5-2018

Effects of Short-Term Strains on Concrete Bulb-Tee Bridge Girders and Analysis of the Practicality of Using Three-Dimensional Models Based on Drone Imagery to Aid in Bridge Inspections

Justin M. Pace
Utah State University

EFFECTS OF SHORT-TERM STRAINS ON CONCRETE BULB-TEE BRIDGE GIRDERS AND ANALYSIS OF THE PRACTICALITY OF USING THREE-DIMENSIONAL MODELS BASED ON DRONE IMAGERY TO AID IN BRIDGE INSPECTIONS

by

Justin M. Pace

A project report submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Civil and Environmental Engineering
(Structural Engineering and Mechanics)

Approved:

_____	_____
Paul J. Barr, Ph.D.	Marc Maguire, Ph.D.
Major Professor	Committee Member

_____
Ziqi Song, Ph.D.
Committee Member

UTAH STATE UNIVERSITY
Logan, UT

2018

ABSTRACT

Effects of Short-Term Strains on Concrete Bulb-Tee Bridge Girders
and Analysis of the Practicality of Using Three-Dimensional Models
Based on Drone Imagery to Aid in Bridge Inspections

by

Justin M. Pace, Master of Science

Utah State University, 2018

Major Professor: Dr. Paul J. Barr
Department: Civil and Environmental Engineering

The University Transportation Center at Utah State University sponsors research through the Federal Highway Administration's University Transportation Center Program. The purpose of this research is to advance technology and expertise in transportation-related fields. This includes research pertaining to bridge design, monitoring, and inspection (the focus of this research).

The first focus area of this research monitored short-term strains at the bottom of girders in a concrete bulb-tee girder bridge. The strains were caused by traffic loading and recorded using strain transducers attached to the bottom of each bridge girder. Patterns in the strain loading were monitored to draw conclusions about the girder distribution factors and the maximum strains caused by various vehicle types. The largest loadings were evaluated and compared against the design loading.

The second focus area of this research examined whether three-dimensional modeling software could create a bridge model of sufficient quality to be used as a supplementary aid in bridge inspections. This was done by capturing imagery of three bridges using a standard, commercially available drone. The imagery was then input into three different modeling programs, and the results were evaluated to determine their suitability.

(187 pages)

PUBLIC ABSTRACT

Effects of Short-Term Strains on Concrete Bulb-Tee Bridge Girders
and Analysis of the Practicality of Using Three-Dimensional Models
Based on Drone Imagery to Aid in Bridge Inspections

Justin M. Pace

The University Transportation Center at Utah State University sponsors research through the Federal Highway Administration's University Transportation Center Program. The purpose of this research is to advance technology and expertise in transportation-related fields. This includes research on the best methods to design, monitor, and inspect bridges.

When a vehicle drives over a bridge, each bridge girder carries a portion of the vehicle's weight. This load causes strain in the bottom of the bridge girders. However, depending on where the vehicle is located on the bridge, some bridge girders carry a higher percentage of the vehicle's load. This study investigated the amount of strain typical vehicles caused in the bottom of each girder. This research also examined the maximum strains in the bridge girders, which is of interest in girder design.

Bridges need to be inspected regularly to determine whether they are structurally sound. However, some bridges have components that can be difficult to reach without specialized equipment. The advent of drone technology potentially offers a cheap, easy method to access and inspect these bridge components.

This research also examined whether a 3D model could be developed using a commercially available drone that could be used to aid bridge inspections. It did so by using a drone to take several pictures of multiple bridges. The pictures were then used to create 3D models of each bridge, and the models were inspected to investigate whether they were of sufficient quality for bridge inspections.

ACKNOWLEDGMENTS

This research was the result of efforts from many people beyond myself who deserve recognition for helping to bring this project to fruition.

First and foremost, I would like to thank my major professor, Dr. Paul J. Barr, for providing me with the opportunity and resources to conduct this research. Without his support, this project would never have been possible. Thank you for your time and encouragement throughout the entire research and writing process. Similarly, I would like to thank my committee members, Dr. Marc Maguire and Dr. Ziqi Song, for answering my questions, guiding this project, and being part of my education at Utah State University.

Thank you to Bridge Diagnostics, Inc. (BDI) for providing continual technical support throughout the short-term strain monitoring process. Special thanks is owed to BDI employee Rene Hamer for helping me troubleshoot the strain acquisition system multiple times throughout the year.

Other people who deserve recognition include Ethan Pickett, the graduate student who initially installed the strain data acquisition system and trained me in its use, and Piero Caputo, who helped me relocate several sensors in September. Recognition should also be given to the City of Nibley for allowing researchers to monitor its bridge and to the University Transportation Center at Utah State University for funding this project.

Finally, I would like to acknowledge the continual encouragement of my family, especially my parents Michael and Rhonda Pace. Their support has allowed me to pursue my graduate degree at Utah State University and to be a part of this project.

Justin M. Pace

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Context

To better understand how concrete bulb-tee girder bridges respond to traffic loading, the University Transportation Center (UTC) at Utah State University sponsored research through the Federal Highway Administration UTC program. This funding is part of a regional transportation center with North Dakota State University. The goal of the research was to quantify short-term strains in concrete bulb-tee girders. The University Transportation Center also sponsored research to determine whether drone imagery could be used to create bridge models to supplement bridge inspections. The findings of both areas of this research are included in this report.

To monitor the short-term strain response of the bridge, eight strain transducers were installed on a recently constructed bulb-tee girder bridge with permission from the City of Nibley. The strain transducers were attached to the bottom of the bridge girders at mid-span. The strain transducers, datalogger, and other instrumentation were provided by Bridge Diagnostics Inc. (BDI). Data from the strain transducers was gathered over a total of five weeks and then analyzed for patterns and trends in traffic flow.

Additionally, to investigate whether images captured via a standard, commercially available drone could be used to create bridge models of sufficient quality to supplement bridge inspections, a DJI Phantom 4 Pro drone was purchased. The drone was used to capture images of three bridges, and the images were input into three different computer modeling programs. The resulting 3D models were examined to determine whether they adequately modeled bridge properties.

1.2 Research Objectives

The objective of the short-term strain monitoring area of research was to analyze the short-term strains recorded in a bulb-tee girder bridge due to everyday traffic flow. This was achieved by monitoring various bridge girders in a selected bridge. Monitoring took place over a total of five weeks at different times of the year.

The resulting strain data was analyzed to investigate trends in both everyday traffic and extreme loading events. This included comparison of actual traffic loads to design loads and analysis of potential causes of extreme loading events. Conclusions were also drawn about the distribution of strain within the bridge girders due to everyday traffic loading. Finally, this research examined short-term loading trends and analyzed the strain caused by various vehicle types.

The objective of the drone modeling area of research was to determine whether a three-dimensional computer model based off of drone imagery could be used to supplement bridge inspections. To accomplish this goal, images were captured of three bridges using a drone. The images were then input into three different modeling programs. The resulting 3D models were then compared qualitatively and quantitatively against each other and against actual measurements taken of the bridge. Conclusions on the viability of the drone modeling procedure were drawn based on these results.

1.3 Organization of Paper

This paper is organized as follows:

Chapter 2: Examines previous research related to girder distribution factors, load distribution in bridges, live load testing, short-term strains, drone

imagery capture, and 3D modeling. Reviews details pertinent to the research conducted for this project.

Chapter 3: Provides details about the monitored bridge and the instrumentation used. Discusses how stain data was collected and processed. Examines the implications of the processed strain data as it relates to this research.

Chapter 4: Provides details about the drone and modeling software used for this project. Also discusses how drone imagery was captured and examines the results of the finished models. Draws conclusions about the viability of using drones and 3D models to conduct bridge inspections.

Chapter 5: Presents the final conclusions of this project as well as recommendations for additional research.

CHAPTER 2

LITERATURE REVIEWS

2.1 Live-Load Analysis of Posttensioned Box-Girder Bridges

In this journal article, Hodson et al. (2012) summarize a study that was performed on the Lambert Road Bridge, a Type “d” box-girder bridge near Elk Grove, California. As part of the study, a live load test was performed on the bridge, and the results were used to validate a finite-element model (FEM) of the bridge. The FEM was then used to obtain moment distribution factors for the exterior and interior girders of the bridge. The finite-element analysis was performed while including the stiffness of bridge parapets and again without their effect. The obtained distribution factors were then compared to distribution factors calculated using procedures in the AASHTO Standard Specifications.

The researchers found that the AASHTO Standard distribution factors were conservative for both the interior and exterior girders when the stiffening effects of the bridge parapets were not included. However, when the bridge parapets were included in the FEM, the distribution factors for the exterior girders were unconservative by approximately 2 to 9%. The distribution factors for the interior girders remained conservative. This finding is significant because the AASHTO LFRD specifications do not provide for the stiffening effect of parapets in distribution factors for Type “d” box-girder bridges. Thus, design using the AASHTO Standard Specifications for external girders is unconservative.

A parametric study was performed to investigate properties that could potentially influence the distribution factors of the bridge. The seven factors that were evaluated

were span length, girder spacing, skew, deck overhang, continuity, deck thickness, and the thickness of the bottom girder flange. The first four parameters are accounted for in the AASHTO Standard Specifications, while the later three were not.

Analyses were performed by incrementally changing one bridge parameter at a time in the FEM. From these analyses, it was concluded that span length and girder spacing have the greatest effect on the distribution factors. Skew and continuity had a lesser effect on the distribution factors, while deck thickness, bottom flange thickness, and deck overhang have little to no effect on distribution factors.

Based on their calculation of the FEM and parametric study, Hodson et al. proposed a new formula for the distribution factor of an external girder. This formula accounts for both span length and girder spacing and is viewable in Equation 1, where g = exterior distribution factor applied to the interior girder, d_e = horizontal distance from the centerline of the exterior web to the inside of the traffic barrier [feet (m)], S = girder spacing [feet (m)], and L = length of the span [feet (m)].

$$g = \left(\frac{(d_e + S) \cdot 0.85}{S}\right)\left(\frac{1}{L}\right)^{0.15}$$ Equation 1 (Hodson et. al. 2012).

Finally, the load rating of the bridge was calculated using both the distribution factors from the AASHTO LFRD Specifications and from the proposed formula. The load rating from the FEM analysis was 29% higher than that obtained using the AASHTO LFRD Specifications. Thus, it was concluded that the AASHTO LFRD Specifications were overly conservative, and the proposed formula was recommended.

2.2 Field Test and 3D FE Modeling of Decked Bulb-Tee Bridges

In this journal article, Ma et al. (2007) summarized research on the effects of intermediate diaphragms and shear connectors on a bridge's live load distribution factors. The research also compared values for a single-lane loading condition to existing research on double-lane loaded bridges. This research focused on bridges with decked precast, prestressed concrete girders.

The study investigated four pairs of bridges. The bridge pairs represented a variety of geometries. The two bridges within each pair had similar geometries in order to allow verification of results.

Prior to testing, strain transducers were attached to the bridge at three locations: near the girders' centroid and end of span to measure shear stresses, at the bottom of the girders at mid-span to measure flexural stress, and on the intermediate diaphragms to measure axial stresses.

Two types of testing were performed on each bridge: continuous and static loading. During continuous loading, a heavily loaded dump truck was driven slowly across the bridge without stopping. During static load testing, the truck would stop at predetermined locations in order to allow changes in strains at that location to be recorded. As opposed to existing research that focused on double-lane loading, only one lane of the bridges was loaded at a time.

Using the results from the live load testing, a FE model was developed using ABAQUS software. The calibrated model was then used to determine three different live load distribution factors for a variety of bridges. Live load distribution factors were

determined when a single lane of the model was loaded, when two lanes of the model were loaded, and using AASTHO LFRD models.

The results showed that in every case the single-lane distribution factor was smaller than the double-lane distribution factor. In fact, the single-lane distribution factor was usually smaller by over 15%. Additionally, both mentioned distribution factors were typically conservative when compared with the distribution factor calculated using LRFD methods. Because the current AASHTO LRFD only allowed the one equation for both the single-lane and double-lane loading types, the researchers recommended the addition of a new equation for the single-lane loading condition.

The researchers also analyzed the effects of intermediate diaphragms and shear connectors on a bridge's live load distribution factors. It was found that the inclusion of a single diaphragm provided significant advantages over no diaphragms. However, adding additional diaphragms did not provide a notable decrease in distribution factors.

When analyzing shear connectors, researchers discovered that changing the shear connector spacing (and thus the number of shear connectors) did not have an effect on the live load distribution factors. The only difference that changing the connector spacing had was in the magnitude of vertical and horizontal forces carried by each connector. Therefore, Ma et al.'s conclusion was that if the shear connectors' capacity was adequate for the carried forces that any number of connectors could be used without affecting the live load distribution factors.

2.3 Response of Prestressed Concrete I-Girder Bridges to Live Load

In 2001, Schwarz and Laman summarized their research on three concrete I-girder bridges. During the study, strain transducers were attached at mid-span to the bottom flange of each bridge girder. The transducers then recorded strains caused in the bridge by both normal traffic and test trucks. Research focused on three factors: dynamic load allowance (DLA), girder distribution factors (GDF), and service level stresses.

The study evaluated different factors that could potentially affect the DLA– number of axles per vehicle, vehicle speed, and span length. First, it was noted that the DLA has an inverse relationship with a bridge's maximum static stress. While the study found that the number of axles per vehicle did not have a statistically significant role in determining the DLA, it did find that the DLA increased by approximately 50% when vehicle speed was increased from 45 to 55 mph (72 to 88 kph). Finally, Schwarz and Laman concluded that there was no definable relationship between span length and a bridge's DLA.

Because GDFs are important factors in bridge girder design, the study also compared the measured GDFs from the live load tests to AASHTO code. GDFs were developed for both single-lane and multi-lane loading. Researchers found that between the three bridges, the measured GDF for a single lane was less than the GDF calculated using AASHTO code by at least 17%. For double-lane GDFs, the code's GDF exceeded the measured GDF by at least 22%.

The graph shown in Figure 1 shows the results from Bridge 1 of the study. The pair of lower, solid lines shows the measured GDFs from single-lane testing, while the dashed lines show the measured GDF ±1 standard deviation. The upper line "Test Trucks

Side-by-Side" shows the measured GDF when a pair of trucks was driven over the bridge simultaneously. The upper line labeled "GDF_2" shows the sum of the two single-lane GDFs. Finally, the four uppermost horizontal lines show the AASHTO LFRD and AASHTO Standard Specifications.

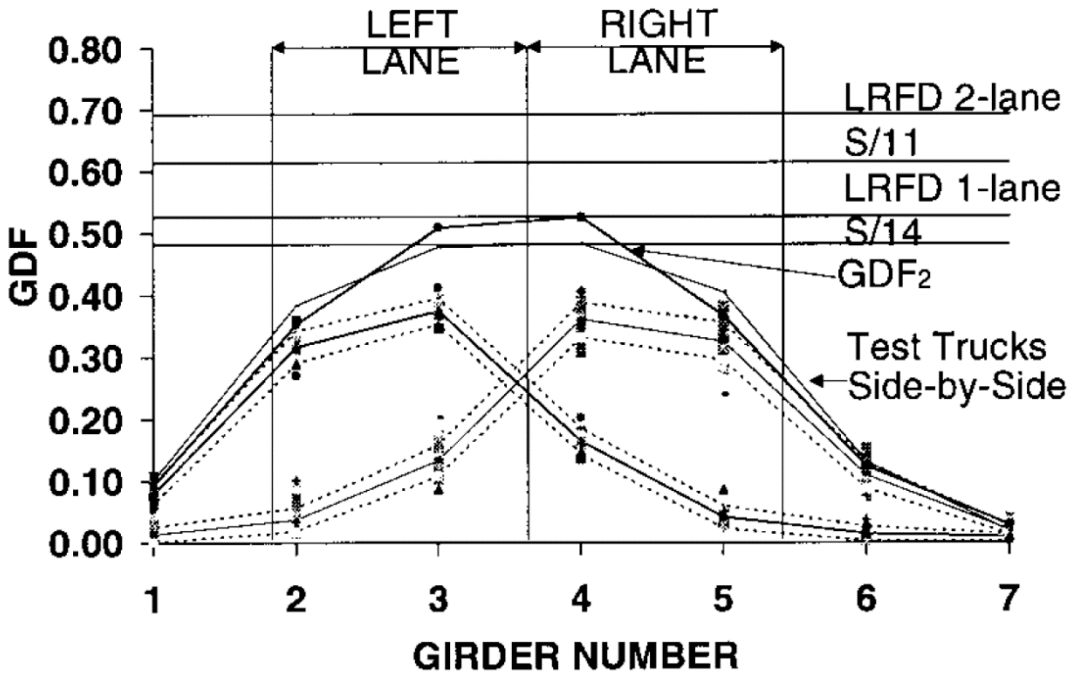

Figure 1. Bridge 1 girder distribution factors by girder (Schwarz and Laman 2001).

A grillage model was also developed for each bridge and compared to the live load test results. It was found that the models agreed closest with the real-world results when mid-span diaphragms were not included. As shown in Figure 2, the grillage models typically agreed well with the measured results. The model GDFs were typically conservative by 2–11%.

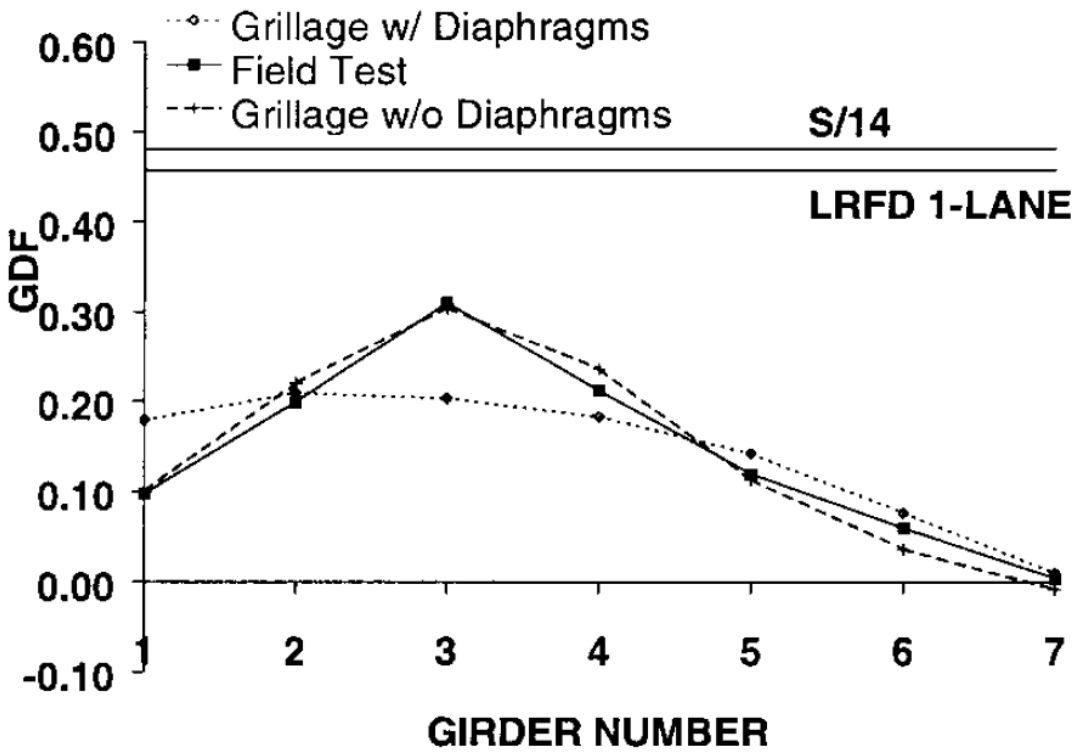

Figure 2. Comparison of Bridge 2 numerical and measured GDFs for test truck runs (truck in left lane on centerline) (Schwarz and Laman 2001).

From the normal traffic data, service level stresses were compiled and compared against a HS-20 truck. From this data, it was found that the interior girders directly beneath the traffic lanes had a far greater likelihood of reaching higher stresses. On Bridge 1, the girders underneath the right traffic lane experienced loading equivalent to a HS-20 truck once a day, while the remaining girders were not likely to ever reach that level of stress within the bridge's lifespan.

Schwarz and Laman reached numerous conclusions due to the findings of this study. They concluded that while vehicle speed plays a factor in DLAs, span length and the number of vehicle axles do not. They found that both AASHTO and modeled GDFs were conservative when compared to measured values. Finally, they also concluded that

interior girders directly below the traffic lane are much more likely to see higher stress values than other girders in the bridge.

2.4 Unmanned Aerial Vehicle Bridge Inspection Demonstration Project

This report was published in 2015 by the Minnesota Department of Transportation (MnDOT). Over the course of the research project, Zink and Lovelace investigated the effectiveness of using drones to conduct bridge inspections. Four bridges of various types and sizes were examined using a variety of image capturing systems, including still images, video, and infrared imagery. The inspections conducted by the drone were then compared to the last routine physical inspection conducted on the bridge in order to determine the drone's effectiveness.

This research was conducted using an Aeryon Skyranger drone (Skyranger), valued at $140,000 (See Figure 3). The Skyranger had the ability to change payloads (i.e. cameras), and standard, optical zoom, and infrared cameras were used over the course of the study. A 360° video camera was also manually installed on top of the drone, but did not work due to Wi-Fi interference. Additional drones were purchased for this project, but were not approved by the Federal Aviation Administration (FAA) in time to be used for this research.

Figure 3. The Aeryon Skyranger (Zink and Lovelace 2015).

One of the largest obstacles encountered during the research was the drone's inability to inspect the underside of a bridge. The gimbal for the Skyranger's camera was unable raise the camera to a purely vertical orientation, making it difficult to inspect the underside of the bridge deck. The Skyranger also featured a "return to home" safety feature, which caused the drone to return to its takeoff point if it lost GPS signal. Because the drone would lose the GPS signal when flying underneath the concrete deck, it would cause the drone to fly straight upwards into bridge deck and crash.

One of the four bridges that the team inspected, Bridge 13509 in Chisago County, was a prestressed concrete girder bridge similar in length to the Nibley Bridge inspected for this project. The research team concluded of Bridge 13509 "Because of its smaller size, this bridge may not be an ideal candidate for using a UAV [unmanned aerial vehicle] as an inspection tool" (Zink and Lovelace 2015). However, this was due in part

to the Skyranger's inability to fly underneath the bridge. The drone performed well in regards to inspecting all other bridge components.

The research also explored mapping and modeling functionality based on the images taken by the drone. At Bridge 13509, the drone was used to create an orthographic overhead map of the bridge and surrounding area (See Figure 4). At the largest bridge (the Arcola Railroad Bridge), researchers used images from the drone to create a 3D model of one of the bridge foundations positioned mid-river (See Figure 5). Both the orthographic map and 3D model were created using Postflight Terra 3D software.

Figure 4. Bridge 13509 Orthographic Map (Zink and Lovelace 2015).

Figure 5. Orthographic model of bridge foundation (Zink and Lovelace 2015).

The researchers drew several conclusions as a result of this project's findings. First, they concluded that drones could safely be used to perform bridge inspections and that they were most suitable for inspecting larger bridges. They also determined that drones could access difficult-to-reach areas cheaper than an under bridge inspection vehicle. However, the research team also found that drones could only provide a level of detail equivalent to a high-quality photograph. While measurements could be estimated from the photos, they could not provide accuracy equivalent to a hand-on inspection.

