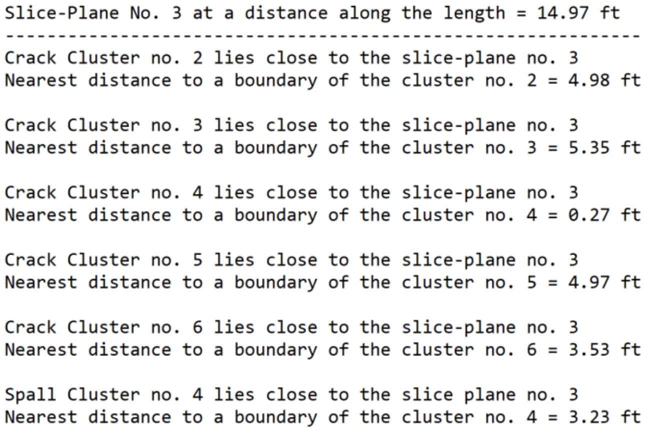


Figure B. 6. Damage Clusters in the IW considered for Strength Estimation at Slice-2

Slice No. 3 (At y = 14.97 ft from the left end of the beam)


22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
Side - b
Side - a
○ Strands discounted to 0% area
⊘ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

a) Overlay of Slice No. 3 with the Cross-Section Drawing



```
Slice-Plane No. 3 at a distance along the length = 14.97 ft
------------------------------------------------------------
Crack Cluster no. 2 lies close to the slice-plane no. 3
Nearest distance to a boundary of the cluster no. 2 = 4.98 ft

Crack Cluster no. 3 lies close to the slice-plane no. 3
Nearest distance to a boundary of the cluster no. 3 = 5.35 ft

Crack Cluster no. 4 lies close to the slice-plane no. 3
Nearest distance to a boundary of the cluster no. 4 = 0.27 ft

Crack Cluster no. 5 lies close to the slice-plane no. 3
Nearest distance to a boundary of the cluster no. 5 = 4.97 ft

Crack Cluster no. 6 lies close to the slice-plane no. 3
Nearest distance to a boundary of the cluster no. 6 = 3.53 ft

Spall Cluster no. 4 lies close to the slice plane no. 3
Nearest distance to a boundary of the cluster no. 4 = 3.23 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No.3

Figure B.7. Tools for Strength Estimation at Slice-3 of Lesner Bridge I-Beam 9

Table B. 4. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 3

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C2, C3, C4, C6	Side-a	1, 7	Unexposed	75%
S4	Side-a	None	-	-
Undetected cracks near Slice 2, Slice-3	Side-a	1, 7	Unexposed	75%
Undetected crack near Slice 4	Side-a	1,7, 2		
C5	-	Noise		-

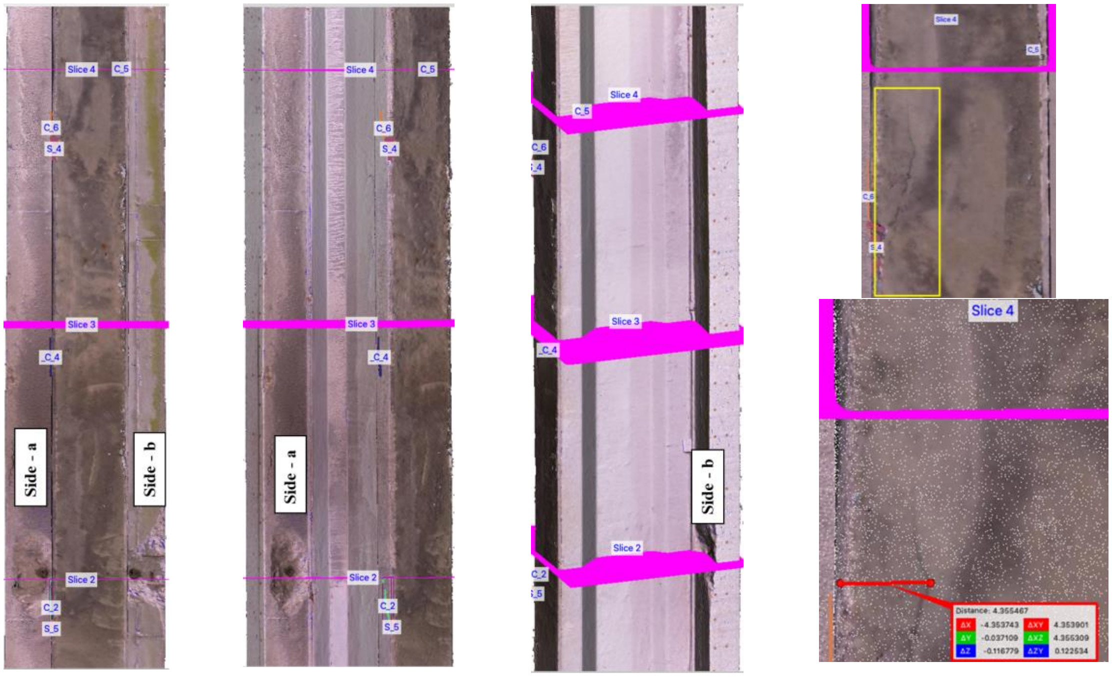
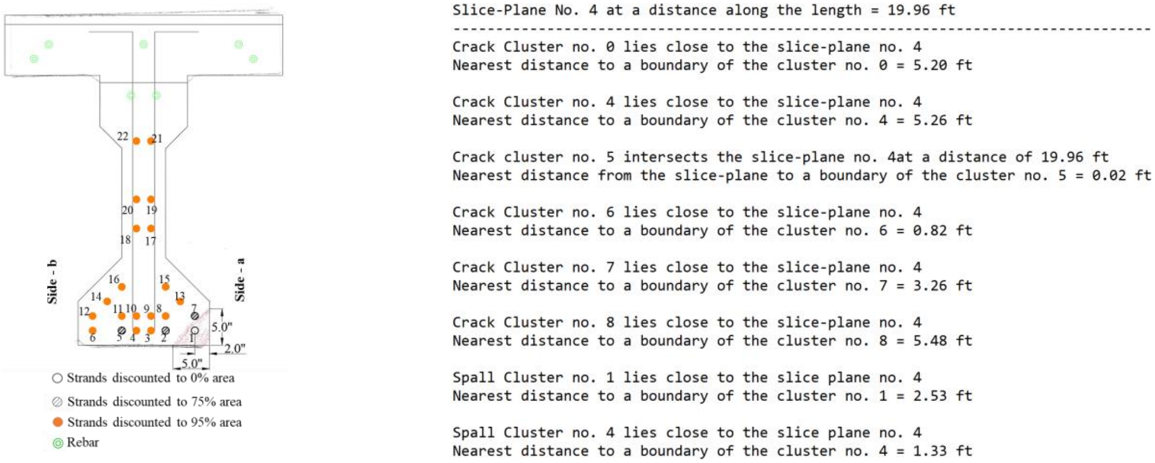


Figure B.8. Damage Clusters in the IW considered for Strength Estimation at Slice-3

Slice No. 4 (At y = 19.96 ft from the left end of the beam)


22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
Side - b
Side - a
5.0"
2.0"
5.0"
○ Strands discounted to 0% area
▨ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

```
Slice-Plane No. 4 at a distance along the length = 19.96 ft
-------------------------------------------------------------------------------
Crack Cluster no. 0 lies close to the slice-plane no. 4
Nearest distance to a boundary of the cluster no. 0 = 5.20 ft

Crack Cluster no. 4 lies close to the slice-plane no. 4
Nearest distance to a boundary of the cluster no. 4 = 5.26 ft

Crack cluster no. 5 intersects the slice-plane no. 4at a distance of 19.96 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 5 = 0.02 ft

Crack Cluster no. 6 lies close to the slice-plane no. 4
Nearest distance to a boundary of the cluster no. 6 = 0.82 ft

Crack Cluster no. 7 lies close to the slice-plane no. 4
Nearest distance to a boundary of the cluster no. 7 = 3.26 ft

Crack Cluster no. 8 lies close to the slice-plane no. 4
Nearest distance to a boundary of the cluster no. 8 = 5.48 ft

Spall Cluster no. 1 lies close to the slice plane no. 4
Nearest distance to a boundary of the cluster no. 1 = 2.53 ft

Spall Cluster no. 4 lies close to the slice plane no. 4
Nearest distance to a boundary of the cluster no. 4 = 1.33 ft
```

a) Overlay of Slice No. 4 with the Cross-Section Drawing

b) Portion of the Text File Showing the Damage Clusters for Slice No.4

Figure B.9. Tools for Strength Estimation at Slice-4 of Lesner Bridge I-Beam 9

Table B.5. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 4

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C0, C4, C6, Undetected cracks near Slice-3	Side-a	1, 7	Unexposed	75%
S4	Side-a	None	-	-
S1	Side-a	1	Exposed	0%
C8	Diaphragm	None	-	-
Undetected crack near Slice 4	Side-a	1,7, 2		
C5, C7	-	Noise		-
Undetected crack near Slice 5	Middle	5	Unexposed	75%

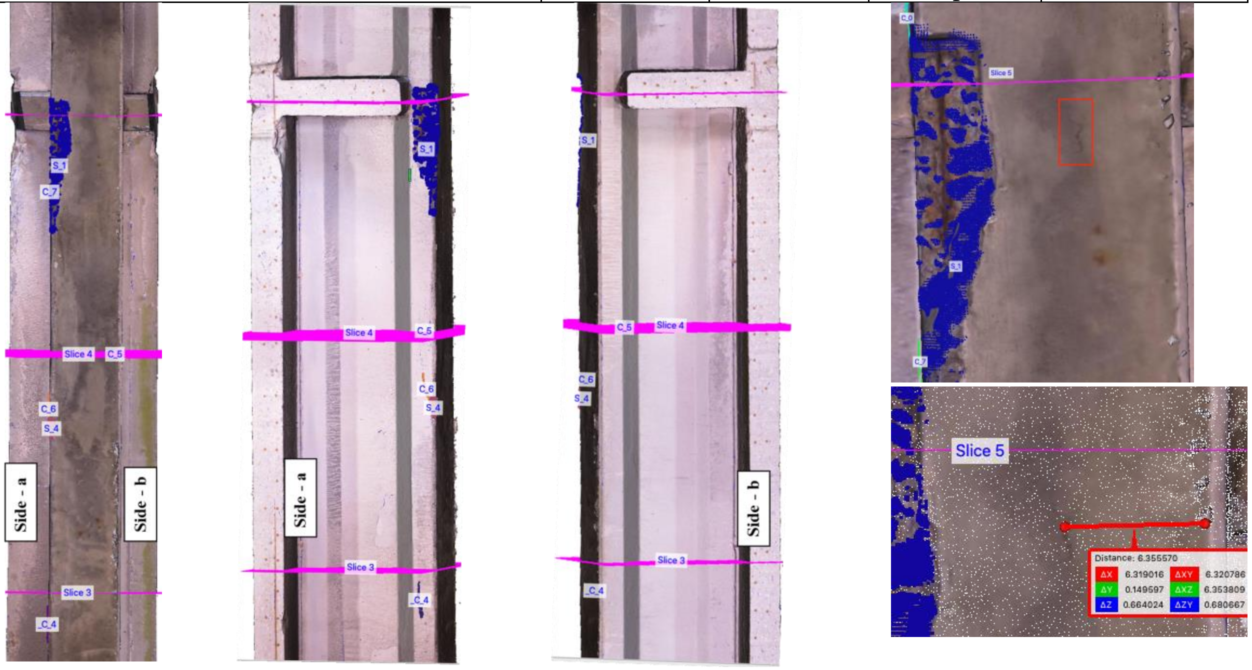