As a result of the project, the researchers recommended that the FAA rules be updated in order to be more accommodating to drone research. They also identified

several areas where drone technology could improve, including the ability to fly underneath bridge decks. The authors were encouraged by the fact that the drone technology was advancing rapidly, and expected it to be able to meet the needs of bridge inspectors within a matter of years. In conclusion, the researchers determined that drones present a potentially useful tool to provide significant cost-savings for bridge inspections when hand measurements are not required.

2.5 3D Scene Reconstruction for Robotic Bridge Inspection

In 2015, researchers David Lattanzi and Gregory R. Miller explored the effectiveness of using robotic systems to create high-quality 3D bridge models for bridge inspections. The goal of this research was to see whether a 3D model could be used to accurately inspect a bridge. The research consisted of two parts: first, comparing 3D modeling methods via a series of small-scale tests and second, using images captured from a drone to create a 3D bridge model.

The researchers began by reviewing the two major methods available to create a 3D model, dense structure from motion (DSfM) and image mosaicing (IM). DSfM modeling creates a 3D model by identifying easily recognizable points on an object across multiple images. It then uses the location of these points in relation to each other to create a 3D point cloud. Finally, Poisson meshing is used to create a triangular mesh between these objects, resulting in a 3D model. For this study, the researchers used 123DCatch software to create the DSfM models.

Image mosaicing is different from DSfM modeling in that the base model is not created using photos. Instead, the model must be created separately beforehand by the user. Images are then superimposed onto the structure to create a textured 3D model.

For the first focus area of the research, the authors tested the limitations of both the DSfM and IM methods by creating three small-scale models. The models were created using images taken of an aluminum S-shape, a skeletal aluminum/acrylic structure, and a steel bridge column embedded in a concrete footing. The images for these objects were taken by hand using a 12MP camera with a focal length of 50 mm at an ISO of 400. The images were taken close to the object and in sequential order, in the same manner that a drone or other robotic system would take them.

Each modeling method was found to have distinct advantages and disadvantages. DSfM modeling had difficulties when points were coplanar. The DSfM modeling software also struggled on smooth, low-texture surfaces where an abundance of recognizable points was not available. This led to poor models of both the smooth S-shape and the skeletal aluminum/acrylic structure (Figure 6). IM modeling did well modeling both of these objects.

Figure 6. 3D models of the aluminum/acrylic structure using IM (left) and DSfM (right) modeling (Lattanzi and Miller 2015).

The principle limitation of IM modeling was that it was only as accurate as the underlying model. If the underlying model was inaccurate, the resulting model would be inaccurate as well. This is easily seen in Figure 7, where the IM model was unable to pick up spalling in the concrete of the bridge column's footing. However, the DSfM model was easily able to detect this irregularity. An additional disadvantage of IM modeling was that if an existing 3D model was not available for an object, one had to be created before IM modeling could be used.

Figure 7. Spalling on the column (left), IM model (center), and DSfM model (right) (Lattanzi and Miller 2015).

Measurement accuracy was also investigated during the three small-scale tests. Before photographing an object, imperfections or marks were measured at three different locations. These baseline measurements were compared to measurements taken from each of the models. Measurements taken by hand and from the models are given in Table 1.

Table 1. Measurements on aluminum section, scale structure, and bridge column (nearest 0.5 mm) (Lattanzi and Miller 2015).

	Hand Measurement (mm)	IM Model (mm)	DSfM Model (mm)
S-Shape			
Pen Marking	32x1.5	32x1.5	34.5x1.5
Pen Marking	79.5x1.5	82.5x1.5	82.5x1.5
Mud Smear	111x25.5	82.5x25.5	111x31
Structure			
Pen Marking	159x3	159x3	—
Tape Residue	35x51	35x54	—
Surface Gouge	391x1.5	365x1.5	—
Bridge Column			
Pen Marking	952.5x1.5	965x3	959x3
Tape	219x25.5	197x25.5	216x25.5
Circular Hole	66.5x66.5	63.5x60.5	63.5x63.5

From the results of the small-scale testing, the researchers selected a target resolution of 20.3 pixels per inch (0.8 pixels per mm) for the full-scale model of the bridge. Images of the bridge were captured using a Parrot AR Drone 2.0 UAV. To achieve the desired pixel density, the drone was flown at a distance 3 feet away from the structure. A photograph of the Parrot AR Drone 2.0 UAV is shown in Figure 8.

Figure 8. Parrot AR Drone 2.0 UAV (Parrot S.A. 2017)

The bridge selected for the full-scale model was a reinforced concrete arch bridge used by pedestrians. Because no preexisting model of the bridge was available, only a DSfM model was created. Figure 9 shows a photograph of the bridge. Figure 10 shows pictures of the DSfM model.

Figure 9. Photograph of Pedestrian Bridge (Lattanzi and Miller 2015).

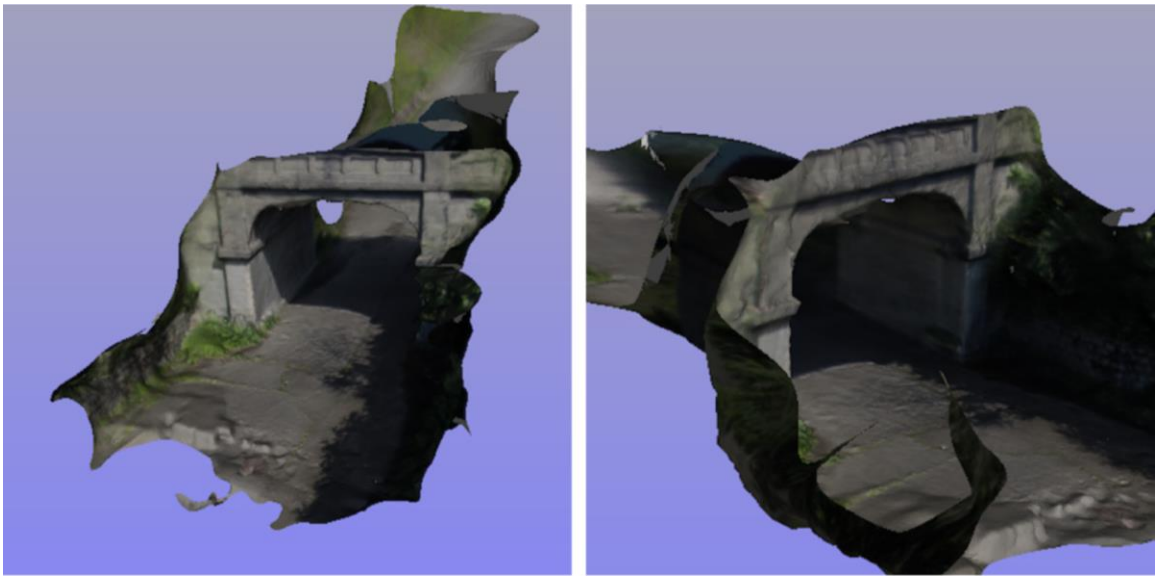

Figure 10. DSfM Model of Pedestrian Bridge (Lattanzi and Miller 2015).

One obstacle that the researchers encountered was the contrast in brightness between the underside of the bridge and the surrounding area. When taking pictures from the surrounding area, the bright sunlight caused the camera to underexpose the area underneath the bridge. The shadows underneath the bridge also made it harder for the DSfM software to identify points for comparison. The researchers suggested that this could be overcome by supplemental lighting attached to the drone.

As a result of this research, the authors concluded that both DSfM and IM modeling worked well for virtual bridge reconstruction. While 3D model research is ongoing, they concluded the two modeling methods used in the study would not likely change significantly in the near future. Most of the modeling errors encountered during the full-scale bridge test were anticipated by the results of the small-scale models, and were inherent to the type of modeling used. Overall, the researchers concluded that while

work remained to be done, it would be feasible to use drones and other robotic systems to create 3D bridge models for inspections.

CHAPTER 3

SHORT-TERM STRAIN MONITORING

3.1 Bridge Description

The Nibley Bridge (National Bridge Inventory Structure Number 005065F) is located in Nibley, UT. The bridge is owned by Nibley City and was designed by Forsgren Associates Inc. in conjunction with Design West Architects. The bridge is located at 41°41’8.41” N and 111°49’56.42”W. The bridge was designed to carry vehicle and pedestrian traffic on 2600 South across the Blacksmith Fork River to service Ridgeline High School. Figure 11 shows a photograph of the Nibley Bridge taken on 28 August 2017.

The Nibley Bridge was constructed from 2015-2016. Because it is the main access point for Ridgeline High School, it experiences relatively high traffic for the area depending on the time of day and season. The calculated Annual Daily Traffic for the bridge is 1706 vehicles/day during the school year and 1436 vehicles/day during the summer. While the majority of vehicles crossing the bridge belong to students, parents and school officials, there is also a relatively high volume of heavy vehicles (i.e. school buses and construction equipment) that use the bridge.

Figure 11. Photograph of the Nibley Bridge.

The bridge spans 88 feet (26.82 m) across the Blacksmith Fork River. It is 60 feet (18.28 m) wide with one 12-foot (3.66-m) lane of traffic each direction. There is also an 8-foot (2.44-m) median, two 5-foot (1.52-m) bike lanes, two 1.42-foot (0.43-m) wide parapets, and two sidewalks with a combined width of 15.17 feet (4.62 m). There is no skew or superelevation in the bridge's structure. However, the roadway itself curves slightly on top of the bridge, causing the width of the sidewalk to vary slightly along the bridge's length. Figure 12 shows a cross-section of the bridge at mid-span.

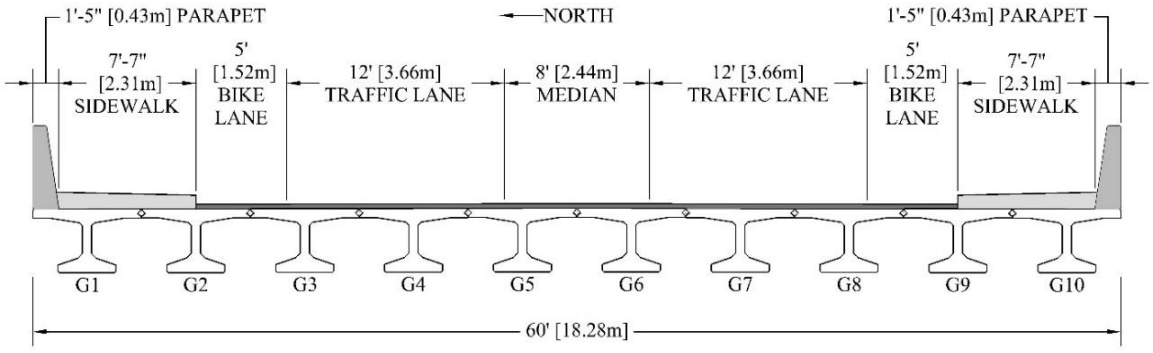

Figure 12. A cross-section of the Nibley Bridge at mid-span showing bridge dimensions.

The 3-inch (7.62 cm) asphalt deck surface is supported by a series of ten bulb-tee girders. Together, the girders form the deck and superstructure of the bridge. For this research, the girders were numbered G1 through G10, with G1 being the northernmost girder and G10 being the southernmost girder.

The basic girder dimensions follow that of a UDBT42 girder, with minor dimension adjustments. Every girder has the same typical cross-section, with modifications to allow tie-ins for utility pipes, barriers, etc. The dimensions of a typical girder are shown in Figure 13. The top of each girder contains a 2% slope in order to allow water to drain off the roadway.

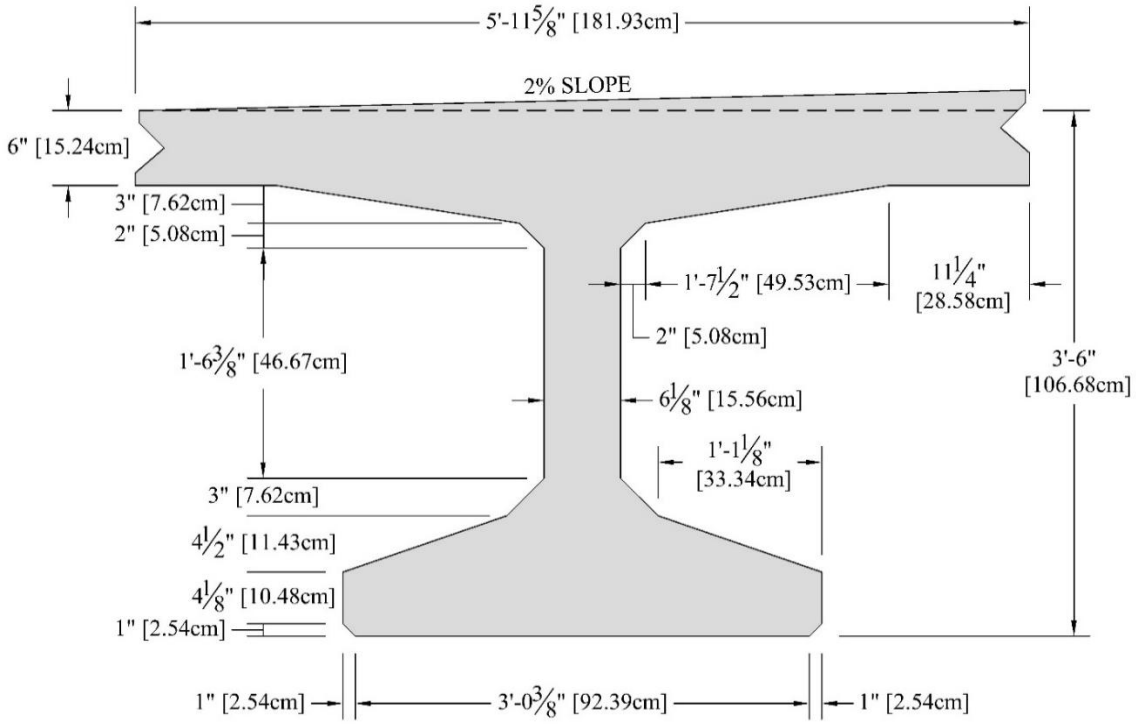

Figure 13. Dimensions of a typical bridge bulb-tee girder at centerspan.

The girders were cast during December 2015 in Salt Lake City, Utah. The girders are prestressed longitudinally with 0.6-inch (1.52-cm) diameter AASHTO M203 Gr270 low relaxation strands. The strands have a yield strength of 270 ksi (1861.6 MPa). Ten harped strands and twenty-four straight strands were used. Each prestressed strand was tensioned with a minimum force of 44 kips per strand, equivalent to 202.5 ksi (1398 MPa) stress.

The girders were designed to have an initial compressive strength (f'_{ci}) of 6.0 ksi (41.37 MPa) and a compressive strength at 28 days (f'_c) of 8.5 ksi (58.61 MPa). During casting, Girders 1 and 5 were monitored to determine actual concrete strengths. For Girder 1, f'_{ci} equaled 6.91 ksi (47.64 MPa) and f'_c equaled 10.94 ksi (75.45 MPa). For Girder 5, f'_{ci} equaled 6.72 ksi (46.33 MPa) and f'_c equaled 11.77 ksi (81.15 MPa). The girders were allowed to cure for six weeks before being transported to the bridge site. The girders were then connected longitudinally using shear connectors that were spaced 5 feet (1.52 m) apart.

While there is 88 feet (26.82 m) between the centerlines of the bearings, individual girders were constructed at 89.5 feet (27.28 m) long. The girders are supported on abutments that measure 91 feet (27.74 m) to the outside face and 85 feet (25.91 m) to the inside face. The abutments are 9 feet (2.74 m) tall at the exterior girders and 9.48 feet (2.89 m) along the bridge centerline. Loads are transferred from the abutments to eight 15-foot-deep (4.57 m) piles that are 12.75 inches (32.39 m) in diameter. A plan view of the bridge is given in Figure 14.

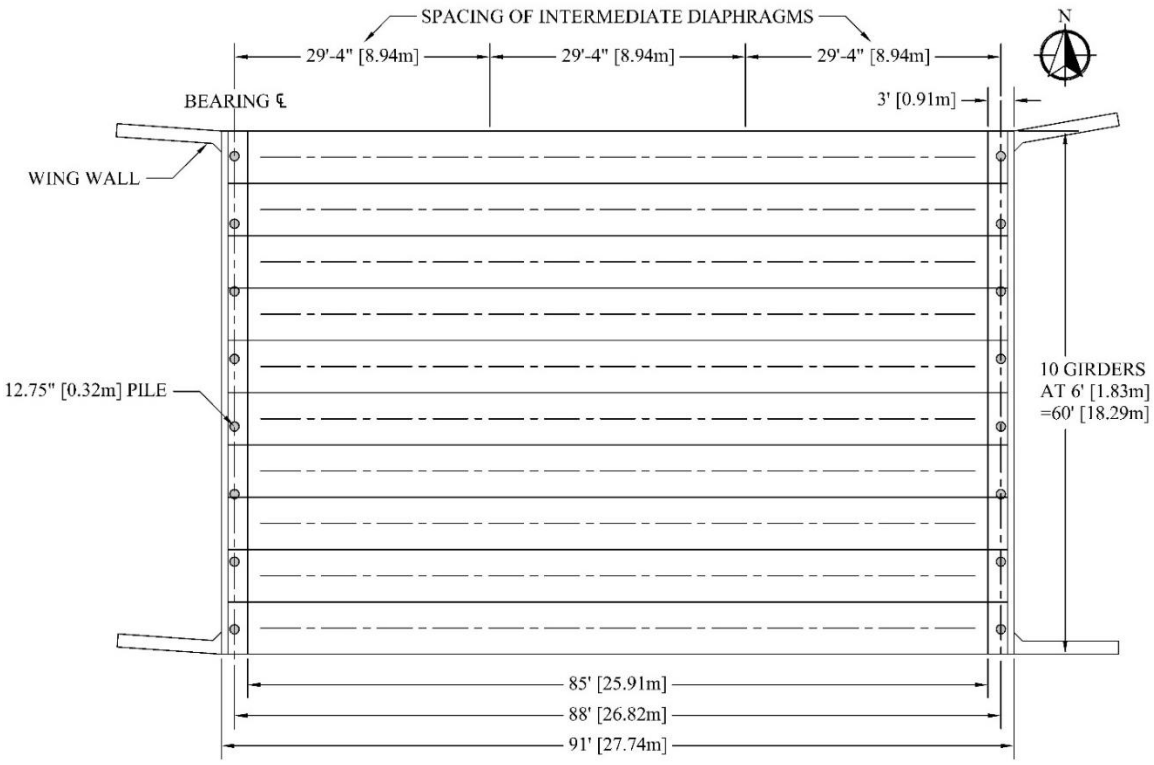

Figure 14. Plan view of the Nibley Bridge.

Other elements of the bridge include the parapets, wing wall, and intermediate diaphragms. The parapets are on each side of the bridge and act as guardrails for pedestrians and traffic barriers for errant vehicles. The parapets are 3.5 feet (1.07 m) high and tie directly into the exterior girders. The wing walls are adjacent to the abutments on each end of the bridge and are 11.58 feet (3.53 m) tall. The wing walls extend 12 feet (3.66 m) back from the abutments and are 1.5 feet (0.46 m) thick.

Finally, two intermediate diaphragms are located at 29.33 feet (8.94 m) from each end of the bridge. The diaphragms are constructed of MC18x42.7 steel channels that span between the girder webs, except for the space between Girders 3 and 4. The space between Girders 3 and 4 uses a concrete diaphragm in order to accommodate a steel pipe that runs longitudinally underneath the bridge (see Figure 15).

Figure 15. Underside of the Nibley Bridge.

3.2 Instrumentation Description

For this study, the Nibley Bridge was instrumented with eight ST350 Model strain transducers (See Figure 16). The strain transducers were attached to the exterior of the bridge using Loctite 410 adhesive and brackets. Once attached, the sensors were used to monitor the strain induced by loading the bridge. Sensors and recording equipment for this research were fabricated by Bridge Diagnostics Inc.

Figure 16. Strain Transducer Attached to the Bridge (Pickett 2017).

Once attached to the bridge, the strain transducers allowed monitoring of the longitudinal strain in eight of the ten girders. All sensors were attached at mid-span of the girders and were aligned parallel with the longitudinal axis of the girder. Strain values in unmonitored girders were assumed to have strains similar to those in monitored girders at mirrored points due to the symmetry of the bridge.

Initially, six sensors were placed at the center of the bottom flange of Girders 1 through 6, with the two additional sensors located near the centroid of Girders 1 and 2 (See Figure 17). Upon review of the data, the two sensors located at the neutral axis (B1987 and B1979) were recording values near zero. The gauge located at the bottom of Girder 1 also recorded near-zero strains. It was decided to move these three sensors to maximize the study. In September 2017, the sensors attached to the girder centroids and to the bottom of Girder 1 were moved to the center of the bottom flange of Girders 7, 8, and 9 in order to monitor those girders as well (See Figure 18).

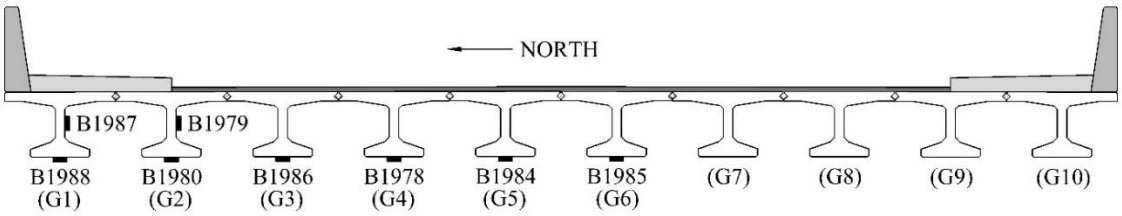

Figure 17. Initial Locations of the Strain Transducers.

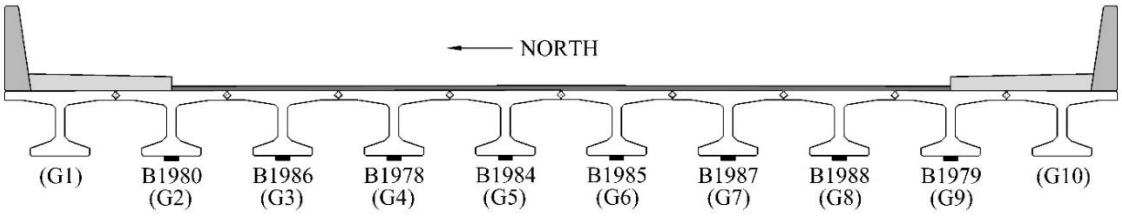

Figure 18. Adjusted Locations of the Strain Transducers.

3.3 Data Collection and Processing

3.3.1 Data Collection

Once the strain transducers were attached to the bridge, traffic over the bridge was monitored over several weeks. Because much of the bridge traffic is destined for the nearby high school, data was recorded during both the summer (July 29-August 11) and the winter (January 7-January 27) in order to determine strain levels while school was both in and out of session.

This research focused on strain "events". Each loading event was characterized by a noticeable increase in strain, followed by a return in strain to near at-rest levels (i.e. approximately 0 με) (See Figure 19). Positive strain values corresponded to tension in the girder.

Typically, events were caused by traffic driving over the bridge. As a vehicle drove over the bridge, the strain at the bottom of each girder would increase until the vehicle was positioned to cause the maximum moment in the girders. It would then decrease as the vehicle drove off the bridge, at which point the strains returned to measure at-rest levels.