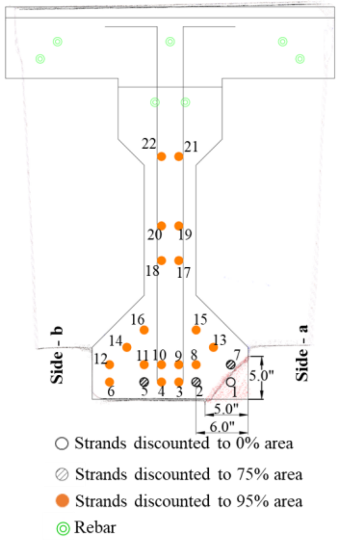
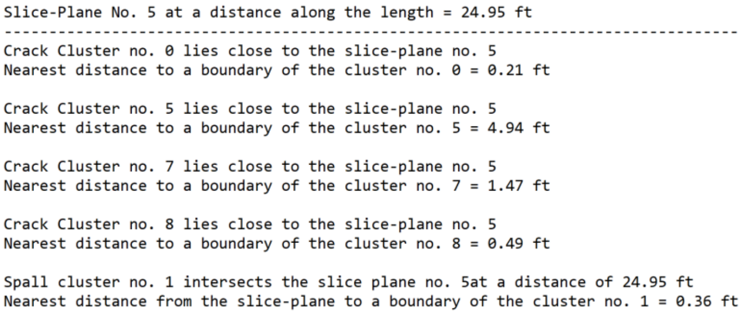


Figure B.10. Damage Clusters in the IW considered for Strength Estimation at Slice-4

Slice No. 5 (At y = 24.95 ft from the left end of the beam)


22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
Side - b
Side - a
5.0"
5.0"
6.0"
○ Strands discounted to 0% area
⊘ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

a) Overlay of Slice No. 5 with the Cross-Section Drawing



```
Slice-Plane No. 5 at a distance along the length = 24.95 ft
-------------------------------------------------------------------------------------------
Crack Cluster no. 0 lies close to the slice-plane no. 5
Nearest distance to a boundary of the cluster no. 0 = 0.21 ft

Crack Cluster no. 5 lies close to the slice-plane no. 5
Nearest distance to a boundary of the cluster no. 5 = 4.94 ft

Crack Cluster no. 7 lies close to the slice-plane no. 5
Nearest distance to a boundary of the cluster no. 7 = 1.47 ft

Crack Cluster no. 8 lies close to the slice-plane no. 5
Nearest distance to a boundary of the cluster no. 8 = 0.49 ft

Spall cluster no. 1 intersects the slice plane no. 5at a distance of 24.95 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 1 = 0.36 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 5

Figure B.11. Tools for Strength Estimation at Slice-5 of Lesner Bridge I-Beam 9

Table B.6. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 5

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C0, Undetected cracks near Slice-3	Side-a	1, 7	Unexposed	75%
S1	Side-a	1	Exposed	0%
C8	Diaphragm	None	-	-
Undetected crack near Slice 4	Side-a	1,7, 2		
C5, C7	-	Noise		-
Undetected crack near Slice 5	Middle	5	Unexposed	75%

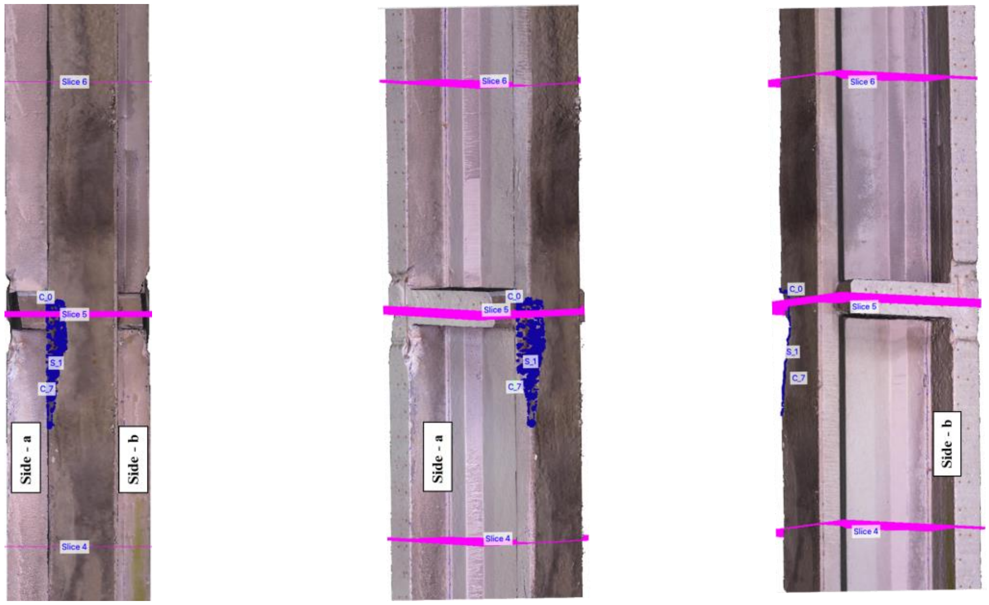
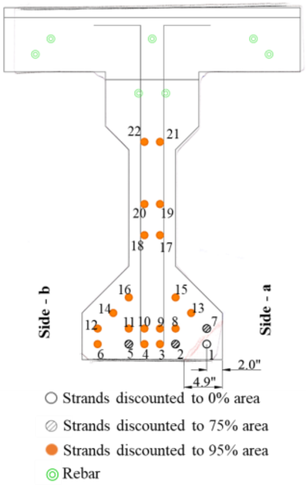
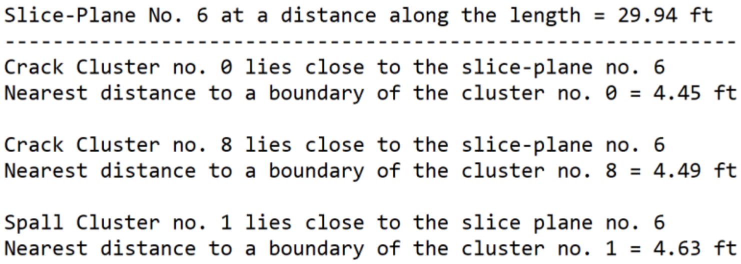


Figure B.12. Damage Clusters in the IW considered for Strength Estimation at Slice-5

Slice No. 6 (At y = 29.94 ft from the left end of the beam)


22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
Side - b
Side - a
2.0"
4.9"
○ Strands discounted to 0% area
▨ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