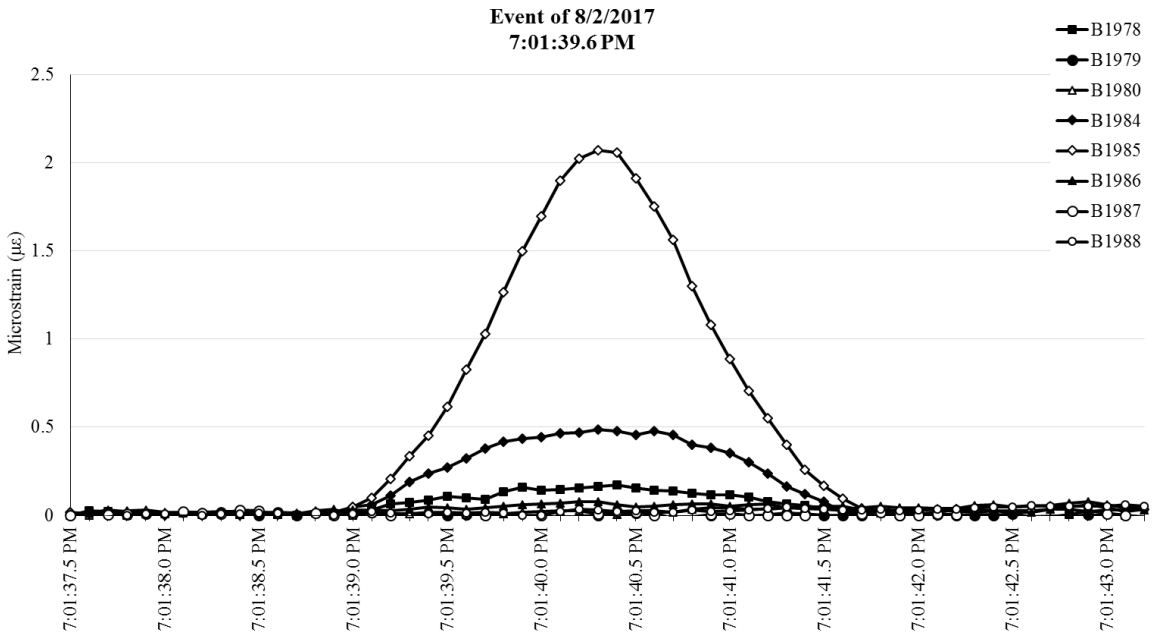

Figure 19. A typical strain event.

Data was recorded using Bridge Diagnostics Inc.'s STS Monitor software (Figure 20). While the STS Monitor software has the ability to record individual strain events, it was discovered that the software failed to record all of the strain events that occurred on the bridge. Instead, the STS Monitor software was set to continuously capture strain data from each sensor every 0.1 seconds. This data was recorded in technical data management streaming (TDMS) files that were saved to the datalogger. An add-in allowed the TDMS files to be opened and manipulated in Microsoft Excel.

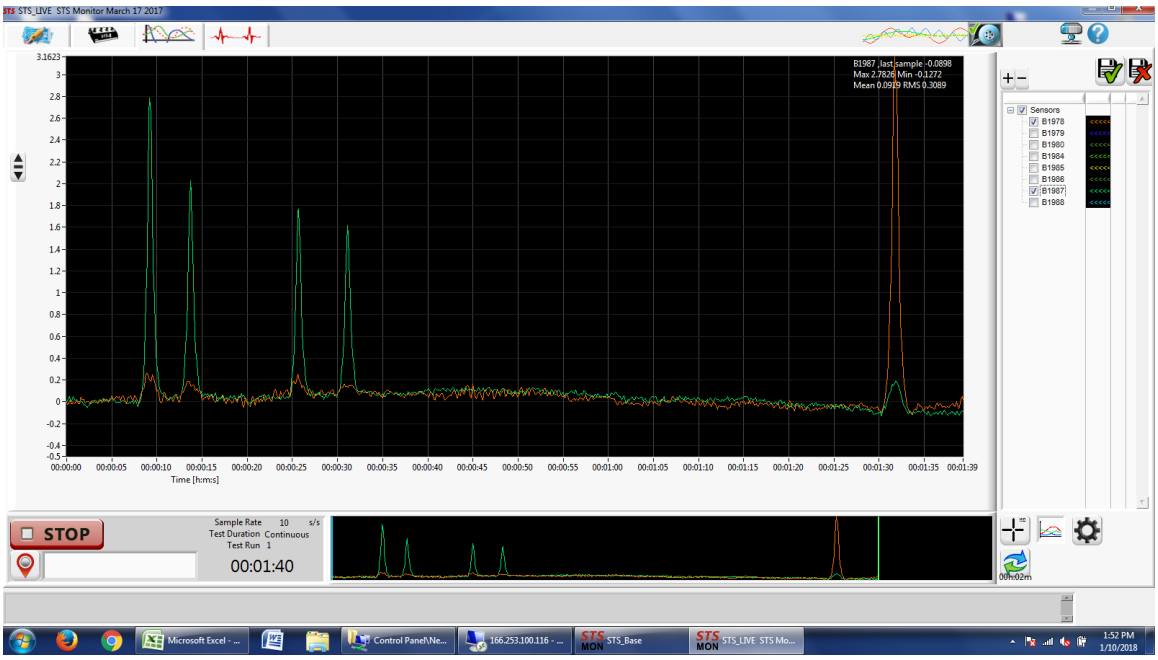

Figure 20. Screenshot of STS Monitor recording data.

3.3.2 Data Processing

The focus of the research called for the examination of individual strain events; however, the TDMS files were recorded as 4-hour-long blocks of continuous data. To avoid combing through hundreds of hours of data to manually identify strain events, two programs were written in Visual Basic for Applications (VBA) that would extract data about individual strain events and then compile the strain data into a consolidated Excel workbook.

The first VBA program identified sections in the continuous data where one or more sensors suddenly experienced a rapid increase in strain. The program identified rapid increases in strain by finding the value of each data point minus the minimum strain value from the previous 6.0 seconds for each sensor. If the difference between the two values was greater than 1.0 με, the program would copy all of the strain data from 3

seconds before the rise in strain began to 3 seconds after the strain values began to lower back to at-rest values (See Figure 21). It will also copy the 6.0-second minimum data. Finally, the VBA program would paste the copied information into a separate Excel workbook, save it, and continue scanning the TDMS files for further events.

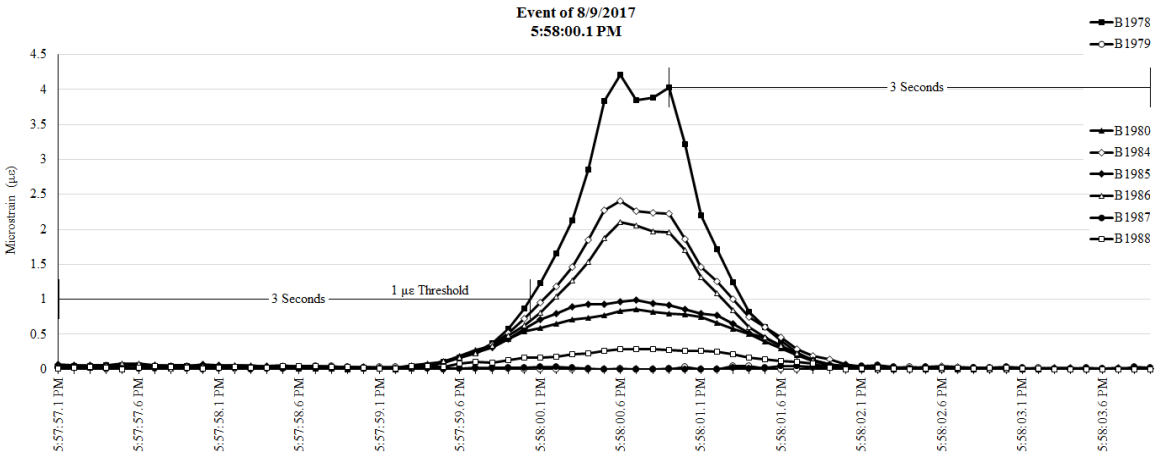

Figure 21. Annoted graph of a strain event showing the 1.0 με threshold as well as start and cutoff times.

Strain events that never crossed the 1.0 με threshold were considered negligible and were ignored by the event extraction program. A 1.0 με threshold was chosen after higher thresholds were shown to exclude a percentage of light motor vehicles. A trial-and-error approach was used to determine that the majority of vehicles caused greater than 1.0 με in bridge girders.

Once each strain event had been extracted into its own Excel workbook, the second VBA program would open the workbook for each event, read the maximum strain values for every sensor, and compile the maximum strain data values into a consolidated Excel workbook. The comprehensive workbook was then used as a basis for the conclusions found in Section 3.4. Copies of both of the VBA programs can be found in Appendix A.

It is important to note that not every vehicle produced a strain event. For example, pedestrians, bicycles, motorcycles, and other light loading might produce a strain event below the 1.0 με threshold. Likewise, some vehicles might produce more than one strain event. It was observed that semis hauling multiple trailers could have multiple, separate events. Finally, if two vehicles were on the bridge at the same time (either in separate lanes or one behind another) and strain levels did not return to a near-zero level between the two events, the combined response of the two vehicles would create a single strain event.

3.3.3 Precautions Against Data Errors

Several safeguards were implemented to make certain that the VBA programs accurately read and extracted the event data. These safeguards included precautions against drift, counting the same event twice, consecutive strain events, noise, and failure to return to at-rest strain levels. To minimize the effect that these potential errors had on the results, sections of code were included in the two programs to address each problem.

3.3.3.1 Drift

Because the BDI sensors were initially intended for short-term recording, they experienced a certain amount of drift throughout the day. The STS Monitoring software was designed to zero out the strain in the sensors when recording begins. As the time passes, changes in temperature and other environmental factors can cause the at-rest value of strain in the sensors to vary, even when there was no applied load on the bridge. For example, sensors that began by recording 0 με in the morning might be reading 30 με in the heat of the day and -20 με at night.

To compensate for drift, the strain compilation program would subtract the 6-second minimum value from the maximum strain value recorded by each sensor. This would transform the event on the left in Figure 22 to the event on the right. The VBA program would then copy the adjusted maximum value to the consolidated workbook as the maximum event strain.

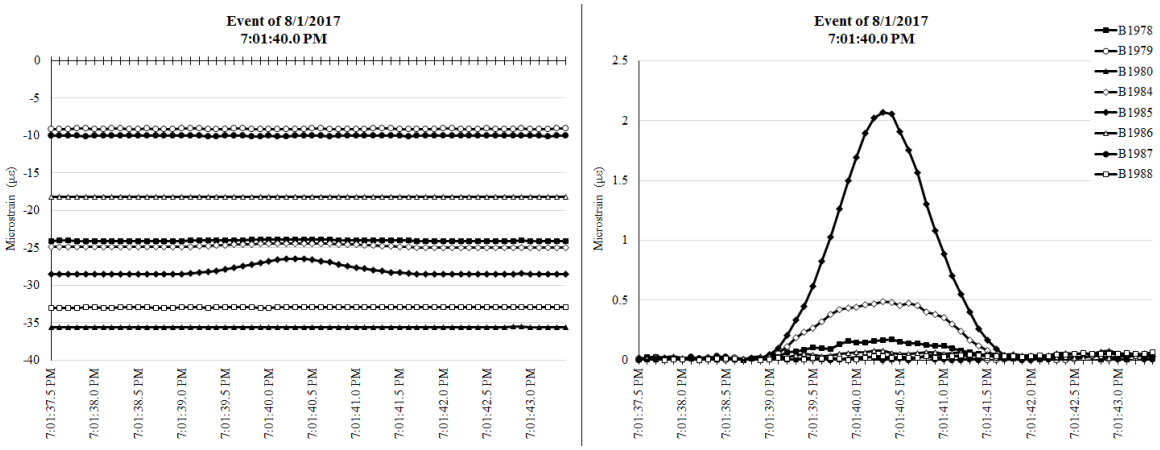

Figure 22. Uncorrected and corrected drift for the same event.

3.3.3.2 Multiple Event Triggers

In several cases, the same strain event would have multiple points that could trigger the program to copy it to a new workbook. For example, every point in the shaded area in Figure 23 could independently trigger the event extraction program to create a separate event. This would result in nineteen separate Excel files of the same event.

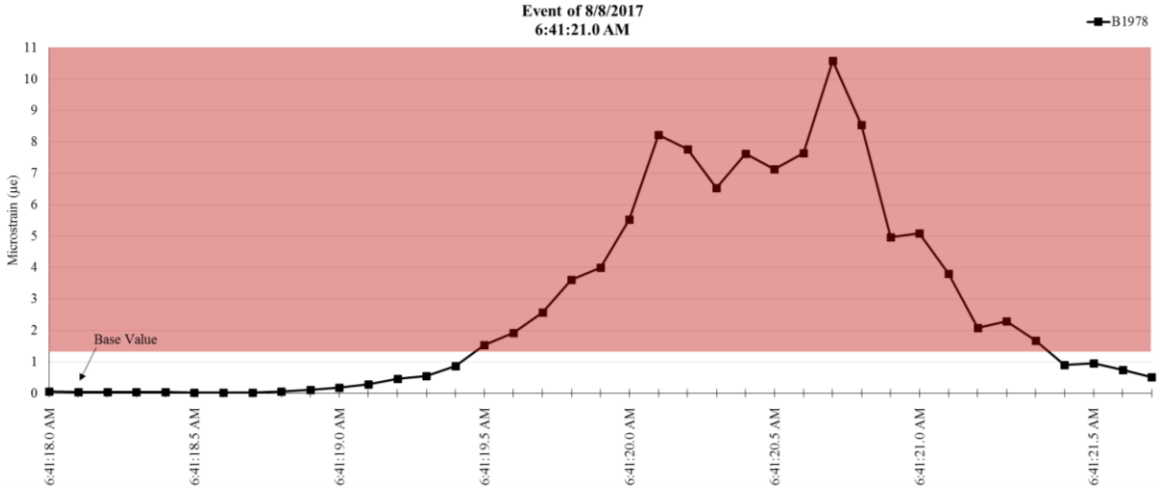

Figure 23. Multiple triggers for the same strain event.

In order to force the program to create only one workbook for each event, a segment of code was used that forced the program to ignore additional event triggers for 0.5 seconds after an event had begun. A 0.5-second window was chosen because, for a vehicle to cross the bridge in one second, it would be traveling 140.4 mph (113.0 kph) in a 25 mph (40.2 kph) speed zone. Therefore, a vehicle is unlikely to cross the entire bridge (i.e. create a complete strain event) in a single second. Any additional vehicles coming onto the bridge during the one-second window would end up contributing to the original event instead of creating a new, separate event.

Additionally, if an event had been triggered within the previous ten seconds, the program checked whether the program was a large strain event. It did so by subtracting a base value from the strain value that created the new trigger. The program would then check to see whether the difference was reasonably close to the 1.0 με threshold. If not, the new trigger was ignored (i.e. for the event shown in Figure 23, the event is initially triggered at 6:41:19.6 AM. One second later, the strain is at 7.64 με. 7.64 με is not

reasonably close to 1.0 με, so the program determines that it is a large strain event and ignores the additional trigger).

3.3.3.3 Consecutive Strain Events

Occasionally, multiple vehicles drove over the bridge within a matter of seconds of each other. When this happened, it was challenging for the event extraction program to count the number of vehicles accurately. The event extraction and strain compilation programs were designed to work best when single vehicles drove over the bridge one at a time.

To demonstrate some of the issues involved with consecutive strain events, Figure 24 shows an example of when two vehicles crossed the bridge within a few seconds of each other. Each vehicle created its own event file, but each event file contained information from both events. The first vehicle's event file spanned from Point A to Point D, while the second vehicle's event file spanned from Point B to Point E.

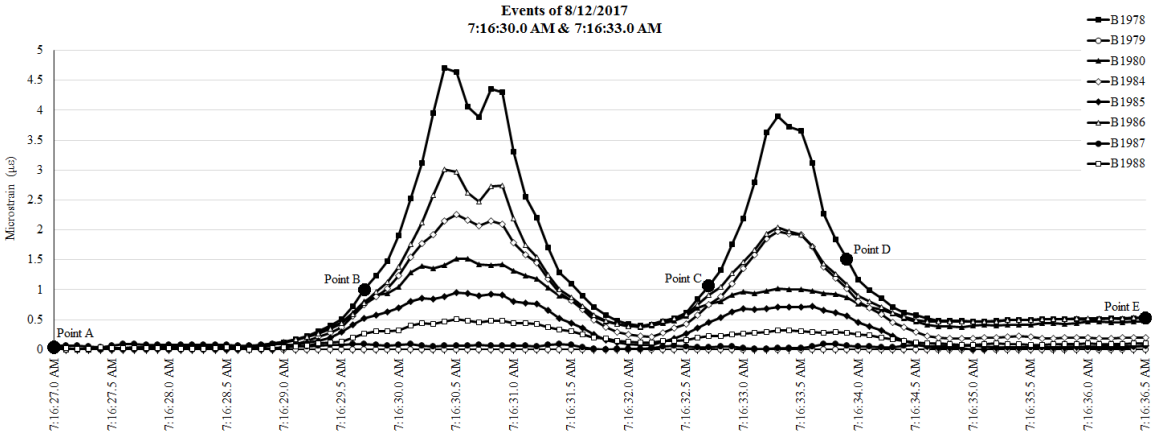

Figure 24. Consecutive strain events.

The above scenario created several problems. The event extraction program originally detected events by comparing each data point against the data point exactly 3.0

seconds before it. For the second event in Figure 24, this was the difference between Point C and Point B. Because the difference between these two points is only 0.06 με, the original event extraction program would not have detected the second event. To address this, the program was modified to subtract the minimum value from the last 6.0 seconds from Point C instead (a difference of 1.07 με).

Currently, the strain compilation program calculates the maximum strain in each file by subtracting the 6.0-second minimum from the maximum strain. Initially, the strain compilation program took the maximum strain value from anywhere in the event file. This caused inaccurate readings, because when it read the file for the second strain event file (comprised of points between Point B and Point E), the program detected the maximum from the first strain event instead.

This issue was addressed by modifying the program to take the maximum strain value from after the 1.0 με threshold was crossed (the second event crosses the threshold at Point C). However, if the second event were larger than the first, the program could still use the maximum strain from the second event when calculating the maximum strain of the first event (which contains data from Point A to Point D). This remains a known issue with the strain compilation program.

If a vehicle stopped on the bridge, it also created problems. For example, in Figure 25 a car stopped on the bridge to pick up pedestrians. Then, at 2:44:28 pm, a second car crossed the bridge. Because the strain compilation program only considers data from up to six seconds before the start of the event, the maximum strain for the second vehicle stopped did not include the initial strain caused by the stopped vehicle.

Instead, it only included the additional strain caused by the second vehicle driving onto the bridge. The strain compilation program faced similar problems if heavy traffic at the nearby stoplight caused vehicles to back up onto the bridge.

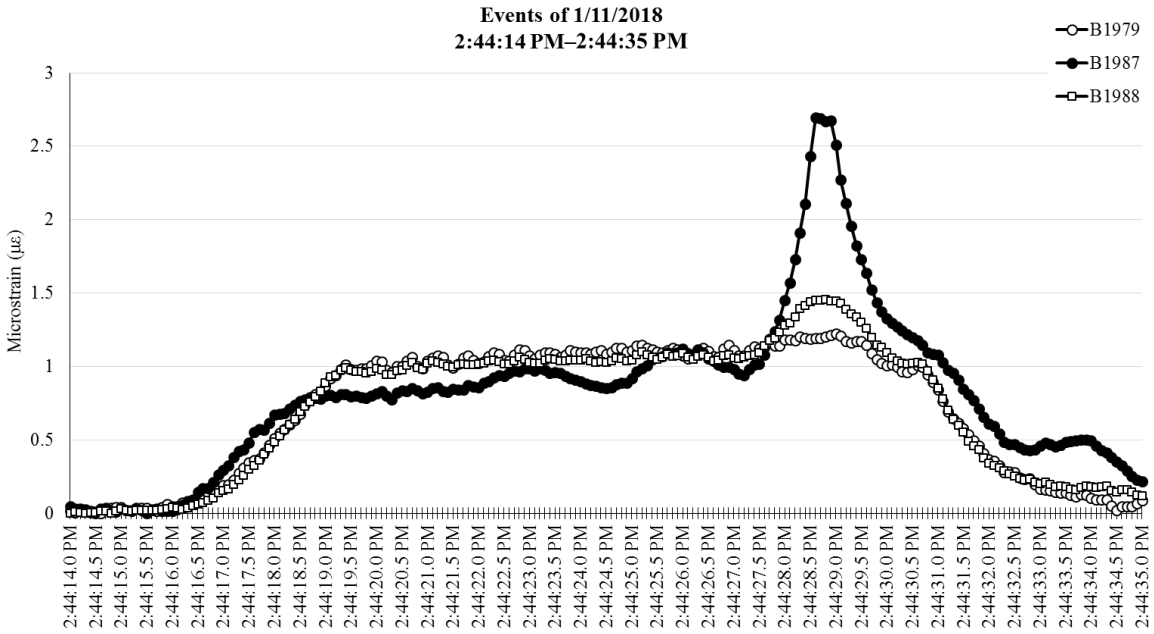

Figure 25. Strain data showing a car stopped on the Nibley Bridge.

3.3.3.4 Noise

All sensors experience some level of variation in their data called noise. Testing the BDI sensors revealed that the strain transducers experienced variation between 0.01–0.15 με while the bridge was unloaded. In order to avoid picking up sensor noise as actual strain events, a 1.0 με trigger threshold was selected for this project. It was observed that the majority of vehicles caused a strain greater than 1.0 με. Any sudden increases in strain that were did not exceed 1.0 με were ignored by the event extraction program. Setting the threshold at 1.0 με also helped combat drift, which could trigger a false event if the threshold were to set lower.

3.3.3.5 Failure to Return to At-Rest Strain Levels

Occasionally, the program would fail to return to normal strain levels at the end of the event. While the exact cause for this is unknown, it could potentially be related to vehicles parking on the bridge. When the program did not return to at-rest levels of strain, the event extraction program was set to extract a maximum of 30 seconds of data. This prevented a single event file from lasting for several minutes and containing multiple events.

3.3.4 Accuracy Verification of Results

To ascertain that the developed event extraction program could accurately locate and extract the strain events caused by traffic, video monitoring was set up at the Nibley Bridge (See Figure 26). The video camera chosen, the GoPro Hero5 Black, recorded video of vehicles driving over the bridge over the period of an hour. The footage of the traffic was then reviewed and compared against processed strain data for the same time period for validation purposes.

Figure 26. Still from recorded video showing a vehicle crossing the Nibley Bridge.

Video was recorded between 2:00–3:00 P.M. on 11 January 2018. This time was chosen because it allowed for video capture during two different levels of traffic flow. During most of the day, traffic on the bridge is comparatively low. The exception is just before school is about to start at 8:00 A.M. and right after school lets out at 2:45 P.M, when for a brief period the traffic volume is comparatively high. For the purposes of comparison, data was analyzed separately from 2:00–2:30 P.M. and from 2:30–3:00 P.M.

Between 2:00–2:30 P.M., the video recording showed that 71 vehicles crossed the Nibley Bridge. The event extraction program was able to detect and extract 67 events. The four omitted events involved vehicles in opposite lanes crossing the bridge within 0.5 seconds of each other; per the definition of a strain event in Section 3.3.2, the strains caused by these vehicles were treated as a single strain event. Under these conditions, 94.4% of vehicles triggered strain events, and 100% of vehicle strain data was accurately extracted.