a) Overlay of Slice No. 6 with the Cross-Section Drawing



```
Slice-Plane No. 6 at a distance along the length = 29.94 ft
-------------------------------------------------------------
Crack Cluster no. 0 lies close to the slice-plane no. 6
Nearest distance to a boundary of the cluster no. 0 = 4.45 ft

Crack Cluster no. 8 lies close to the slice-plane no. 6
Nearest distance to a boundary of the cluster no. 8 = 4.49 ft

Spall Cluster no. 1 lies close to the slice plane no. 6
Nearest distance to a boundary of the cluster no. 1 = 4.63 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 6

Figure B.13. Tools for Strength Estimation at Slice-6 of Lesner Bridge I-Beam 9

Table B.7. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 6

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C0	Side-a	1, 7	Unexposed	75%
S1	Side-a	1	Exposed	0%
C8	Diaphragm	None	-	-
Undetected crack near Slice 4	Side-a	1,7, 2		
Undetected crack near Slice 5	Middle	5	Unexposed	75%

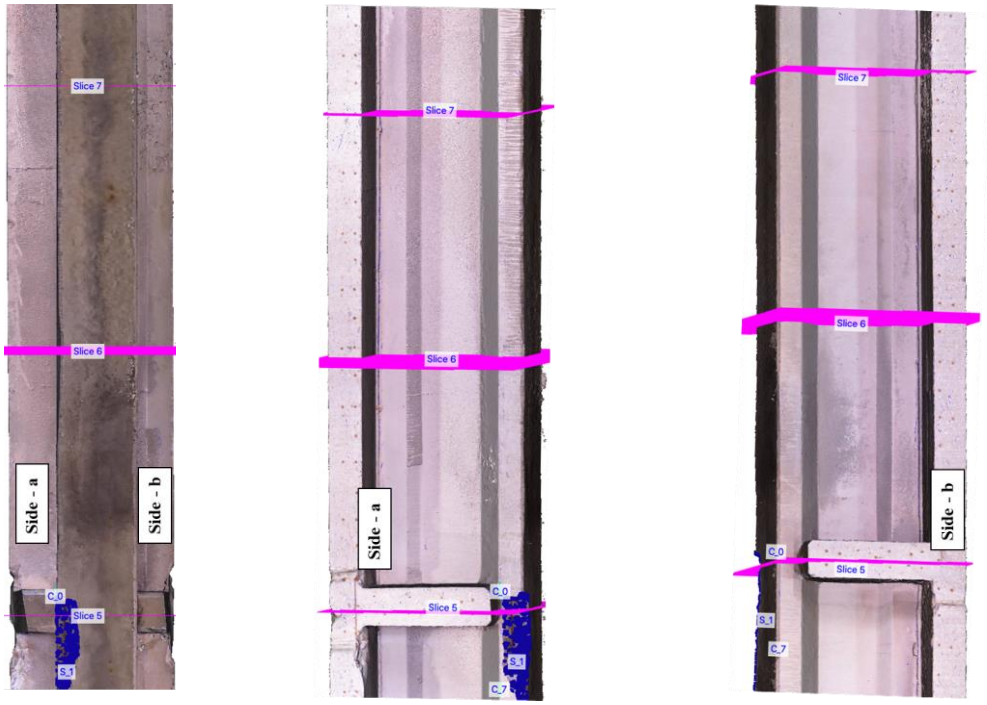
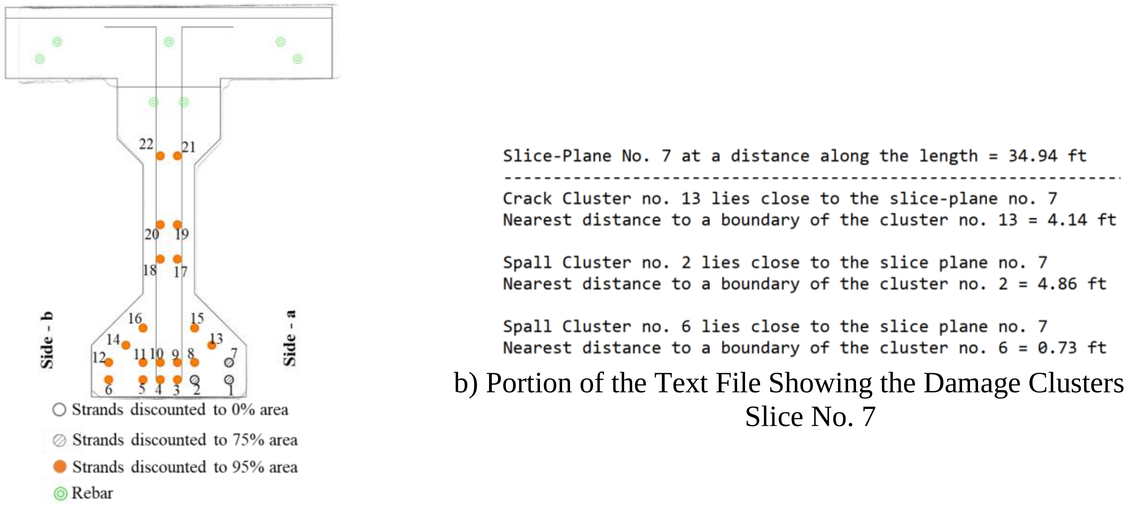


Figure B.14. Damage Clusters in the IW considered for Strength Estimation at Slice-6

Slice No. 7 (At y = 34.94 ft from the left end of the beam)


22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
Side - b
Side - a
○ Strands discounted to 0% area
⊘ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

```
Slice-Plane No. 7 at a distance along the length = 34.94 ft
------------------------------------------------------------
Crack Cluster no. 13 lies close to the slice-plane no. 7
Nearest distance to a boundary of the cluster no. 13 = 4.14 ft

Spall Cluster no. 2 lies close to the slice plane no. 7
Nearest distance to a boundary of the cluster no. 2 = 4.86 ft

Spall Cluster no. 6 lies close to the slice plane no. 7
Nearest distance to a boundary of the cluster no. 6 = 0.73 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 7

a) Overlay of Slice No. 7 with the Cross-Section Drawing

Figure B.15. Tools for Strength Estimation at Slice-7 of Lesner Bridge I-Beam 9

Table B.8. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 7

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C13	Side-a	1, 7	Unexposed	75%
S2	Side-a	None	-	-
Undetected crack near Slice 8	Side-a	2	Unexposed	75%