Between 2:30–3:00 P.M., 311 vehicles crossed the Nibley Bridge. The event extraction program was able to detect and extract 261 strain events. The program also extracted a single event twice due to an exceptionally high amount of noise that coincided with the event. Due to the high volume of traffic, the specific causes of each omitted event were unable to be determined. However, the event extraction program was successfully able to detect and extract strain data files for 83.7% of vehicles.

At approximately 2:48 P.M., the traffic volume was so high that traffic backed up on the bridge in the north lane for approximately 1.7 minutes. Because the event extraction program only extracts data from the previous six seconds, the strain

compilation program only considered additional strain caused by new vehicles driving onto the bridge, not the total strain in the bridge girders, during this window. Additionally, most vehicles followed each other so closely that the heaviest vehicles (i.e. school buses) appeared in multiple events; the strains caused by the heaviest vehicles were counted multiple times by the strain compilation program.

Comparing the event extraction program against the bridge traffic video footage demonstrates that the event extraction program is able to accurately detect and extract events from the raw TDMS files. While the program had difficulty distinguishing between vehicles during levels of high traffic, it was still able to extract events for 83.7% of all vehicles. During periods of low traffic, the program was able to extract 100% of vehicle strain data. Because the bridge experiences high volume of traffic for no more than 5% of the day, the event extraction and strain compilation programs were deemed adequate for the purposes of this research.

Finally, because strain events with exceptionally high levels of strain were of the greatest interest to this research, every high-level strain event recorded over the five weeks of monitoring was hand-checked prior to drawing the conclusions in Section 3.4. This was done by opening the individual file for each event and plotting a graph of strain over time to verify that the maximum strains had been accurately extracted. If necessary, the raw, continuous TDMS file was consulted to give context to the data.

3.4 Short-Term Strain Monitoring Results

3.4.1 Strain Summations

It was discovered that Girders 4 (Sensor B1978) and 7 (Sensor B1987) experienced the highest strains. These girders regularly experienced strains 185% to 250% higher than strains in adjacent girders. As the magnitude of the strains in Girders 4 and 7 increased, the difference in strains between adjacent girders also increased.

Girders 3 (Sensor B1985), 5 (Sensor 1984), 6 (Sensor B1985), and 8 (Girder 1988) experienced the second-highest strains. Each of Girders 3-8 is directly beneath the traffic lanes. The lowest strains at the bottom of a girder were recorded when Sensor B1988 was placed beneath Girder 1, an exterior girder. When the sensors were placed at the neutral axis of the girders, they experienced little to no strains. Figure 27 shows a typical strain event for a single vehicle in the north (left) and south (right) lanes.

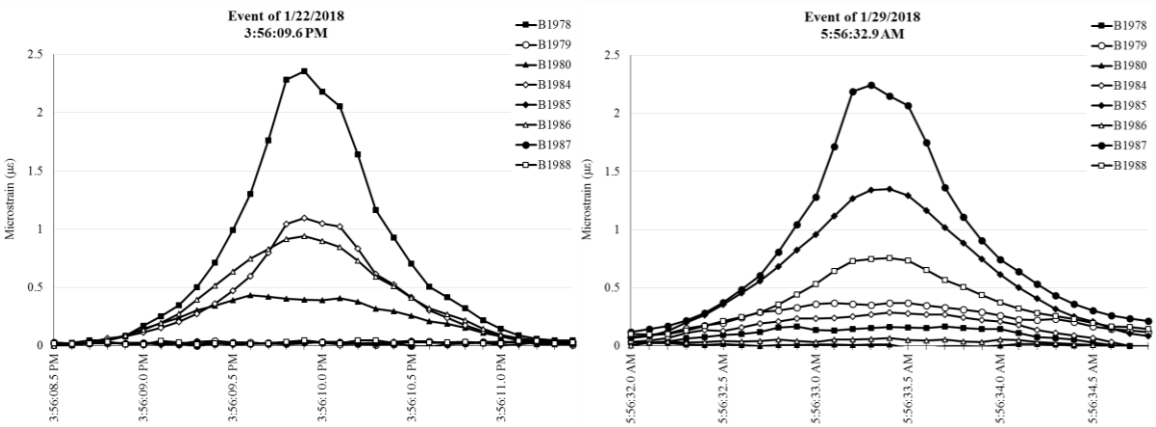

Figure 27. Typical strain event for the north (left) and south (right) lanes.

Between January 14 and February 3 (three weeks), the strain transducers recorded 35,843 events. In the two-week period from July 30-August 12, the strain transducers recorded 19,053 events. However, during the summer Girder 7 was not monitored, which

meant that several vehicles likely crossed the Nibley Bridge in the south lane without triggering an event.

Analyzing the wintertime data shows that 9.0% of events recorded in the south lane did not trigger an event in Girder 6 (the only south-lane girder monitored during the summertime). Assuming that this ratio is constant year-round suggests that an additional 1052 events in the south lane were not recorded by the datalogger during the summer. This gives an estimated 20,105 total events during the summer recording period.

Graphs showing the magnitude of the recorded strains are shown in Figure 28 (summer) and Figure 29 (winter). Both figures show the largest strain that was recorded by any sensor for each event. Each figure subdivides the recorded strains into individual weeks.

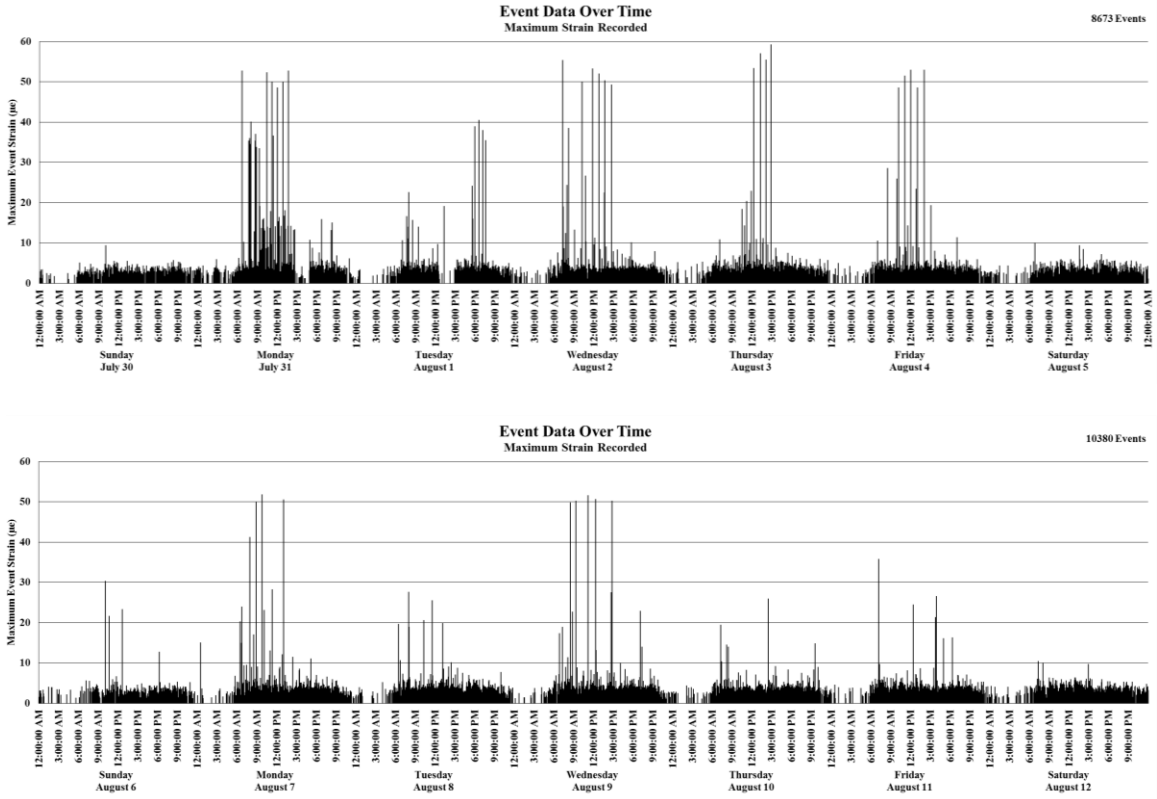

Figure 28. Graph of strain magnitudes between July 30 and August 12.

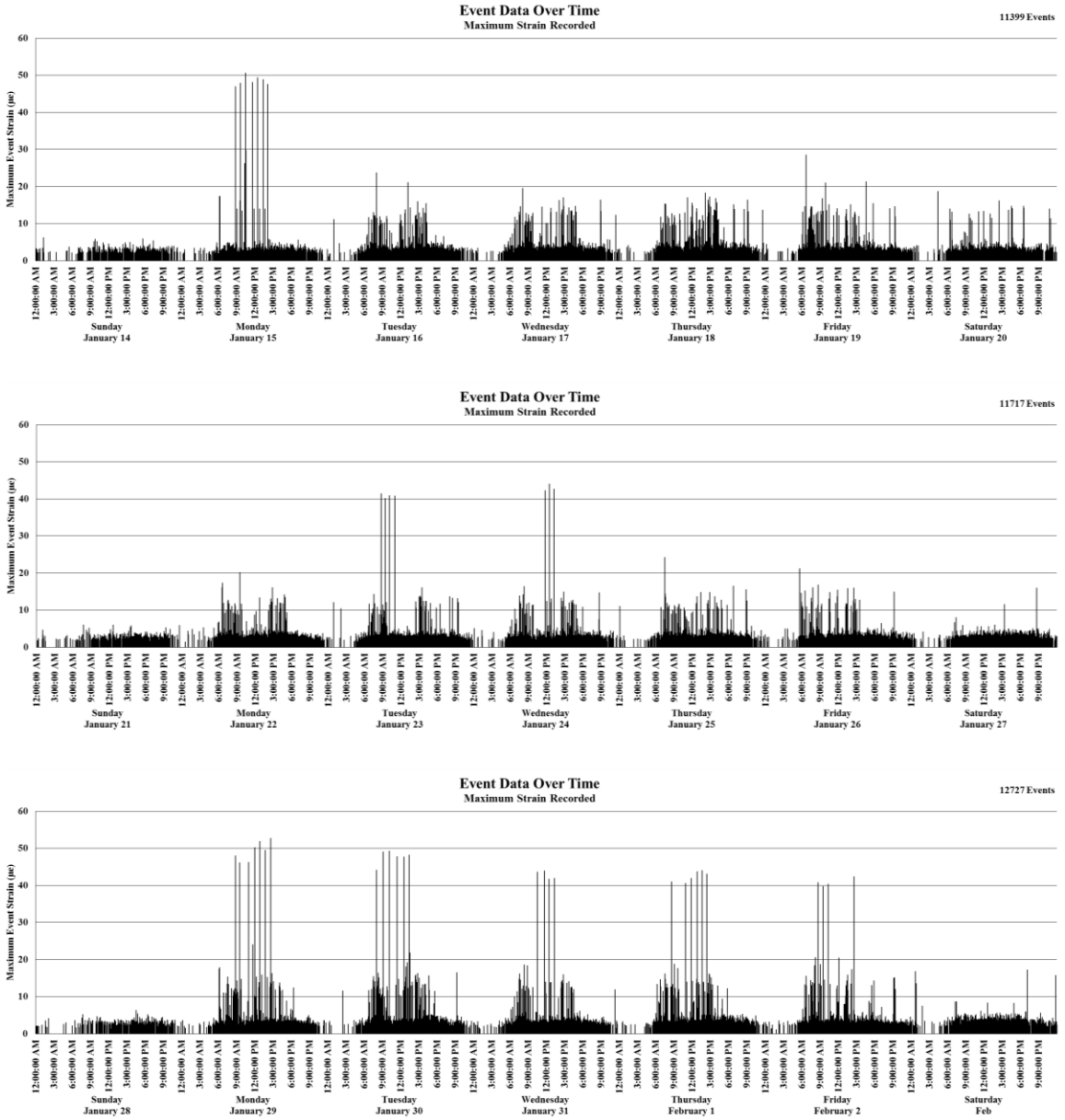

Figure 29. Graph of strain magnitudes between January 14 and February 3.

Figures 28 and 29 show that the traffic flow is relatively consistent throughout the day and that the majority of strains do not exceed 5.0 με. Traffic flow decreases at night, with comparatively little traffic between the hours of midnight and 5:00 am. There was a

noticeable rise in strains in the 10-20 με range during the winter; this is attributable to additional school buses crossing the bridge (note that in Figure 29 there are clusters of strain in the 10-20 με range when buses would be arriving at and leaving the school). There are also several strains significantly higher than the rest; these strains will be discussed further below.

During the summer, the Nibley Bridge had an annual average daily traffic of 1436 vehicles/day. During the winter, the bridge had an annual average daily traffic of 1706 vehicles/day. This corresponds to an 18.85% increase in traffic while school is in session.

The majority of events were observed to last about 3.25 seconds. This appears reasonable, as it corresponds to a vehicle speed of 22.6 mph (26.4 kph). The speed limit on the bridge is 25 mph (40.2 kph). However, some events lasted as long as thirteen seconds. These events mostly correspond with larger vehicles accelerating after turning off the nearby intersection or roundabout.

3.4.2 Distribution of Strains throughout the Day

Further analysis of Figures 28 and 29 resulted in Figures 30 and 31. These figures show the distribution of strains throughout the day, regardless of their magnitude. During the summer, traffic flow was relatively consistent throughout the day and corresponded with anticipated trends relating to traffic patterns. For example, the largest percentage of traffic occurred between 5:00 pm-6:00 pm, which corresponds to employees returning home from work.

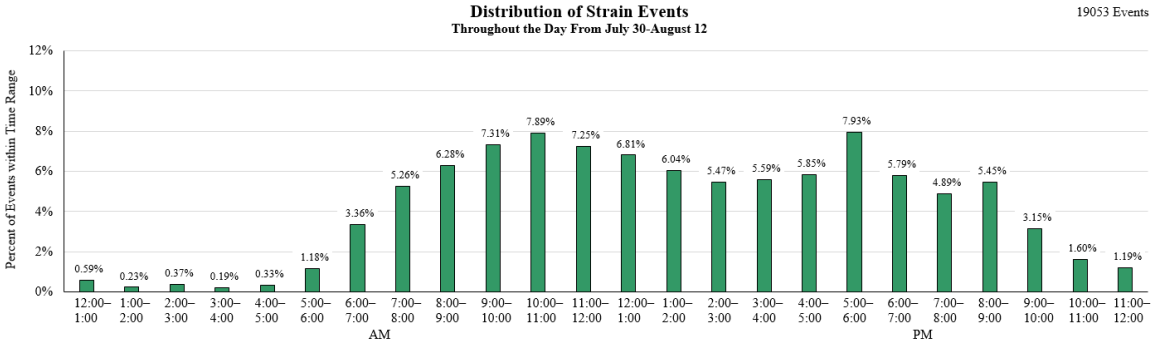

Figure 30. Distribution of strains throughout the day between July 30-August 12.

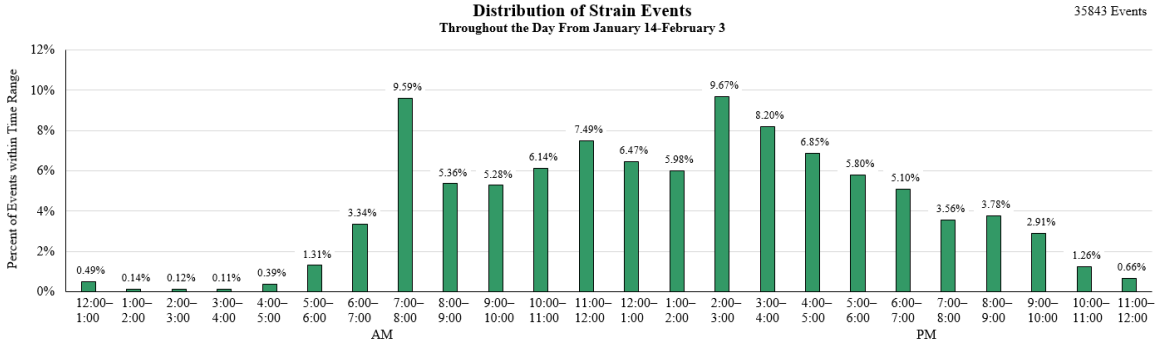

Figure 31. Distribution of strains throughout the day between January 14 and February 3.

During the winter, the highest traffic volumes corresponded with school beginning at 8:00 am and letting out at 2:45 pm. On days when school was in session, an average of 480 vehicles crossed the bridge in the half hour before school started (7.69% of the day's traffic). An average of 469 vehicles crossed the bridge within half an hour of school letting out (7.50% of the day's traffic). From this data, it can be that concluded that traffic flow was more concentrated while school is in session, resulting in multiple vehicles using the bridge at once and larger strains in the bridge girders.

3.4.3 Analysis of Large Strain Events

Figures 28 and 29 also reveal that several large vehicles crossed the bridge on a semi-regular basis. Table 2 shows the largest strain values recorded for in each girder for both monitoring periods.

Table 2. Largest recorded strain values for each girder.

Largest Recorded Strain (με)	**Girder Number & Associated Sensor**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
	B1988	**B1980**	**B1986**	**B1978**	**B1984**	**B1985**	**B1987**	**B1988**	**B1979**	**N/A**
Summer	7.24	18.55	30.87	59.26	40.46	40.46	N/A	N/A	N/A	N/A
Winter	N/A	15.94	28.22	52.78	26.70	20.35	26.31	13.66	10.72	N/A
Overall	7.24	18.55	30.87	59.26	40.46	40.46	26.31	13.66	10.72	N/A

While the largest values recorded in Table 2 tend to not vary significantly between the summer and winter, larger strains occurred during the summer in every monitored girder. This is likely due to the increased use of heavy construction equipment during the summer months.

Table 2 also shows that, to an extent, the assumption of symmetry in strains between girders is valid. The largest exception to this assumption is Girders 4 and 7. When these two girders were monitored simultaneously during the winter, the difference between the recorded maximums is 26.47 με. Potential reasons why the strains in Girder 4 are so much higher are discussed further below.

While Table 2 tabulates the largest strain value seen in each girder, Table 3 presents the largest twenty strain values caused in any girder. Each strain value has been checked for accuracy by plotting a graph of the event to see whether it appeared accurate. It is important to note that every strain in Table 3 occurred in Girder 4 (Sensor B1978).

Table 3. Twenty largest recorded strain values for all girders.

Time of Vehicle Crossing	Microstrain Caused by Vehicle
3 Aug 2017 — 2:47:58 PM	59.26 με
3 Aug 2017 — 1:15:25 PM	57.01 με
3 Aug 2017 — 2:02:33 PM	55.42 με
2 Aug 2017 — 7:15:42 AM	55.31 με
3 Aug 2017 — 12:12:32 PM	53.38 με
2 Aug 2017 — 11:46:29 AM	53.30 με
4 Aug 2017 — 11:59:41 AM	52.98 με
4 Aug 2017 — 2:03:47 PM	52.91 με
29 Jan 2018 — 2:38:59 PM	52.78 με
31 Jul 2017 — 6:40:37 AM	52.78 με
31 Jul 2017 — 1:40:55 PM	52.76 με
31 Jul 2017 — 10:23:31 AM	52.27 με
2 Aug 2017 — 12:43:30 PM	51.96 με
29 Jan 2018 — 12:51:29 PM	51.89 με
7 Aug 2017 — 9:48:30 AM	51.78 με
9 Aug 2017 — 11:07:16 AM	51.57 με
4 Aug 2017 — 11:07:10 AM	51.50 με
15 Jan 2018 — 10:29:46 AM	50.66 με
9 Aug 2017 — 12:17:55 PM	50.62 με
7 Aug 2017 — 1:01:39 PM	50.52 με

Because Girder 4 falls directly underneath the north lane, it experiences the largest strains of any girder due to westbound traffic. However, its strains are exceedingly bigger than the largest strains in the corresponding south-lane girder. Four possible explanations are presented as to why Girder 4 experienced larger strains than other girders.

First, the strain transducer attached to Girder 4 (Sensor B1978) could need calibration. However, we know that Girder 7, the corresponding girder in the south lane, regularly experiences strains 185% to 250% higher than strains in its neighboring girders. This ratio is similar for the north lane as well, suggesting that Sensor B1978's readings may be correct.

Second, the traffic light to the west of the bridge provides vehicles an easy way to turn south onto Utah State Road 165. Because UT-165 has a posted speed limit of 55 mph (88.51 kph), large vehicles turning left need to use a traffic signal to give them time to make the turn and accelerate. The traffic light near the Nibley Bridge is the only signal for 1.19 miles (1.92 km) to the north and 0.75 miles (1.20 km) to the south. Because the Utah Department of Transportation is currently doing roadwork on UT-165 to the south of Ridgeline High School, there are multiple construction vehicles which use this intersection to turn south onto UT-165. Vehicles turning south would cross the north lane of the bridge, but not the south lane.

Third, the girders in the south lane may have a better joint connection than the girders in the north lane. If so, Girder 7 would distribute more load into its neighboring girders than Girder 4, resulting in higher comparative strains in Girder 4. A difference in the degree of composite behavior could potentially be caused by discrepancies in the way the girders were installed or damage in the connectors by excessive loading.

Fourth, trucks could be retrieving material from an area to the east of the bridge and transporting it to an area to the west of the bridge. If this is true, trucks crossing the bridge would be heavier when they crossing the north lane, and lighter when they returned using the south lane.

Referring back to Figures 28 and 29 shows that multiple large events often occurred in a single day. On other days, there were few, if any, large events. Reviewing Table 3 shows that many events with similar magnitudes of strain occurred on the same

day. Because of this, it is likely that the largest strains for each day were caused by the same vehicle.

This conclusion is supported by Figure 32, which plots two large events from July 31, 2017 (left) against two large events from January 29, 2018 (right). Even though the events on both sides produced similar magnitudes of strain in the bridge, they appear drastically different. Note that the two events shown for August 3 are similar in both shape and the time it took to cross the bridge, suggesting that they were caused by the same vehicle. The same applies to the two events from January 31. Incidentally, the vehicle on the left likely had four axles (because there are four peaks within the event), while the vehicle on the right likely only had two axles.

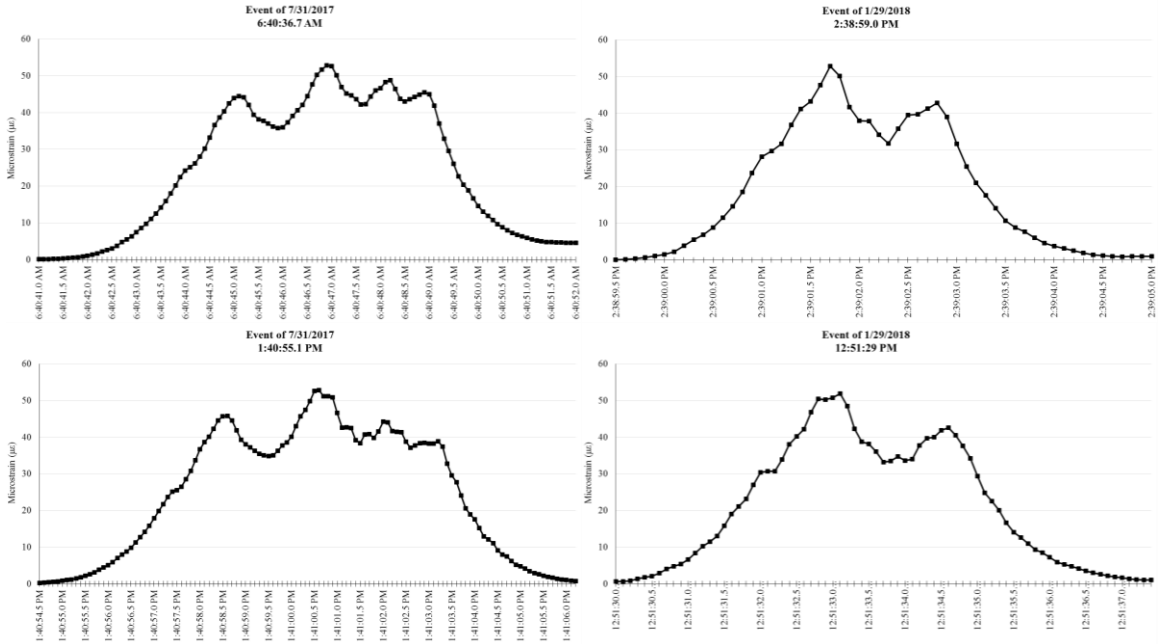

Figure 32. Comparison of large strain events between days.

The largest recorded strain event occurred at 2:47:58 PM on August 3, 2017 (See Figure 33). The vehicle took 8.3 seconds to cross the bridge and caused a strain of 59.26

με. While the exact type of vehicle that caused this strain event is unknown, analysis of Figure 33 shows that it was likely a double trailer dump truck with six axles crossing the north lane. Given the magnitude of strain it caused, it would have been carrying a heavy load, likely for construction purposes.

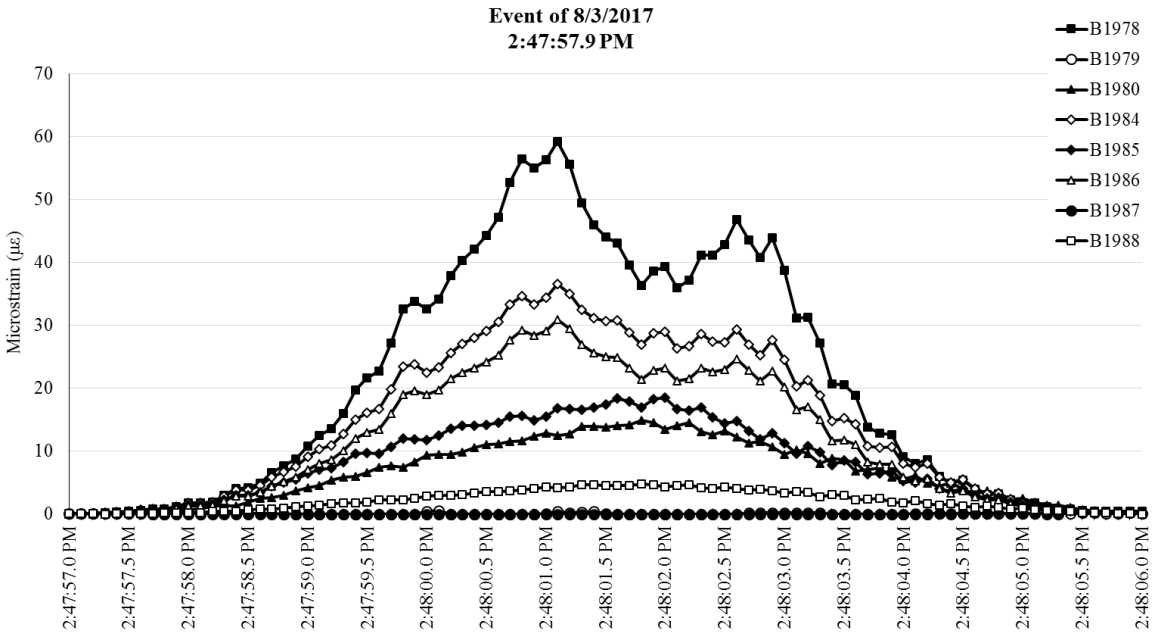

Figure 33. Largest recorded strain event.

3.4.4 Comparison of Large Strains with Design Loading

In order to compare strain magnitudes with a vehicle of a known weight, a live load test was conducted using an International 7700 dump truck. The axle configuration of the truck is shown in Figure 34; an image of the dump truck used is shown in Figure 35.

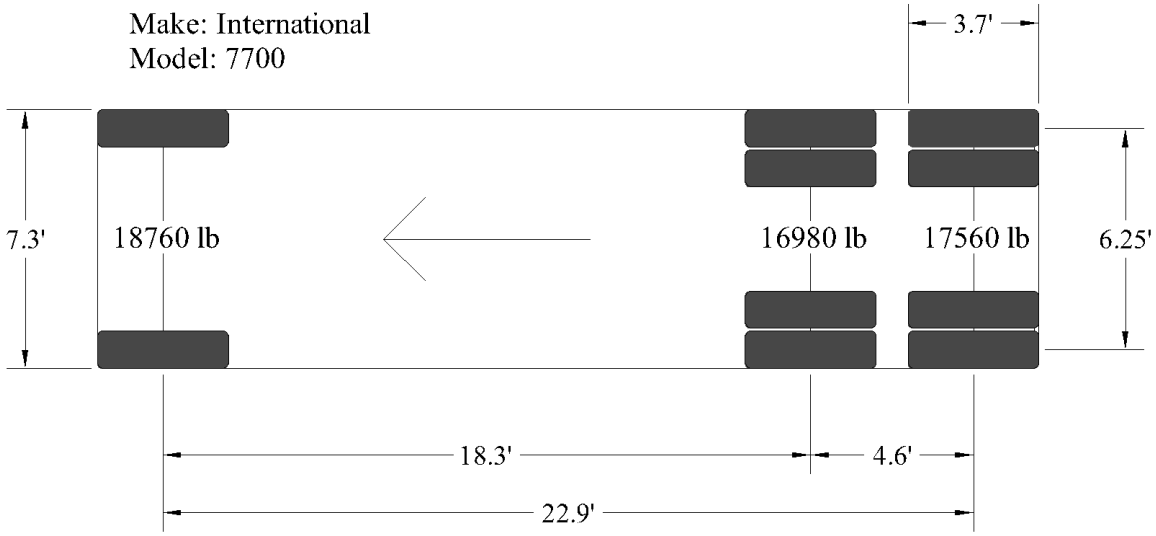

Figure 34. Axle configuration of the International 7700 dump truck.

Figure 35. Image of the International 7700 dump truck used.

The dump truck was driven over the bridge along five different load paths. Each load path placed the front-right tire of the dump truck directly over Girders 2, 3, 4, 5, and 6. These load paths were chosen because the test took place on July 18, 2017 and the corresponding girders were the only ones being monitored at the time (A load path over

Girder 1 was not tested, as the truck would have had to drive up onto the sidewalk). The dump truck drove along each load path twice, and the results were averaged for each load path.

The dump truck caused the largest strain in the bridge when it was positioned with its right tire directly over Girder 5, placing it approximately in the center of the bridge. When positioned there, the truck caused a maximum strain of 33.12 με in Girder 5, but only 13.14 με in Girder 4. When the truck remained in the north lane, it caused a maximum strain of approximately 30.29 με in Girder 4. A more detailed analysis and report on the live load test results will be published at a future date.

While a more detailed analysis of the live load test incorporating a finite element model will be conducted in the future, it can be shown using statics that the dump truck caused a maximum moment of approximately 412.57 kip-feet (559.37 kN-m) in the bridge. Assuming that the relationship between the maximum moment caused by a vehicle and the maximum strain induced in the bridge is linear, the strain caused by the dump truck can be used to approximate the strain caused by other types of loading.

For example, an HS-20 truck would create a moment of 604.61 kip-feet (819.74 kN-m) in the bridge. The ratio between the two moments is 1.465, meaning that an HS-20 truck should cause a maximum strain of 42.23 με in Girder 4 (30.29 με * 1.465 = 42.23 με) if it were driven across the north lane. The controlling HL-93 design loads (consisting of an HS-20 truck plus a lane load of 650 lbs/foot (9340.10 N/m)) would cause an approximate strain of 62.59 με in the bridge.

The calculated moments above assume a partial fixity of 90% at the ends of the bridge girders. However, the fixity of the bridge makes relatively little difference to the moment ratios. For example, the moment ratio between the HS-20 truck and the dump truck is 1.394 if the bridge if the completely pinned and 1.486 if the bridge is completely fixed. For the HL-93 design loads and the dump truck, these ratios are 2.067 if pinned and 2.066 if fixed.

Because the moments and strains caused by an HS-20 truck and the controlling HL-93 design loads are an integral part of bridge design, it is significant to note that the traffic at Nibley Bridge regularly exceeded the strains that the AASHTO truck loadings should cause. For example, if an HS-20 truck crossed the Nibley Bridge in the north lane, it could be expected to cause a strain of 42.23 με in Girder 4. During the five weeks of recording, 58 vehicles caused a strain greater than 42.23 με. This accounts for 0.18% of all vehicles that crossed the bridge in the north lane during this period.

The HL-93 design loads would be expected to induce a strain of 62.59 με in Girder 4. While no events ever exceeded 62.59 με, the maximum recorded strain in Girder 4 was 59.26 με. This comes within 5.61% of exceeding the design loads for the bridge. Considering the relatively short monitoring window of five weeks, it is safe to assume that the design loads could be periodically exceeded every year.

These findings are consistent with the studies on which the bridge design code is based. According to Nowak and Collins, the current bridge code is based off of surveys on highway loadings in sixteen different locations. In these surveys, approximately 0.003% of vehicles exceeded HL-93 loading for a 90-foot (27.43-meter) bridge. With

54,896 recorded events over 5 weeks, the Nibley Bridge could only expect to see a single event exceed the HL-93 loading.

It is to be expected that the Nibley Bridge did not see any events exceed the HL-93 loading. Because it is in a rural area, the Nibley Bridge sees less traffic than the highway loading surveys. The monitoring window was also smaller for this project than for the surveys used to create the bridge code. Thus, the Nibley Bridge can be expected to see similar results on a smaller scale than the highway surveys.

There is also the possibility that the data includes larger strains than would be considered normal for the Nibley Bridge. As noted, the Utah Department of Transportation was doing construction on a portion of UT-165 near the Nibley Bridge during the monitoring window. Because of this, an abnormally high number of construction vehicles could be skewing the data.

3.4.5 Vehicle Strain Comparisons

Using the video recorded by the GoPro video monitoring done in Section 3.3.4, strain events were associated with the different vehicle types that caused them. Table 4 shows strain data corresponding to each vehicle type. Figures 36-39 graphically show strain events associated with typical sedans, school buses, semis, and dump trucks. “Typical” events were found by calculating the median for the maximum strain values caused by each vehicle type, then choosing an event whose maximum value was close to the median value.

Table 4. Vehicle strain data.

Vehicle Type	Mean (με)	Median (με)	Standard Deviation (με)	Minimum (με)	Maximum (με)	# Samples
Sedans	2.417	2.282	0.551	1.099	4.529	100
Minivans	2.874	2.882	0.457	1.893	3.716	17
Jeeps	2.882	2.887	0.401	2.124	3.409	8
SUVs	3.029	3.054	0.540	2.034	4.269	29
Suburbans	3.249	3.237	0.504	2.322	4.654	32
Vans	3.315	3.237	0.404	2.912	3.833	6
Pickup Trucks	3.454	3.372	0.566	2.582	5.051	27
Delivery Trucks	5.733	5.882	0.556	5.000	6.168	4
Semi Truck	13.350	12.542	2.265	11.599	15.908	3
School Buses	13.729	13.649	1.949	11.212	16.581	11
Dump Trucks	39.218	42.621	7.808	30.286	44.748	3

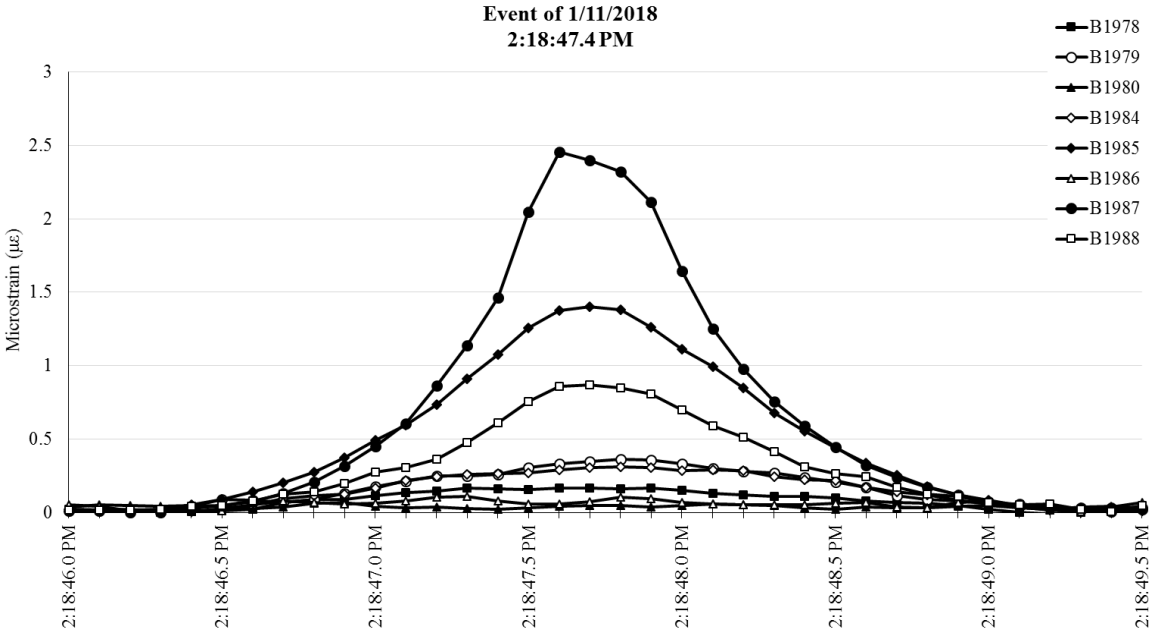

Figure 36. Strain event from a sedan.

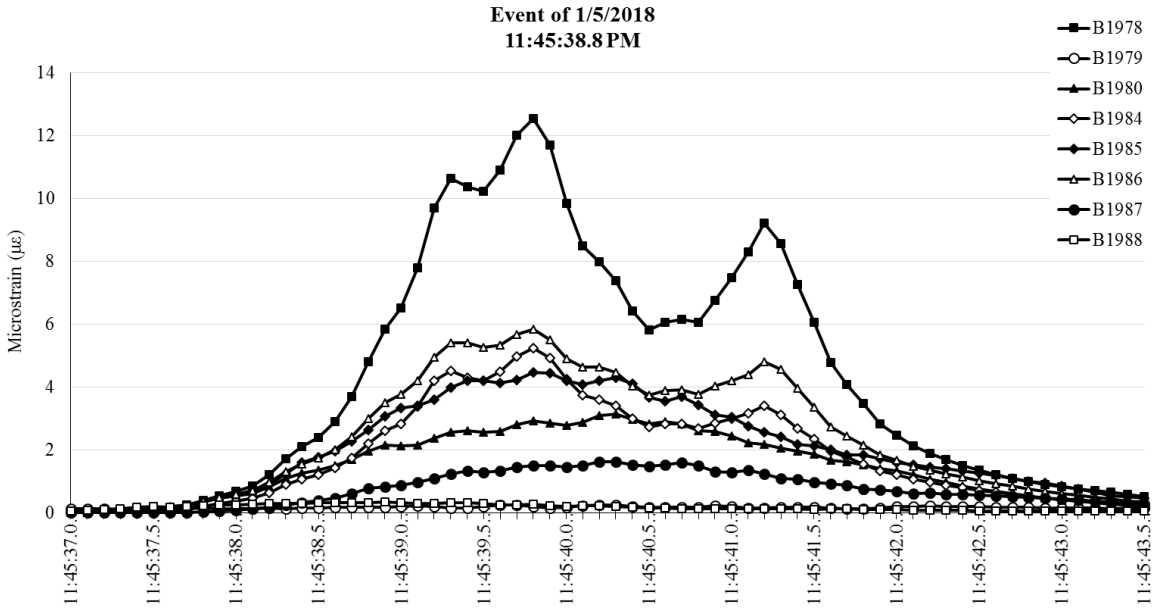

Figure 37. Strain event from a semi truck and trailer

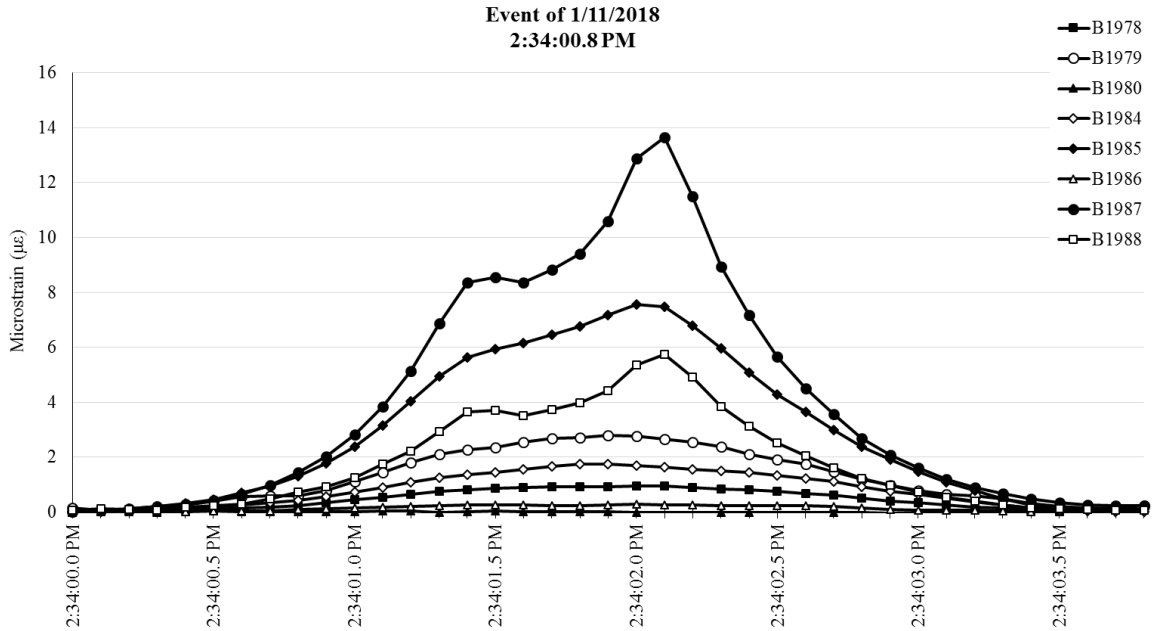

Figure 38. Strain event from a school bus.

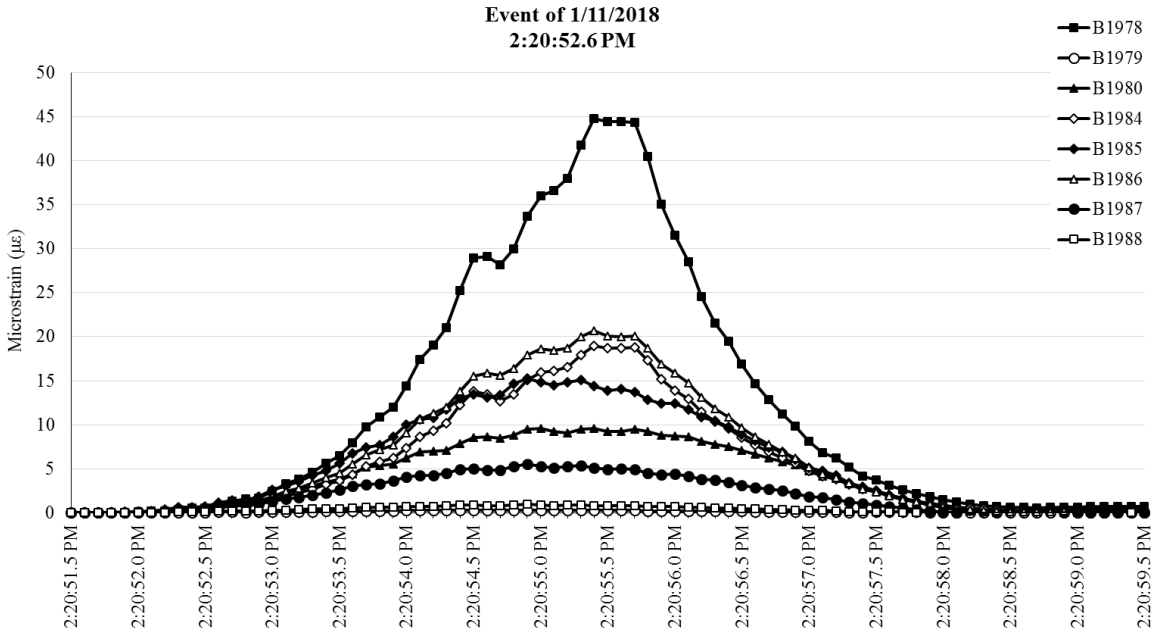

Figure 39. Strain event from a dump truck.

The data in Table 4 shows that there is little variation in the strains caused by smaller vehicles; sedans, minivans, jeeps, SUVs, suburbans, vans, and pickup trucks all had mean strains that fell between 2.42–3.45 με. Most school buses tended to cause a strain of 13.73 με, while semis caused a strain of 13.35 με. Dump trucks caused the highest strain of all, 39.22 με. However, it should be noted that the sample size for both semi trucks and dump trucks was limited. Given the variability in gross vehicle weight for both vehicle types, it is expected that the strain caused by each vehicle would vary greatly as well.

Figures 36, 37, 38, and 39 also demonstrate that as a vehicle's size and weight increases, it becomes possible to distinguish the strains caused by each vehicle axle. This is partially because the increased spacing between the axles prevents the axles from

effectively acting as a single force and partially because larger vehicles cross the bridge at a lower speed, allow for more points to be captured per axle.

3.4.6 Frequency Histograms

The data from all five weeks of recording was combined and analyzed to create frequency histograms of the strains in each girder. Figures 40-50 show these frequency histograms. The histograms have been divided into bins with a range of 1.0 με each. Each histogram uses this spacing between 0-10 με. After approximately 7 με, the bins level out and most bins afterwards account for less than 0.1% of the events.

After 10 με, the graphs contain a bin between 10-42.2 με. A strain of 42.2 με (corresponding with an HS-20 truck) has been included for comparison purposes. The final bin, extending from 42.2-59.3 με, recorded the percentage of events whose strains exceed that of an HS-20 truck.

It is important to note that each histogram shows the strain in the girder for every event where any sensor crossed the 1.0 με threshold. This is why the bins from 0-1 με contain the highest percent of events for every girder. If a light vehicle crosses the Nibley Bridge in the south lane, it will not cause noticeable strain in the girders on the north end of the bridge. However, it will still contribute to the histograms for the north girders. Thus, the high percentages in the 0-1 με range are not due to exceptionally light vehicles; rather, they are related to the positioning of the vehicles on the bridge.

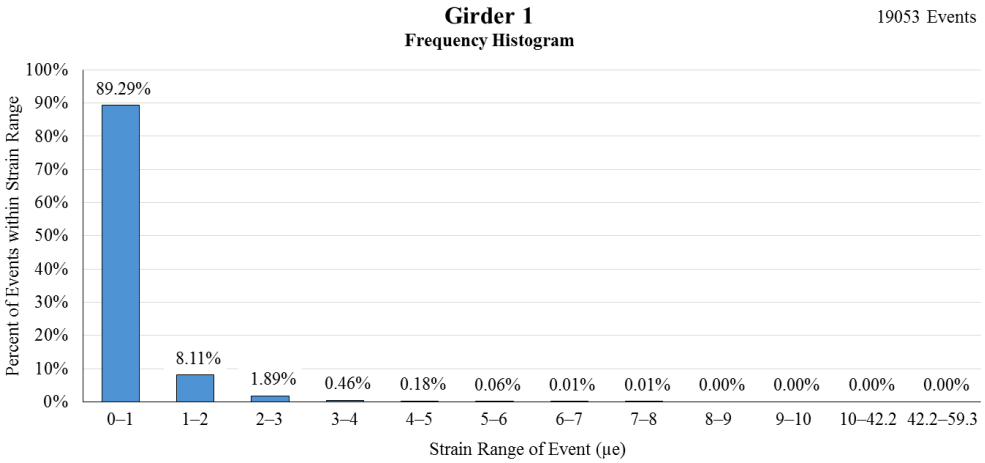

Figure 40. Frequency histogram of recorded strains in Girder 1.