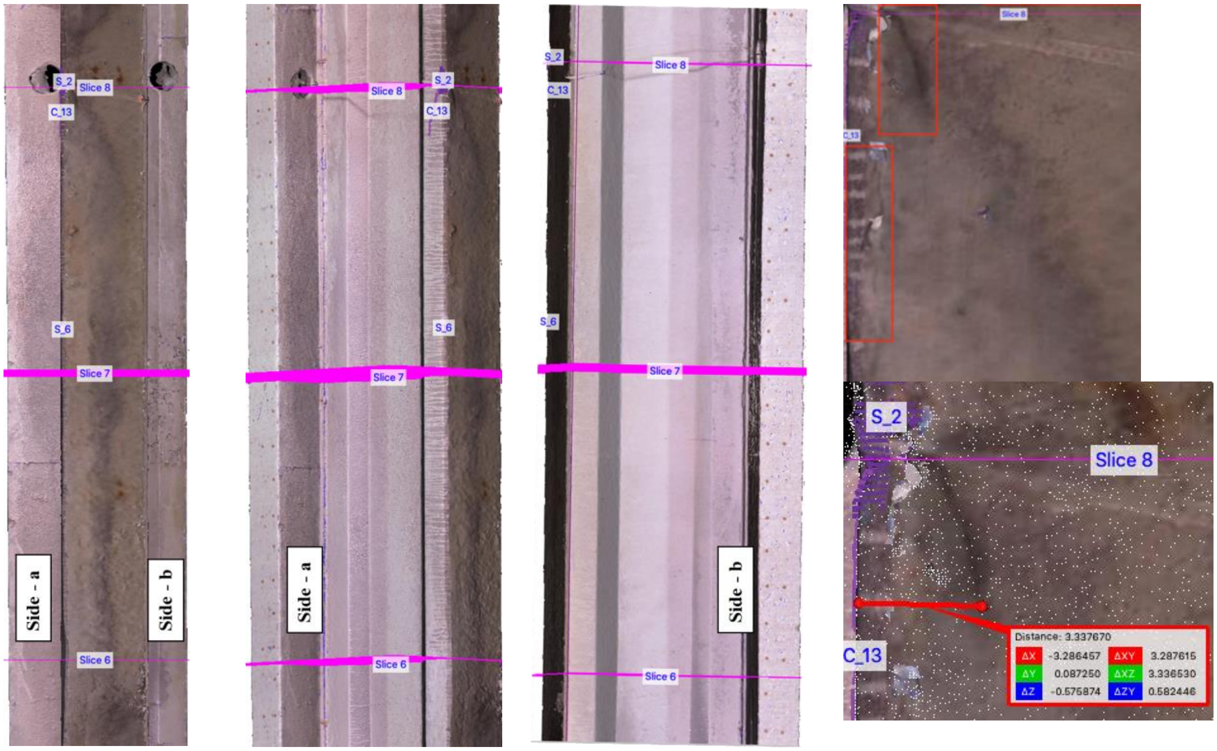
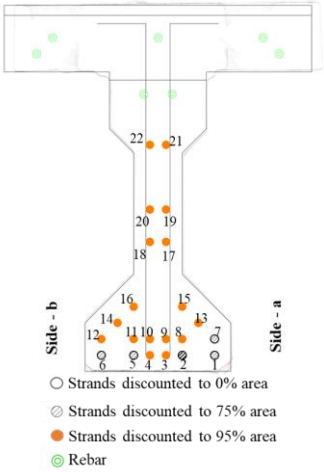


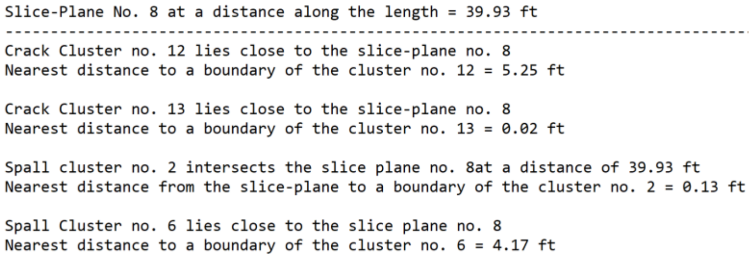
Figure B.16. Damage Clusters in the IW considered for Strength Estimation at Slice-7

Slice No. 8 (At y = 39.93 ft from the left end of the beam)