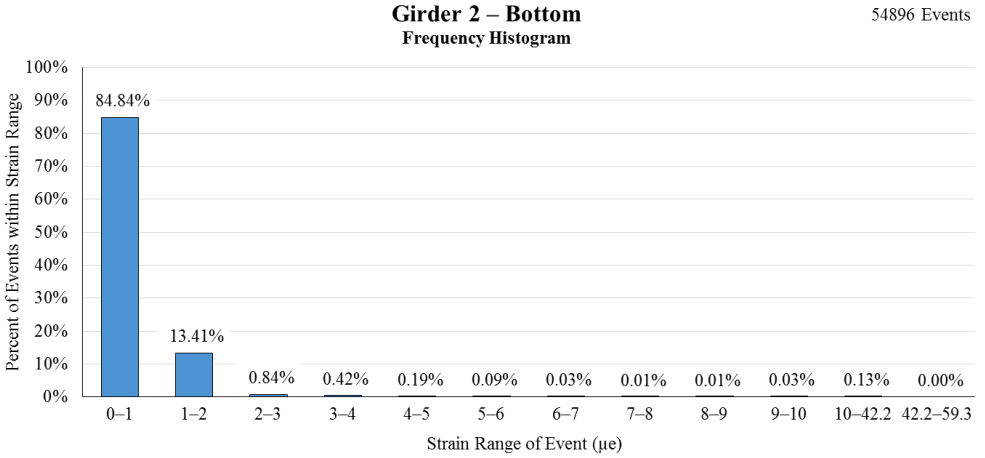

Figure 41. Frequency histogram of recorded strains at the bottom of Girder 2.

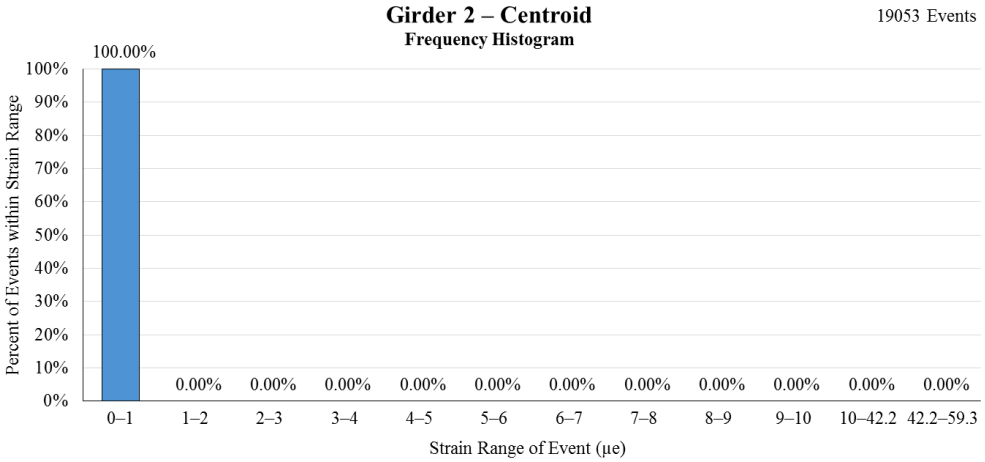

Figure 42. Frequency histogram of recorded strains at the centroid of Girder 2.

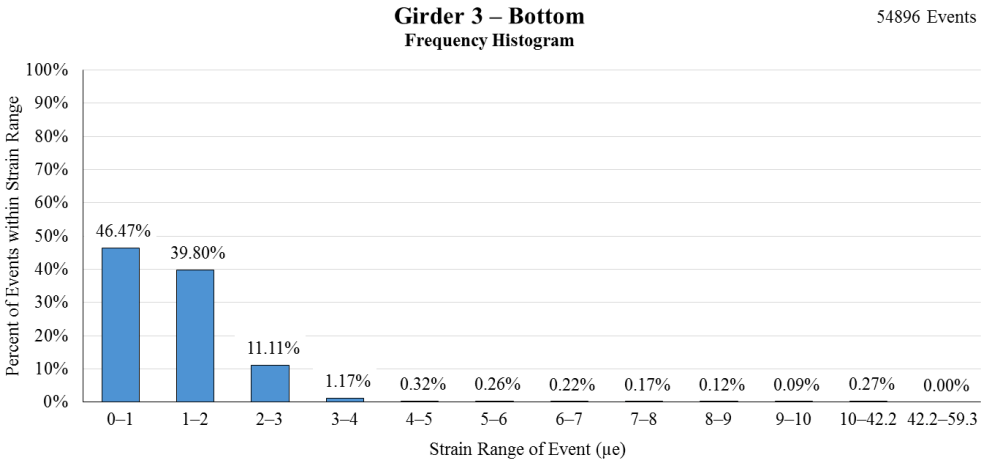

Figure 43. Frequency histogram of recorded strains at the bottom of Girder 3.

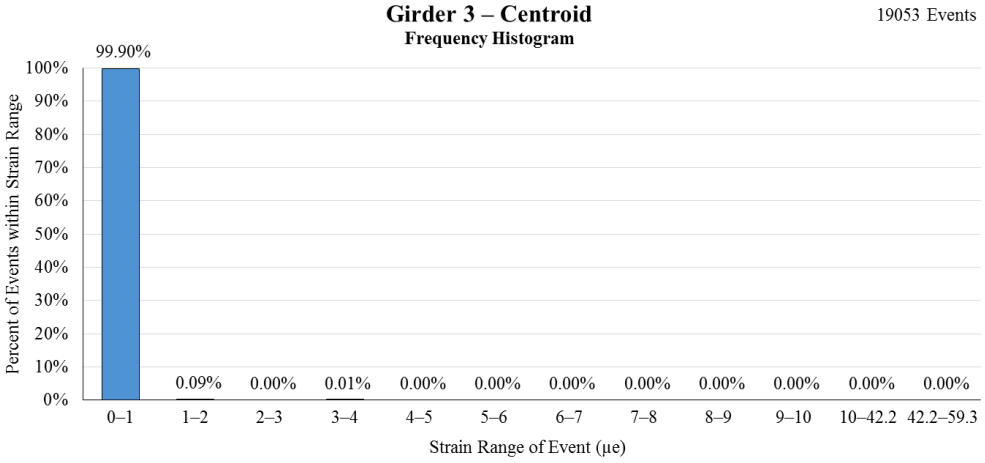

Figure 44. Frequency histogram of recorded strains at the centroid of Girder 3.

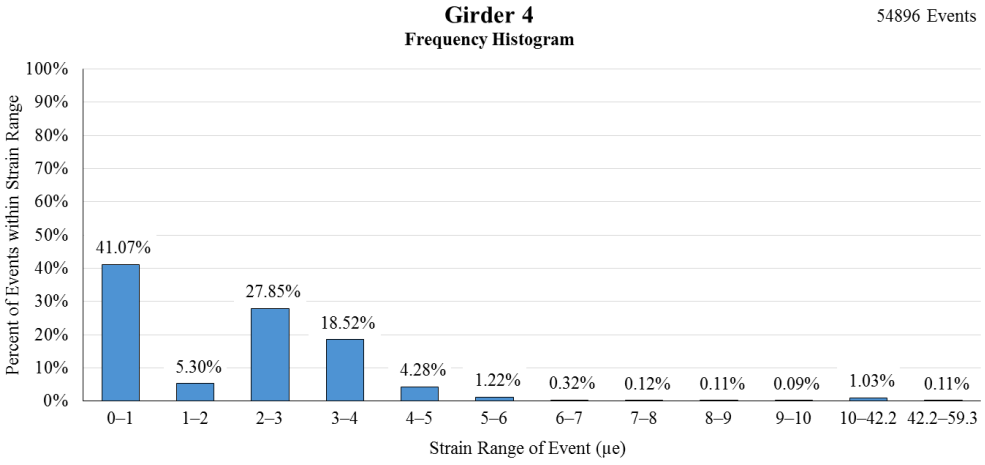

Figure 45. Frequency histogram of recorded strains in Girder 4.

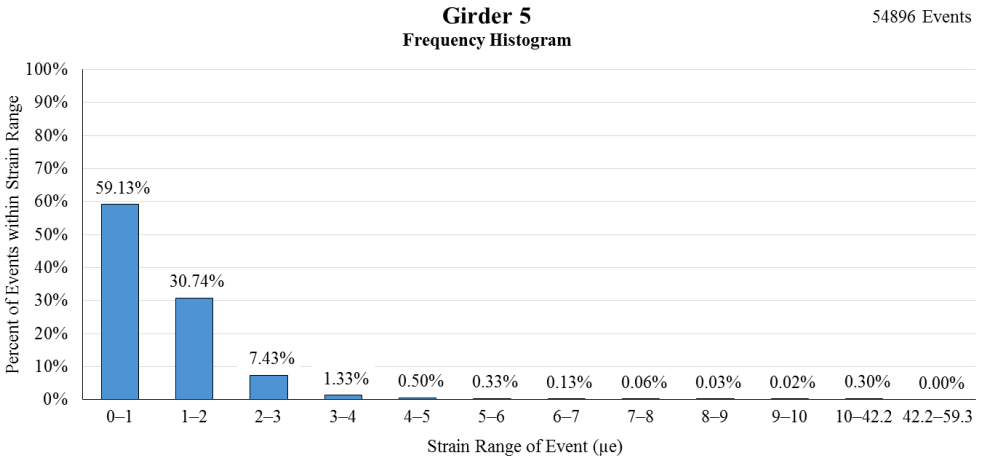

Figure 46. Frequency histogram of recorded strains in Girder 5.

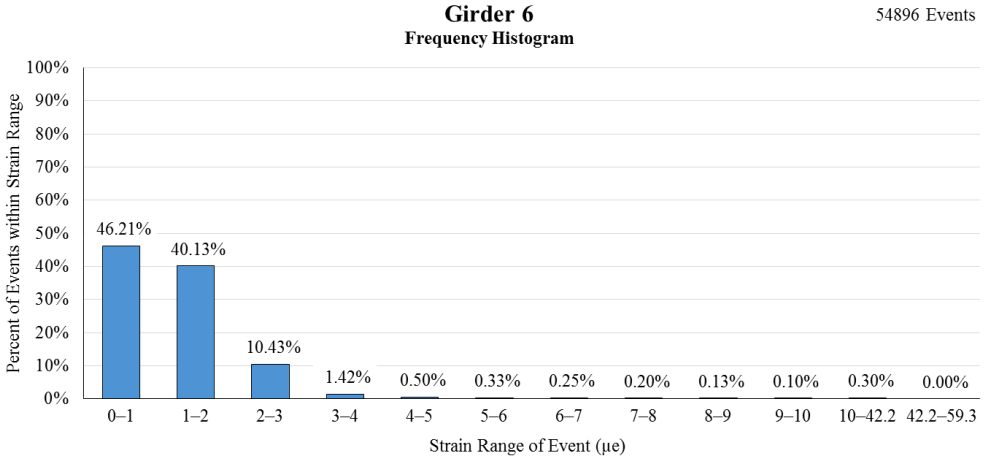

Figure 47. Frequency histogram of recorded strains in Girder 6.

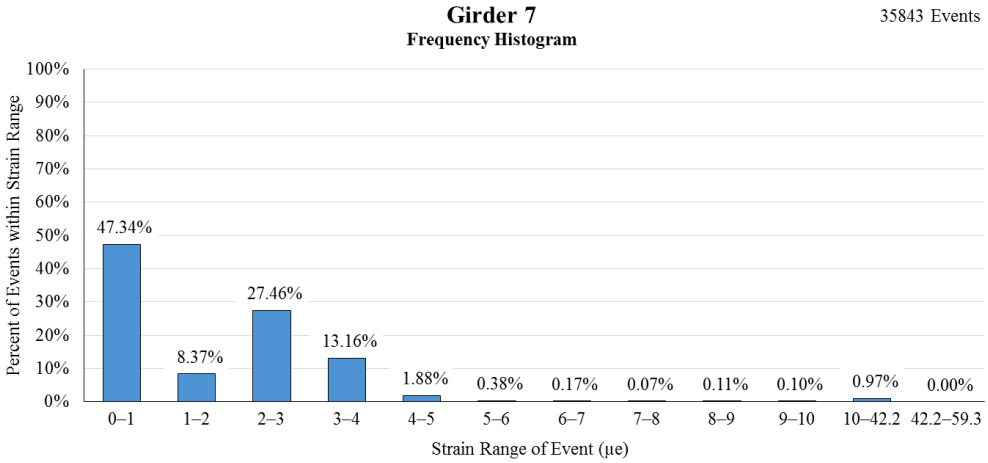

Figure 48. Frequency histogram of recorded strains in Girder 7.

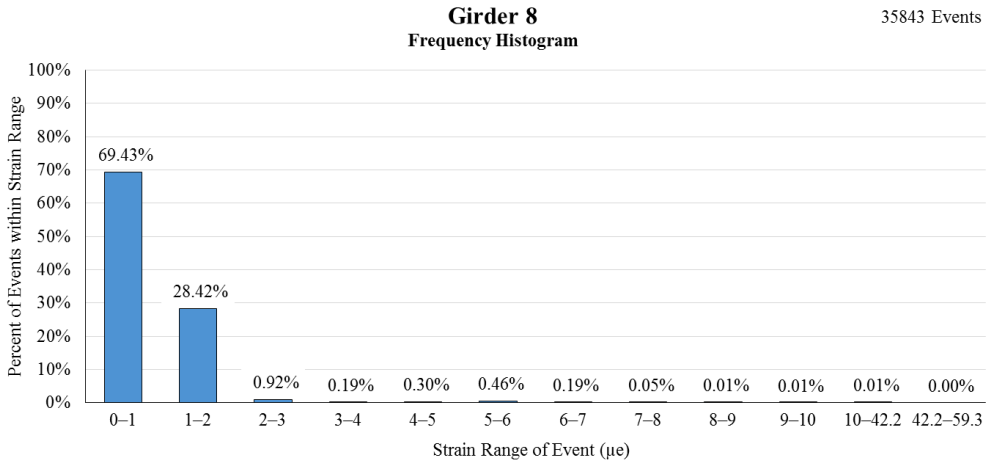

Figure 49. Frequency histogram of recorded strains in Girder 8.

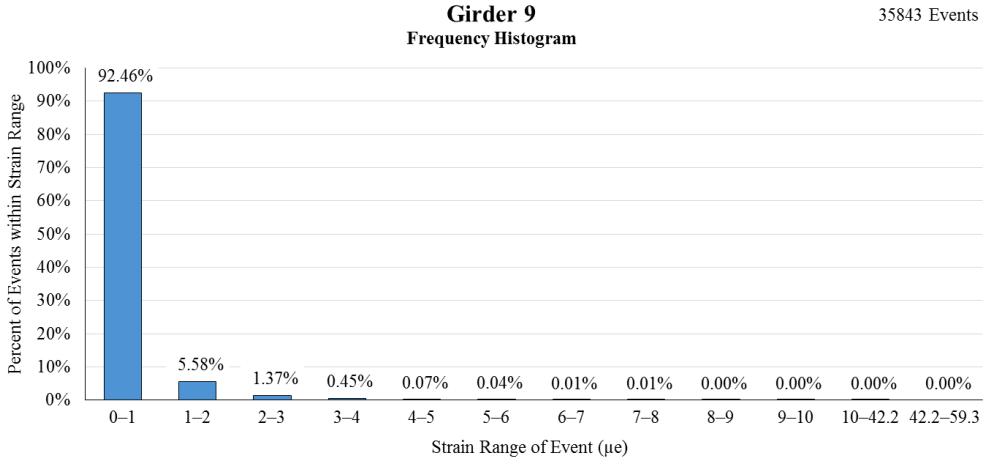

Figure 50. Frequency histogram of recorded strains in Girder 9.

As expected, the sensors located at the centroids of Girders 2 and 3 recorded zero strain 99.9% of the time. This is because there is no strain due to moment at the elastic neutral axis of a girder. The events where the strains are greater than 1.0 με are likely due to either excessive noise in the sensors or errors in the data processing programs.

The monitored exterior girder, Girder 1, also experienced little to no strains. This is because Girder 1 lies directly beneath the sidewalk, where there are little heavy loads. Girder 1's largest recorded strain, 7.24 με, was the result of an even larger strain in

Girder 4, 52.78 με. In fact, the three outside girders along each side of the bridge all experienced their strains as a result of higher loading in a nearby girder. Analysis shows that 97.70% of strains in Girders 1, 2, 3, 8, and 9 above 5.0 με were caused by larger strains in other girders.

The histograms show that the level of strain that a girder experiences is a direct function of where the girders are positioned in relation to the traffic lanes. The girders directly beneath the traffic lanes experienced much higher levels of strain than girders that were positioned farther away. The histograms also reflect the symmetry of the bridge. Girders 4 and 7 have similar histograms, as do Girders 5 and 6, Girders 2 and 9, etc.

After analysis of Figures 40-50 was complete, a frequency histogram (Figure 51) and normalized distribution plot on probability paper (Figure 52) were created using the maximum strain in any girder on the bridge for each event. Unlike Figures 40-50 above, Figures 51 and 52 do not consider the positioning of the vehicle on the bridge. Instead, the data is directly related to the gross vehicle weight (GVW) of the vehicles crossing the Nibley Bridge. Because the event extraction program ignored strains below 1.0 με, none of the maximum strains were below 1.0 με.

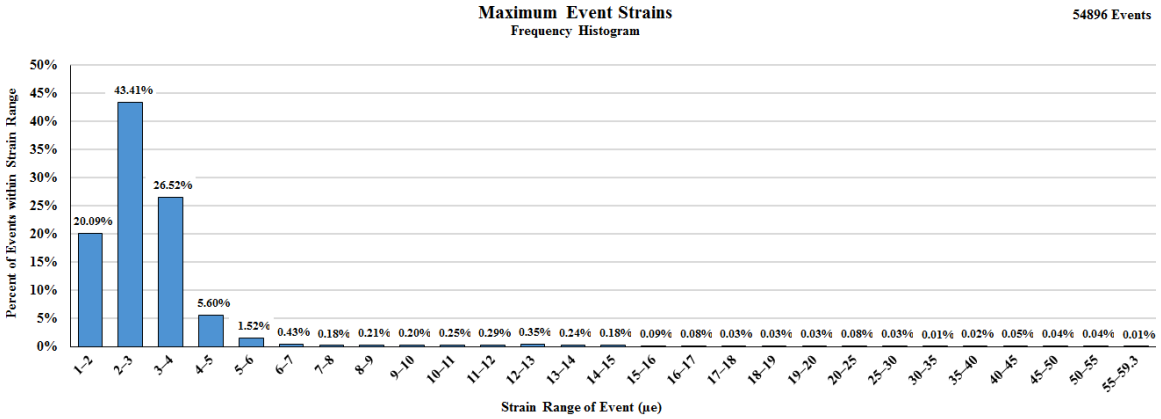

Figure 51. Frequency histogram of maximum recorded event strains in any girder.

Figure 51 shows that the vast majority of vehicles (95.62%) cause less than 5.0 με strain in the bridge. The frequency of strain events flattens out after approximately 7 με. However, there are clusters of vehicles at approximately 12.5 με, 22.5 με, 42.5 με, and 52.5 με. The cluster of vehicles at 12.5 με most likely corresponds to school buses and semis crossing the bridge. Likewise, the clusters of vehicle strains at 22.5 με, 42.5 με, and 52.5 με likely correspond to construction equipment.

Figure 52 plots the distribution of the maximum strains in the bridge on normal probability paper. From this figure, it can be shown that the vehicles on the bridge caused a mean strain of 2.67 με with a standard deviation of approximately 0.886 με. It also reveals a large variation in the intensity of strains that occur on the Nibley Bridge.

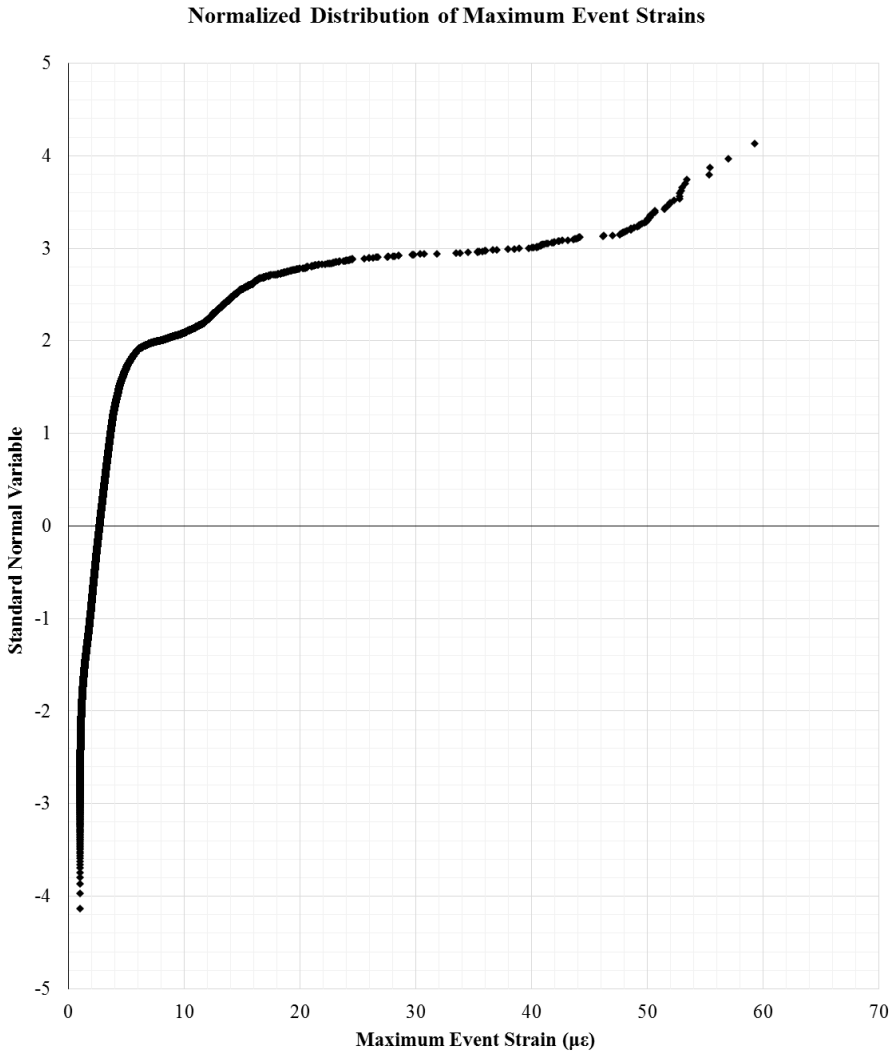

Figure 52. Maximum bridge strains plotted on normal probabality paper.

Comparing Figure 52 against Figure 53 reveals that the Nibley Bridge traffic data collected closely resembles the lower leg of similar traffic studies done by the Federal Highway Administration (FHWA). This is to be expected, as many of the largest strains that occur on highway bridges are not present in rural Nibley. It also serves to validate the findings of this project by showing that the collected data compares well with previously collected data. (While the horizontal axis of Figures 52 and 53 are different,

the maximum event strain and gross vehicle weight are directly correlated, allowing for a direct comparison).

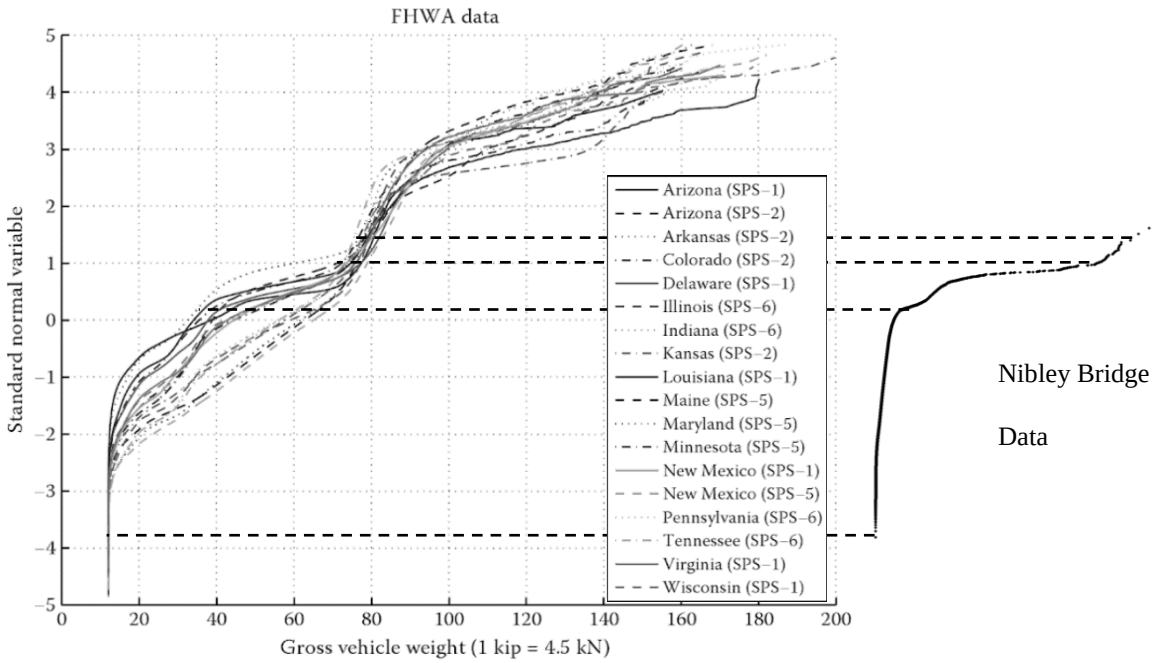

Figure 53. GVW of trucks suveyed on roads from 16 different states, annotated (Nowak and Collins 2013).

3.4.7 Girder Distribution Factors

Using the 54,896 data points collected during the summer and winter, girder distribution factors (GDFs) were calculated for the north and south lanes. This was done by first sorting the events by the lane in which they occurred. Then, average strain values were calculated for each girder. The symmetry of the bridge was used to fill in gaps in the data. Finally, the girder distribution factors were calculated using Equation 2. The multiple presence factor, m, equals 1.2 when a single lane is loaded and 1.0 when both lanes are loaded. The results are shown graphically in Figures 54 and 55.

$$\mathrm{GDF}_i = \mathrm{m}\frac{\varepsilon_i}{\sum \varepsilon_i} \qquad \text{Equation 2.}$$

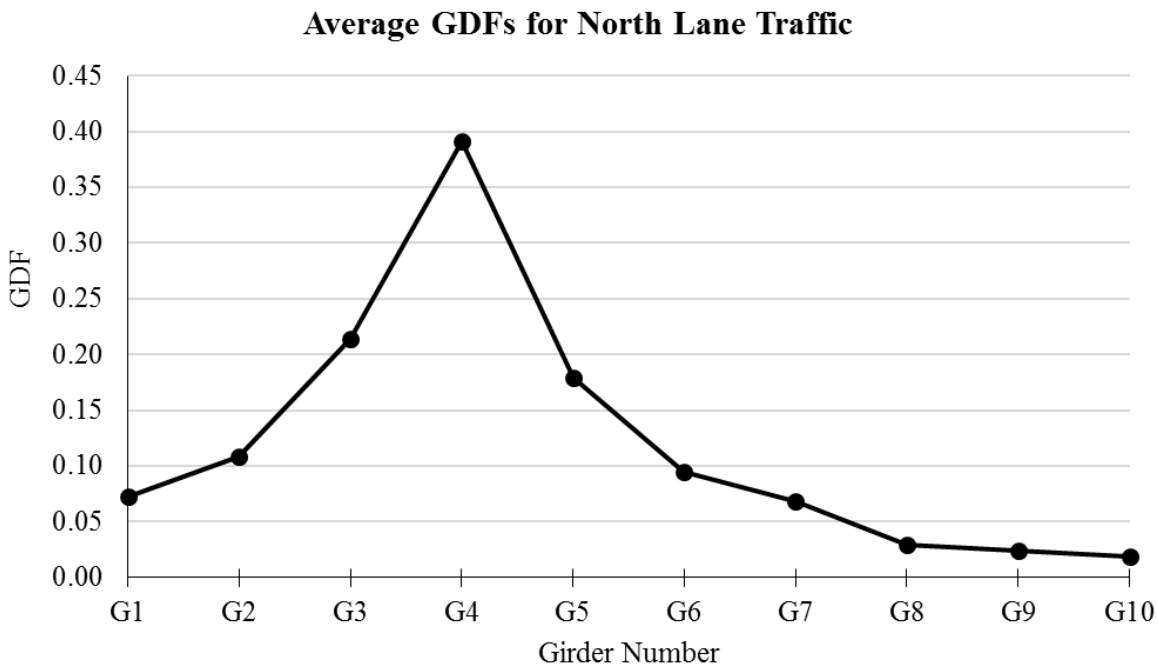

Figure 54. Girder distribution factors for north lane.

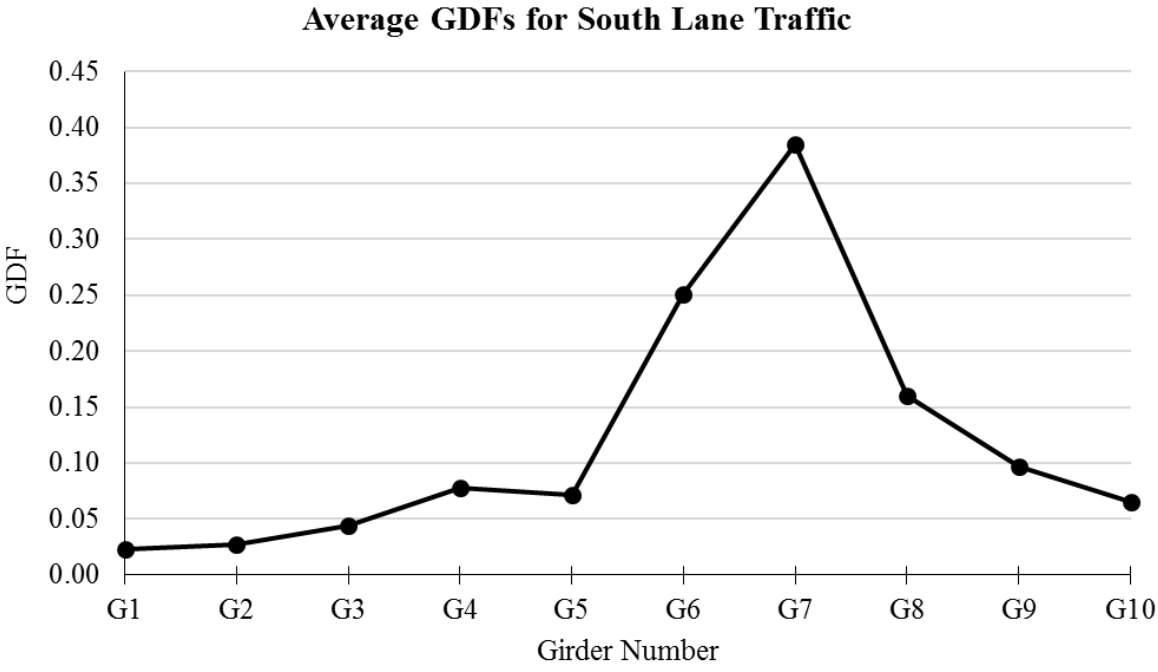

Figure 55. Girder distribution factors for south lane.

Figures 54 and 55 show that the girders directly below the traffic lanes typically carried the majority of the load. The fourth girders in from each end typically carried

approximately 32.3% of the load by themselves. All other girders carried less than 25% of the load, and the girders on the farthest side of the bridge from the loaded lane typically did little to carry the load.

The calculated girder distribution factors for the south lane appear to be slightly skewed to favor the girders on the north end of the bridge. In Figure 55, Girder G6 takes a higher share of the load than its symmetric girder, Girder 5, does in Figure 54. The girder distribution factor for Girder 4 in the south lane is higher than that of Girder 5, which seems incorrect. This could support the theory that Sensor B1978 attached to Girder 4 has malfunctioned and needs to be calibrated.

By adding the average strains from each lane together, girder distribution factors for multiple loaded lanes can be calculated. The girder distribution factor for loading multiple lanes is shown in Figure 56. As can be seen, loading multiple lanes does not significantly impact the girder distribution factors for the worst-case girders, Girders 4 and 7. Table 5 numerically contains the data shown in Figures 54, 55, and 56.

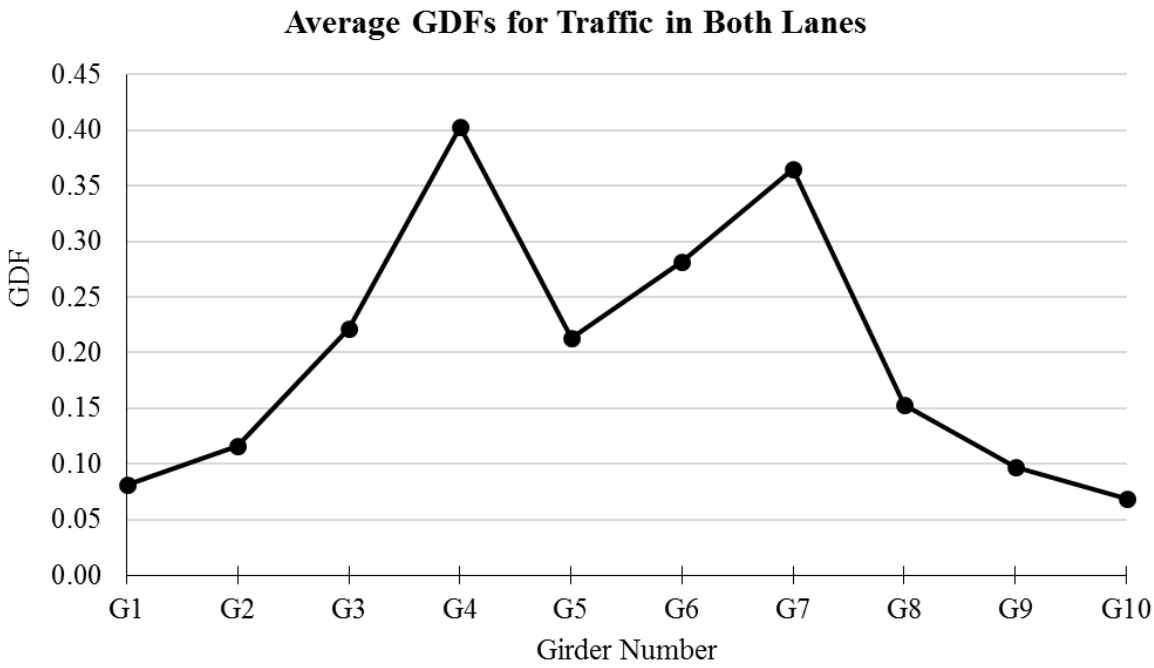

Figure 56. Girder distribution factors for multiple loaded lanes.

Table 5. Girder distribution factors.

Girder	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
North Lane	0.0728	0.1082	0.2134	0.3916	0.1793	0.0945	0.0681	0.0292	0.0235	0.0193
South Lane	0.0225	0.0275	0.0442	0.0774	0.0711	0.2509	0.3849	0.1603	0.0963	0.0648
Multiple Lanes	0.0814	0.1163	0.2215	0.4034	0.2130	0.2817	0.3648	0.1527	0.0970	0.0682

Comparing the controlling girder distribution factors from Table 5 against the AASHTO LRFD Bridge Design Specifications (4th Edition) reveals that the girder distribution factors for a single lane are appropriately conservative. For a bridge with the Nibley Bridge's specifications, the code requires a GDF of 0.4014 for a single loaded lane; the controlling single-lane GDF was 0.3916.

However, the AASHTO design GDF for when multiple lanes are loaded is 0.5506. The controlling measured GDF for this scenario was only 0.4034. Thus, the girder distribution factor for multiple loaded lanes is overly conservative by 36.51%.

CHAPTER 4

DRONE MODELING

4.1 Drone Description

4.1.1 Factors Influencing Drone Selection

Originally developed for military use, unmanned aerial vehicles (UAVs) or "drones" experienced a surge of growth in the commercial sector in the late 2000s. Today, drones have become commonplace in many industries including agriculture, construction, and real estate. There are currently a variety of commercial drones on the market manufactured by multiple different companies, including Dà-Jiāng Innovations (DJI), Parrot SA, and Xiaomi Inc. However, due to the unique nature and functions of bridges, not every drone is a suitable candidate for bridge inspection and modeling.

One of the main limitations of using drones for bridge inspection and modeling is that most drones on the current market cannot capture images of the underside of a bridge. This is because most commercial drones are designed to capture aerial footage of the ground below the UAV and cannot rotate their cameras to capture pictures higher than the horizontal plane (See Figure 57). Thus, these drones cannot look directly vertical to take pictures of the underside of bridge girders.

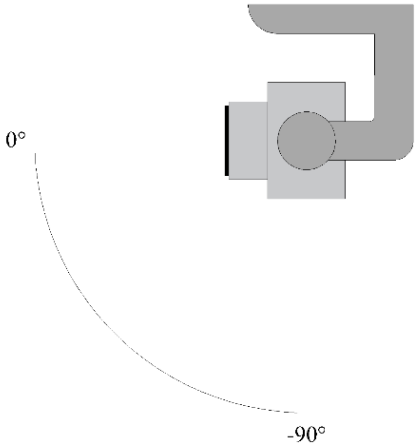

Figure 57. Gimbal Pitch Range of Typical Commerical Drones.

The market is currently expanding to meet these needs. For example, the senseFly Albris and the DJI Matrice 210 are both capable of looking directly up. However, quotes acquired for these drones priced them at $20,000 and $9,853 respectively, which was outside the budget for this project.

Another feature that factors into selection of the drone is the limitations of current GPS technology. For the most accurate modeling results, the drone should be capable of geotagging the images that it captures. Geotagging refers to the process whereby the drone records the latitude, longitude, and elevation at each point an image is taken and appends this location data to the image data. When the photos are imported into modeling software, the software uses this data to pinpoint the precise location where the images were taken.

When a drone flies underneath a bridge, especially when the bridge deck is made with reinforced or prestressed concrete, it tends to lose GPS signal. Some drone software includes a built-in failsafe, where if the drone loses GPS signal it will fly directly

upwards and return to its takeoff point. If the drone is underneath a bridge deck, this can cause the UAV to strike the structure, damaging the drone.

Currently, if the drone does lose GPS signal, the best-case scenario is that the drone continues to take pictures without geotagging the image. Although this makes it harder for the software to recognize the spatial location where each image was taken, it allows the pilot to continue to fly the drone underneath the bridge.

A third factor in drone selection is the maximum distance that the drone can be away from the surface being inspected. This distance is based off of the horizontal sensor density (L_H, pixels) and horizontal field of vision (HFOV, degrees) of the camera attached to the drone. It can be calculated using Equation 3, where D is the maximum viewing distance and PPI is the required image resolution:

$$D = \frac{L_H}{\text{PPI}} \cot\left(\frac{\text{HFOV}}{2}\right)$$

Equation 3 (Lattanzi and Miller 2015).

In their 2015 study, Lattanzi and Miller determined that a minimum resolution of 20 pixels per inch (0.8 pixels per mm) was required for successful reconstruction of concrete structures. Thus, the drone chosen should be able to take photos at the required resolution based on the appropriate distance for the given bridge. If the distance is lower than a few feet, it can be difficult to maneuver the drone around the structure without crashing the drone into the bridge. However, a larger HFOV allows the camera to capture more of the surrounding details per image, which can aid in the modeling process.

Related to the previous factor is the standard camera attached to the drone. Most drones come with a standard optical camera; however, some higher-end drones offer interchangeable cameras that can be detached from the drone mount. This allows the

operator to attach an optical camera with a different zoom lens, infrared, or thermal camera. If necessary, a drone with this feature could be a valuable asset to a bridge inspection project.

Finally, drone stabilization, maneuverability, and obstacle avoidance should be taken into account. A drone's ability to stabilize itself will insure that pictures are non-blurry and keep wind from blowing the drone off-course. Maneuverability is critical because it allows the drone to navigate around vegetation, power lines, and other structures that surround many typical bridges. Finally, some drones now feature auxiliary sensors that can keep the drone from crashing into obstacles mid-flight in case of user error.

4.1.2 Features of the Selected Drone

The drone selected for this project was the DJI Phantom 4 Pro (See Figure 58). The Phantom 4 Pro was selected due to the compatibility of DJI drones with third-party software and because it is representative of the many commercial drones on the market. The cost for this model was $1500, with an additional $200 spent on extra batteries, propeller guards, etc.

Figure 58. DJI Phantom 4 Pro (Dà-Jiāng Innovations 2017).

The Phantom 4 Pro is a quadcopter UAV with a 20 MP camera attached on the underside of the drone. The gimbal can rotate the camera between 0° and -90°. With HFOV=60° and L_H=5400 pixels (a 20 MP image is 3600 x 5400 pixels), the camera has a maximum viewing distance of 38.5 feet (11.7 m). For this drone, the camera is non-detachable.

The Phantom 4 Pro contains infrared sensors on each side of the drone, which allow it to detect obstacles and automatically brake to prevent collisions. This was useful while maneuvering through the tight areas surrounding and underneath each bridge. The GPS system is accurate ±1.64 feet (0.5 m) vertically and ±4.92 feet (1.5 m) horizontally. The drone is able to continue to fly and capture images without GPS signal. Additional specs for the Phantom 4 Pro can be found in Appendix B.

The Phantom 4 Pro also comes with a remote control pad (See Figure 59). While users can fly the drone solely using the remote control pad, the controller has ports that

allow the operator to connect a phone or tablet into the control pad. DJI's downloadable Android and Apple apps then allow the operator to view the field of vision of the drone's camera on the screen of the device.

Figure 59. Remote Control Pad for the DJI Phantom 4 Pro (Dà-Jiāng Innovations 2017).

This feature is particularly useful during image capture, as it allows the inspector to focus in on problem areas of the bridge. Connecting a device to the remote control pad also allows third-party apps to control the drone in-flight, allowing the drone to automatically fly grid and orbit patterns for 3D modeling purposes.

4.1.3 Drone Mapping Apps

While several drone mapping apps exist, all perform the same primary function: the user sets boundaries on an area to be mapped, then the app takes control of the drone

and flies it over the selected area. During flight, the app uses the drone to take the pictures required for a three-dimensional model.

The advantage of using an app is that a computer is able to take photos more consistently and evenly spaced than a human operator. However, a certified remote pilot is still required to supervise the flight, and the drone operator should be able and ready to take control of the drone at any point during the operation.

After experimenting with third-party drone mapping apps, the decision was made to use DroneDeploy's Android app to automatically guide the drone inflight. A screenshot of the DroneDeploy flight-planning software as viewed in a web browser is shown in Figure 60.

Figure 60. Screenshot of the DroneDeploy Flight-Planning Software (DroneDeploy Labs 2017).

The above screenshot shows an early version of the flight over the Nibley Bridge. As shown in the figure, the drone will take off and proceed to the point labeled "Start".

From there, it will proceed due east (right) along the horizontal line, automatically taking pictures of the ground with the camera angled straight down.

When the drone reaches the boundary line, the drone will turn, proceed to the next horizontal line, and continue the pattern. The drone will continue along the back-and-forth grid pattern until it has covered all of the ground within the boundary lines.

Once the drone has completed its back-and-forth pattern, it will proceed to the edge of the circle shown in Figure 60. It will then descend several feet and adjust its camera angle to an angle between -90° and 0° (The exact distance and angle depends on the flight altitude and perimeter of the boundaries). The drone will then fly in a circle, or orbit, facing inward, automatically taking pictures of the object at the center.

Flying an orbit around the bridge allows the drone to capture oblique imagery. Oblique imagery (side shots of an object taken slightly from above) is critically important to the modeling process because it allows for a more detailed reconstruction of the sides of the model.

Several settings must be considered prior to attempting a flight via a drone mapping app. Care must be taken when setting the altitude and boundaries to ensure that the drone will not collide with any obstacles midflight. The spacing of the lines in the back-and-forth pattern can be adjusted to force the drone to increase or decrease the number of passes over the object. The frequency at which the drone takes pictures can also be adjusted. Finally, it is advisable to perform an additional manual flight after the automated flight is complete. Doing so will allow the operator to focus in on problem areas and capture pictures from angles where the drone mapping app did not fly.

4.2 Modeling Software Description

4.2.1 Modeling Software Attributes

Two main methodologies exist to recreate structures digitally: dense structure from motion (DSfM) and image mosaicing (IM). Both of these methodologies have inherent pros and cons that should be considered prior to any modeling project.

DSfM modeling uses photos taken of the structure by a drone to create a digital model of the structure. This is done by first loading the images into the software and selecting the software's equivalent of "Create Model". The software then compares the location of thousands of identifiable points relative to each other in each image. For example, the software may identify a corner of a deck slab, bolts on a steel member, or striping on a road. Using these points as a reference in conjunction with any GPS data appended to the image, the software identifies the location and angle at which the image was taken, as shown in Figure 61.