22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
Side - b
Side - a
○ Strands discounted to 0% area
▨ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

a) Overlay of Slice No. 8 with the Cross-Section Drawing



```
Slice-Plane No. 8 at a distance along the length = 39.93 ft
--------------------------------------------------------------------------------
Crack Cluster no. 12 lies close to the slice-plane no. 8
Nearest distance to a boundary of the cluster no. 12 = 5.25 ft

Crack Cluster no. 13 lies close to the slice-plane no. 8
Nearest distance to a boundary of the cluster no. 13 = 0.02 ft

Spall cluster no. 2 intersects the slice plane no. 8at a distance of 39.93 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 2 = 0.13 ft

Spall Cluster no. 6 lies close to the slice plane no. 8
Nearest distance to a boundary of the cluster no. 6 = 4.17 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 8

Figure B.17. Tools for Strength Estimation at Slice-8 of Lesner Bridge I-Beam 9

Table B.9. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 8

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C12, C13	Side-a	1, 7	Unexposed	75%
S2	Side-a	None	-	-
Undetected crack near Slice 8	Side-a	2	Unexposed	75%
Undetected cracks near Slice 9	Side-b	5, 6	Unexposed	75%

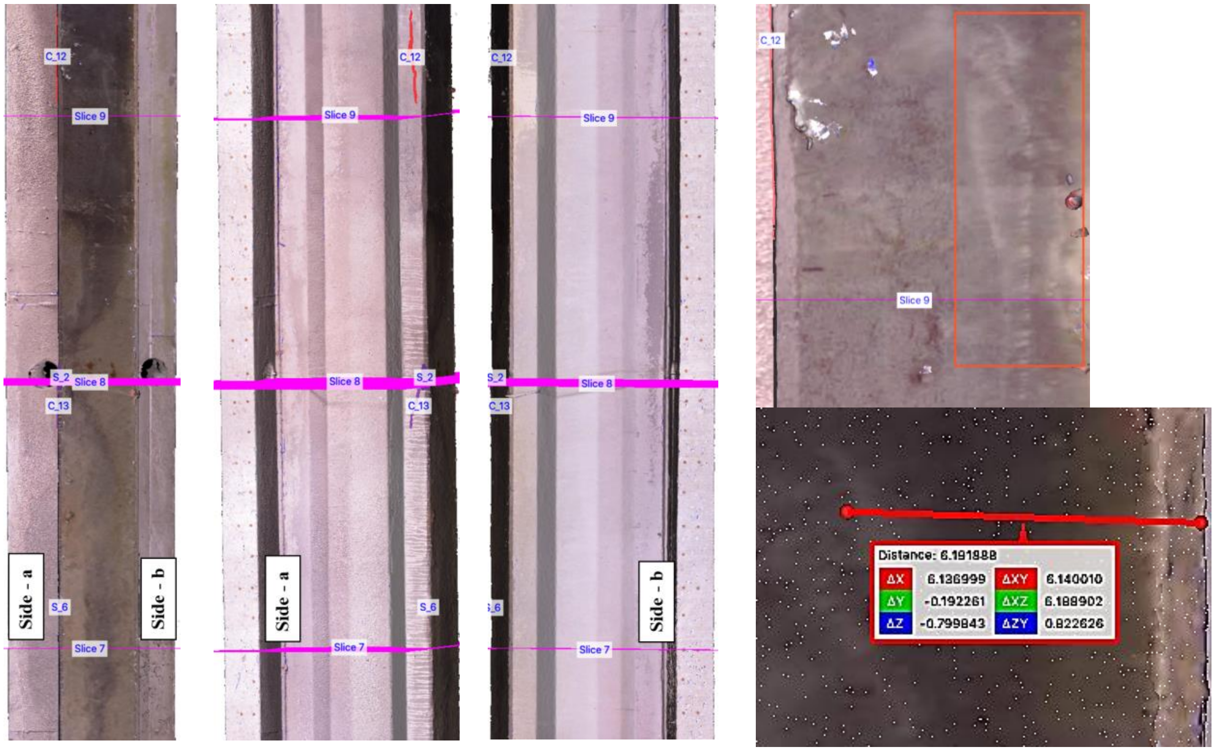
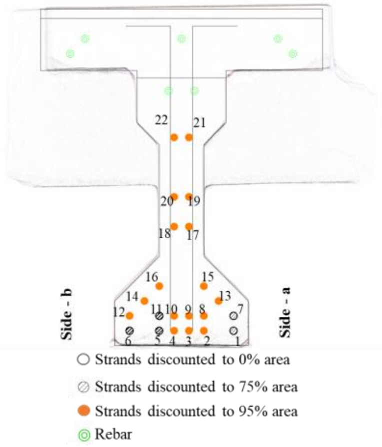
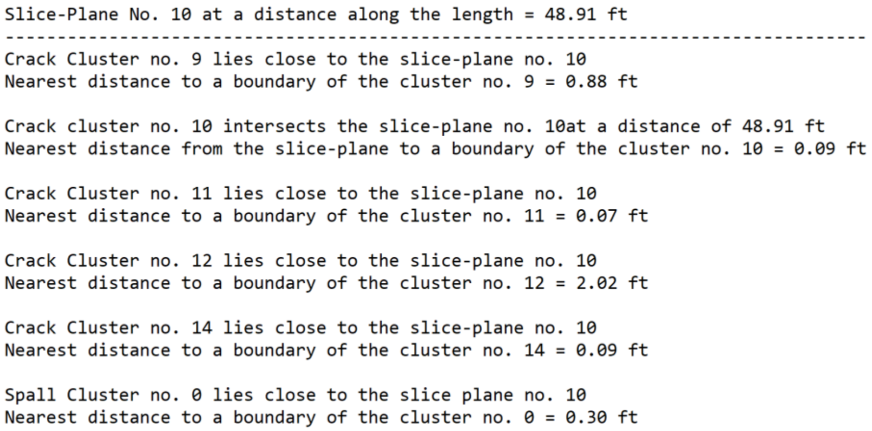


Figure B.18. Damage Clusters in the IW considered for Strength Estimation at Slice-8

Slice No. 10 (At y = 48.91 ft from the left end of the beam)


22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
Side - b
Side - a
○ Strands discounted to 0% area
▨ Strands discounted to 75% area
● Strands discounted to 95% area
◎ Rebar

a) Overlay of Slice No. 10 with the Cross-Section Drawing



```
Slice-Plane No. 10 at a distance along the length = 48.91 ft
--------------------------------------------------------------------------------
Crack Cluster no. 9 lies close to the slice-plane no. 10
Nearest distance to a boundary of the cluster no. 9 = 0.88 ft

Crack cluster no. 10 intersects the slice-plane no. 10at a distance of 48.91 ft
Nearest distance from the slice-plane to a boundary of the cluster no. 10 = 0.09 ft

Crack Cluster no. 11 lies close to the slice-plane no. 10
Nearest distance to a boundary of the cluster no. 11 = 0.07 ft

Crack Cluster no. 12 lies close to the slice-plane no. 10
Nearest distance to a boundary of the cluster no. 12 = 2.02 ft

Crack Cluster no. 14 lies close to the slice-plane no. 10
Nearest distance to a boundary of the cluster no. 14 = 0.09 ft

Spall Cluster no. 0 lies close to the slice plane no. 10
Nearest distance to a boundary of the cluster no. 0 = 0.30 ft
```

b) Portion of the Text File Showing the Damage Clusters for Slice No. 10

Figure B.19. Tools for Strength Estimation at Slice-10 of Lesner Bridge I-Beam 9

Table B.10. Effect of the Damage Clusters on the Prestressing Strands Inside the IW for Slice No. 10

Damage Cluster in IW	Location	Strands Affected	Strand Exposure	% Area Considered
C9, S0	At the end of the beam	Effect on the strength already accounted for	-	-
C10, C11	End diaphragm	Do not affect the strength	-	-
C12, C14	Side-a	1, 7. It was assumed that Strand 13 was not affected due to C13, C14	Unexposed	75%
S2	Side-a	Does not affect any strands	-	-
Undetected Cracks at Slice-9	Side-b to Middle	5, 6 and 11	Unexposed	75%

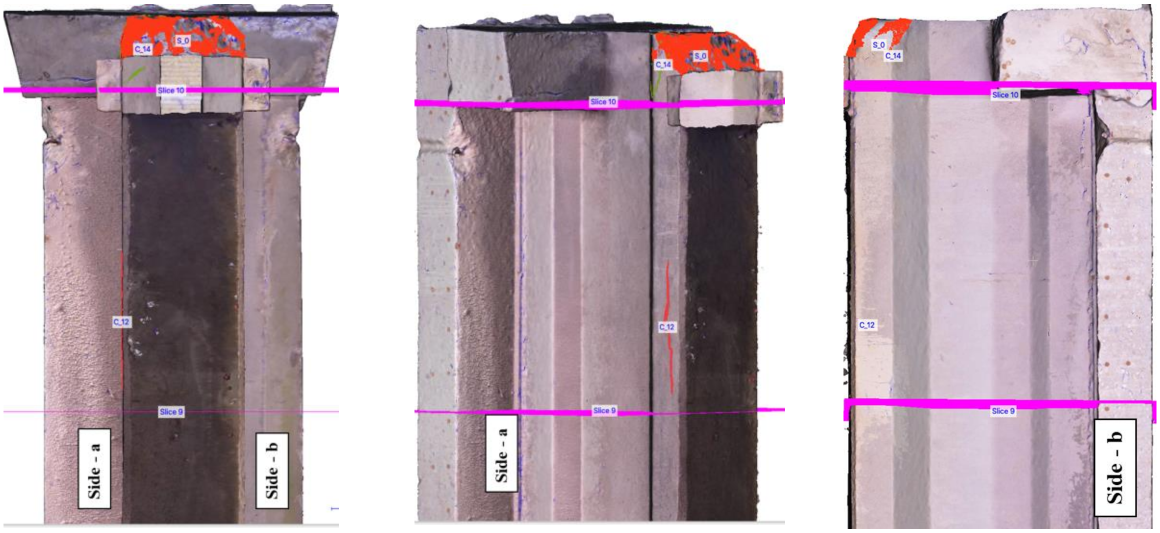


Figure B.20. Damage Clusters in the IW considered for Strength Estimation at Slice-10