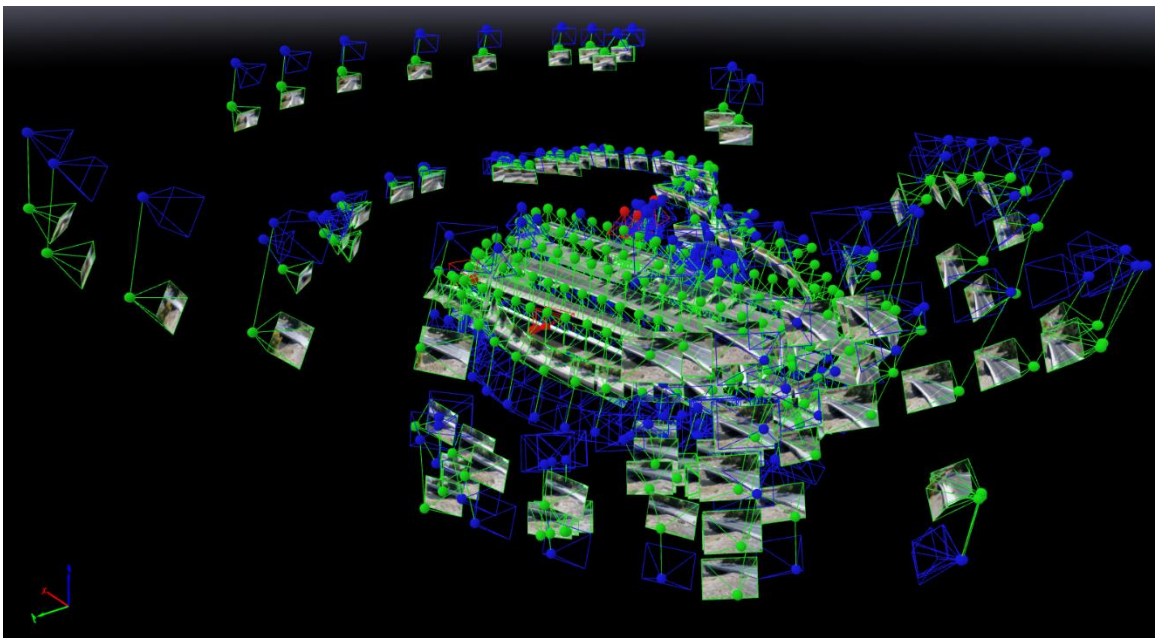

Figure 61. Geolocated images in Pix4D.

Once the camera location and angle of each image has been determined, the modeling software then creates a three-dimensional point cloud of the structure by again comparing the location of each identifiable point between the various images. It then calculates the location of the point, resulting in a cluster of unconnected points as shown in Figure 62.

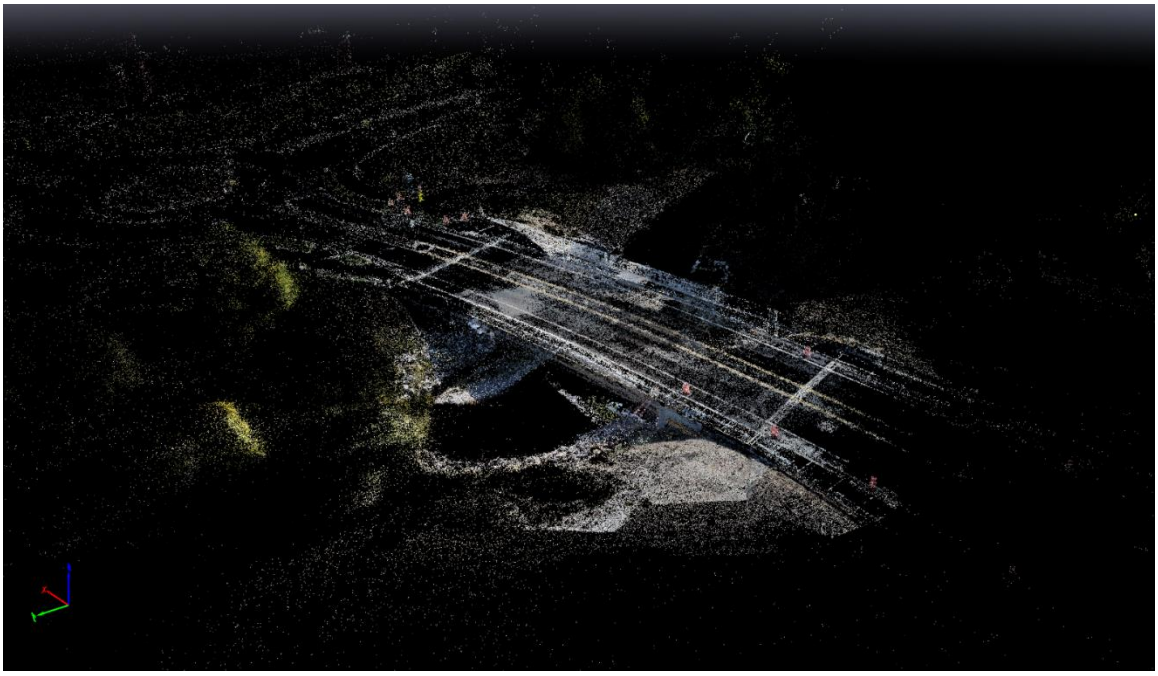

Figure 62. Three-dimensional point cloud of a structure in Pix4D.

Finally, the modeling software uses Poisson meshing to connect the points in the three-dimensional point cloud and create a blank three-dimensional mesh. It then combines and projects the images onto the structure to color it and add texture. A finished 3D model is shown in Figure 63.

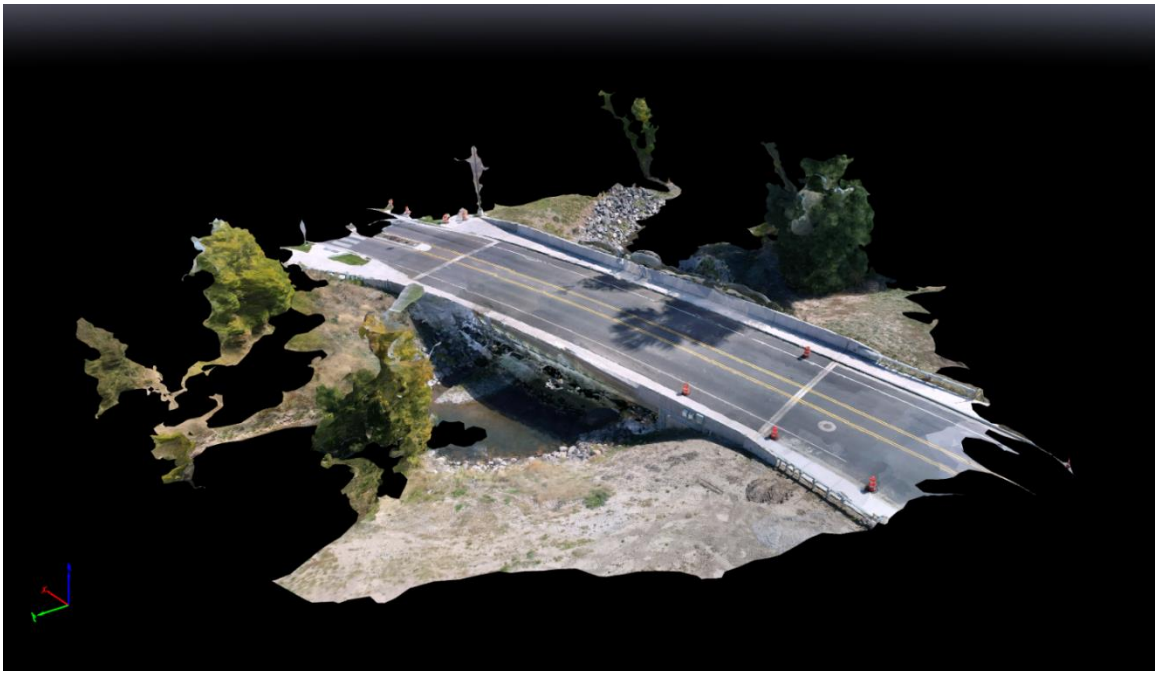

Figure 63. Finished three-dimensional model in Pix4D.

DSfM modeling has an advantage over IM modeling in that it can create a model using solely images. However, the resulting model is only as accurate as the algorithms the software uses to create it. For example, can be difficult for DSfM software to recreate smooth surfaces and thin objects (e.g. a roadway surface or the wire in a chain-link fence, respectively).

Image mosaicing or IM modeling, on the other hand, uses a pre-existing base model of a structure in its modeling process. This allows the modeling software to skip creating a three-dimensional point cloud and mesh, instead directly projecting the images onto the model. This usually creates a cleaner-looking model in comparison to the DSfM process.

However, creating the base model for IM modeling requires knowledge of the exact measurements of the bridge as constructed. The time required to collect these

measurements and create the 3D base model can outweigh the benefit of this method of modeling in practice, especially if an organization monitors and inspects several bridges.

Because three-dimensional models of the bridges investigated in this study do not exist, this study focused solely on DSfM modeling. Most of the commercially available software for creating 3D models from drone images uses DSfM modeling, which worked well for this project.

4.2.2 Modeling Software Used

In order to test a broad variety of modeling software available on the market, several different brands of software were tested. The testing included three different brands of modeling software: Pix4D, DroneDeploy, and Agisoft PhotoScan. Models were created of all three bridges using all three software products. Each model was created using the exact same photographs of each bridge. If the software placed a limit on the maximum number of photos that could be used for a project, the best photos of each bridge were used.

4.2.2.1 Pix4D

Pix4D is a desktop-based modeling software, although browser-based modeling is available. It was first developed in 2011 by its parent company, Pix4D SA. There are no limitations on the number of photos that can be used to create a model in Pix4D. However, including too many photos significantly increases the processing time required to generate the model. Additionally, Pix4D allows pictures to be included in a model even if they are not geotagged.

One of Pix4D's unique features is the ability to manually add "tie points", Pix4D's term for the easily identifiable points discussed above. While the software automatically identifies as many points as possible, the ability to manually add tie points allows the user to increase the accuracy of the model. For example, identifying the corner of an overhang in multiple photos can allow the software to recognize that the overhang exists. However, the time that it takes to add these points makes it impractical to do more than a handful of manual tie points per model. Manually added tie points are not a substitute for good photography practices.

Another useful feature of the Pix4D app is its ability to use videos to create a model. The software does this by separating individual video frames into still images. However, this often results in thousands of images, which can result in extremely slow processing times if video is used extensively in model creation.

Pix4D has a free mobile app that can be used to control a drone in a similar manner as the DroneDeploy app described in Section 4.1.3. However, after several attempts, the Pix4D app was unable to communicate with the Phantom 4 Pro drone. Because of this, the DroneDeploy Android app was used to fly the overhead back-and-forth pattern. However, because each software used the same images to create models, this did not have an impact on the final results.

4.2.2.2 DroneDeploy

DroneDeploy is a browser-based modeling software. It is the newest of the three modeling software brands selected for this project, having first been released in 2013.

Because DroneDeploy is a browser-based modeling software, photos used to create a 3D model are uploaded to DroneDeploy's server, which then processes the data and creates the model. Because photos are stored on their server, DroneDeploy limits uploads to 500 pictures per model. It also recommends using only 3.0 GB of pictures per model.

It was found that for this project, the 3.0 GB limit effectively limited each model to approximately 388 pictures. Additionally, images used to create a model in the DroneDeploy software must be geotagged, meaning that the supplementary pictures taken using a Nikon D90 camera could not be included in the DroneDeploy models.

The DroneDeploy app was the only app tested in this research that was able to communicate with the Phantom 4 Pro drone. For the most part, the app was able to function well. One useful feature of the app was that if the app crashed, the drone would continue to fly its mission and would reconnect to the app once it was reopened. One drawback of the app was that if the mission parameters were set too high (by increasing the required overlap between pictures beyond approximately 80%), the drone would take off but remain motionless in the air while taking pictures of the same spot repeatedly. Another drawback was that all flights using the app needed to be planned using an internet connection. This limited the versatility of the app and made it difficult to make changes to the flight plan in the field.

4.2.2.3 Agisoft PhotoScan

Agisoft PhotoScan is a desktop-based modeling software similar in many ways to Pix4D. However, while Pix4D and DroneDeploy are specifically designed to process

images taken using a drone, PhotoScan is a more general modeling software. It has also been around for longer than the other two software brands, having first been released by Agisoft LLC in 2010.

Agisoft PhotoScan has the ability to use images that are not geotagged, which allowed supplementary pictures taken using a camera to be included in the model. There is also no limitation on the number of images that can be included in the model. However, including too many photos has a diminishing return and can significantly lengthen the time it takes the software to process the model. Agisoft PhotoScan does not have an app able to control drones in-flight.

One of Agisoft PhotoScan's most useful features is the ability to crop finished models. As shown later, finished models usually contained errors. For example, the algorithms in all three versions of software often created shapes floating in the air above the bridge deck. The ability to crop the model allowed these shapes to be deleted, giving the model a clean, crisp look. Cropping the model also allows the user to delete nearby obstacles (such as trees) that may make viewing the bridge difficult.

4.3 Drone Flights and Testing

Three bridges near Utah State University were selected to be modeled for this project: the Nibley Bridge, the Trenton Bridge, and the Elwood Bridge. At each bridge, hand measurements were taken of any defects in the bridge (cracks, flaking, etc.). Artificial "cracks" were also created by applying masking tape to the bridge surface, drawing lines with a marker, and then measuring the lines (See Figure 64). Finally, four

coins (a dime, penny, nickel, and quarter) were placed on the bridge deck for comparison in the finished model.

Figure 64. Artifical “crack” at the Trenton Bridge.

A similar flight pattern was followed at each bridge. First, the DroneDeploy Android app was used to automatically fly the Phantom 4 Pro drone in a back-and-forth pattern and capture aerial footage directly over the bridge. The app also controlled the drone for a single oblique orbit in order to capture pictures of the side of the bridge. Following the flight controlled by the DroneDeploy Android App, the DJI Android app was used to manually fly the drone and capture supplemental images.

A limited number of additional images were also taken using a Nikon D90 camera with an 18-105 mm zoom lens. This allowed pictures to be taken in areas where it was

difficult to fly the drone, such as the underside of the bridge. The resulting images from the drone and camera were then loaded into the modeling software for analysis.

4.3.1 Nibley Bridge

The first bridge modeled for this project was the same bridge in Nibley, Utah, that was being monitored for periodic loading. A comprehensive description of the Nibley Bridge is given in Section 3.1. Figure 65 shows an image of the Nibley Bridge captured using the Phantom 4 Pro drone.

Figure 65. Photograph of the Nibley Bridge.

This was a difficult bridge to model due to the volume of traffic crossing the bridge (an important consideration when picking a bridge to model). Federal law prohibits flying a UAV above moving vehicles and pedestrians not taking part in the

flight, which meant that the drone could not fly above the bridge deck while either was on the bridge.

To help guarantee a safe flight experience, the decision was made to fly the drone at least 25 feet (7.62 m) above the bridge deck. Flying the drone at a higher altitude decreased the resolution of the photos of the bridge. The high volume of traffic also invalidated several oblique photos taken by the drone, as moving vehicles/pedestrians can confuse the modeling software (See Section 4.5.2.1).

Surrounding vegetation was also an issue. Several trees border the Nibley Bridge to the north and south. There is also a light pole to the southeast of the bridge. This meant that the boundaries of the automatic flight had to be decreased in order to avoid a collision. It also made it difficult to capture oblique photos from the direction of the trees.

The flight path for this bridge consisted a single back-and-forth pattern at 25 feet above the bridge deck and three oblique orbits at different heights around the bridge. The drone was also flown along each side of the bridge only a few feet above the water, in order to capture images looking underneath the bridge. The flight was completed by taking several photos of the bridge deck from a higher altitude and by flying close to the cracks being measured and taking pictures.

4.3.2 Trenton Bridge

The Trenton Bridge (National Bridge Inventory Structure Number 005034C) is a steel truss bridge located near Trenton, UT (41°56’20.20” N 111°56’26.00”W). Constructed in 1925, this 151-foot (46-meter) bridge carries a county road over the Bear River. The Trenton Bridge was selected in order to see how well the software could

model a steel truss as opposed to a concrete girder bridge. Figure 66 shows a photograph of the Trenton Bridge.

Figure 66. Photograph of the Trenton Bridge.

Given the rural location of the Trenton Bridge, there was a relatively low volume of traffic compared to the bridge in Nibley. This resolved many of the concerns about traffic flow. The largest obstacles associated with this bridge stemmed from the overhead truss of the bridge itself and from a string of power lines that run over the river approximately 45 feet south of the bridge. Additionally, while setting the drone up for flight it was discovered that the batteries were only charged at 60% capacity, which limited the drone's flight time and the number of photos that were taken of this bridge.

The flight path for the Trenton Bridge consisted of a single back-and-forth pattern to capture aerial footage at 35 feet (approximately 10 feet above the truss), followed by an oblique orbit. The drone was flown along each side of the truss in order to capture

more detail from the side, followed by a handful of oblique images captured by flying the drone manually.

4.3.3 Elwood Bridge

The Elwood Bridge is a steel girder bridge located in Elwood, UT (41°38'50.17"N 112° 8'28.65"W). Built in 2001, the bridge spans the Corinne Canal and provides farmers access to their fields from 5200 W. This bridge was specifically chosen because of the attributes listed in the following paragraph, which made the Elwood Bridge an ideal candidate for 3D modeling. Figure 67 shows a photograph of the Elwood Bridge

Figure 67. Photograph of the Elwood Bridge.

After flying the Nibley and Trenton Bridges, the decision was made to find a bridge that had no surrounding obstacles, a low volume of traffic, and a short span. Because the Elwood Bridge crosses a rural canal and is surrounded by fields, the

surrounding landscape is completely flat with no potential obstacles for several hundred feet. The Elwood Bridge carries a dirt road and is intended solely for access by farmers, so traffic on the bridge was non-existent during the drone's flight. Finally, the short span of 24 feet (7.32 m) allowed the drone to capture more overlapping pictures, potentially upping the quality of the model.

More pictures were captured of the Elwood Bridge than any other bridge. The DroneDeploy app was not functioning properly on the day of the test, so the back-and-forth pattern was flown manually at four different heights: 1 foot, 3 feet, 8 feet, and 15 feet (0.30 m, 0.91 m, 2.44 m, and 4.57 m). This was followed by image capture during three oblique orbits at varying heights. Several close-up photos were taken of each side of the bridge, followed by images of the underside of the bridge taken just above the water level from each side.

4.4 Modeling Results

This section consists of three parts. First, an evaluation is provided of each modeling software examining specific strengths and weaknesses of each individual program. Second, models of each of the three bridges are compared qualitatively. Components unique to each bridge and how the software modeled them are addressed. Third, a quantitative assessment of the three brands of software is conducted using measurements taken at each bridge.

It should be noted that several models were created of each bridge by varying which photos were input into each software. For example, it was discovered that the Pix4D model of the Nibley Bridge looked most correct when lower-quality photos were

eliminated from the model. A good-faith effort was made to create the best bridge model using each software within the time constraints of the project. The models presented for each bridge are the “best” models created.

4.4.1 Evaluation of Individual Software Results

4.4.1.1 Pix4D

The most common error in Pix4D models was the creation of floating objects. These consisted of irregularly-shaped objects that did not exist in real life, but were created in the model. The floating objects could be floating in the air above the model or attached to the model itself. Pictures of several floating objects are shown in Figure 68.

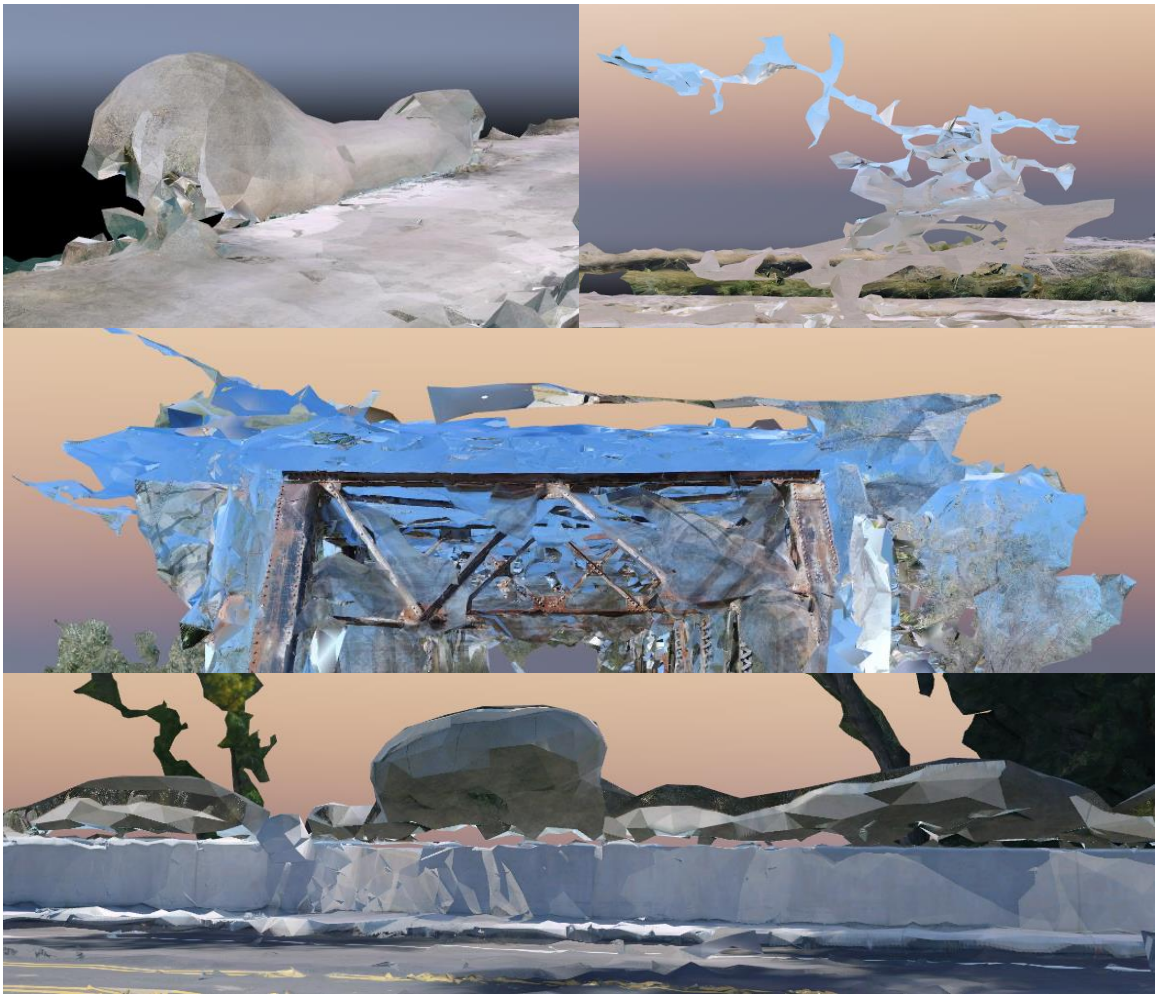

Figure 68. Floating objects from various Pix4D models.

Floating objects could be created by a variety of factors. Many were the result of including low-quality photographs in the model. These low-quality photographs typically included shots of the sky and/or horizon. As used in this paper, the term “horizon” includes the horizon line and objects a significant distance away from the bridge.

Floating objects were also often caused by water surfaces, because most water surfaces lack identifiable points for the software to reference. If the water was exceptionally still, the reflection of the bridge in the water could also cause floating objects.

Sometimes the floating objects were not the result of a discernable user error. Floating objects commonly occurred around skeletal objects like traffic poles and steel truss members. They occurred to a lesser extent where the Pix4D software had difficulty processing a planar surface or where an object protruded out of a smooth surface. Some floating objects simply reflect the limitations of using a computer algorithm to recreate a three-dimensional model from two-dimensional images.

Another large drawback to the Pix4D software was the amount of time it took to create a model. In order to create the best-looking model, the processing options were set to the highest-quality settings. Also, as many quality pictures as possible were used to create the model. Under these conditions, it typically took the software about 24 hours to completely process a model. Because the desktop version of Pix4D was used, when the software was processing a model it often consumed the majority of the available RAM on the computer, slowing the computer down significantly.

While it was usually able to recreate the shape of a structure, the Pix4D software would also frequently superimpose images in the wrong location. For example, when creating the model of the Nibley Bridge, the software superimposed an image of the underside of the bridge onto the exterior of the north concrete parapet (Figure 69). The incorrectly superimposed images were usually photos taken at an odd angle or from an unexpected location.

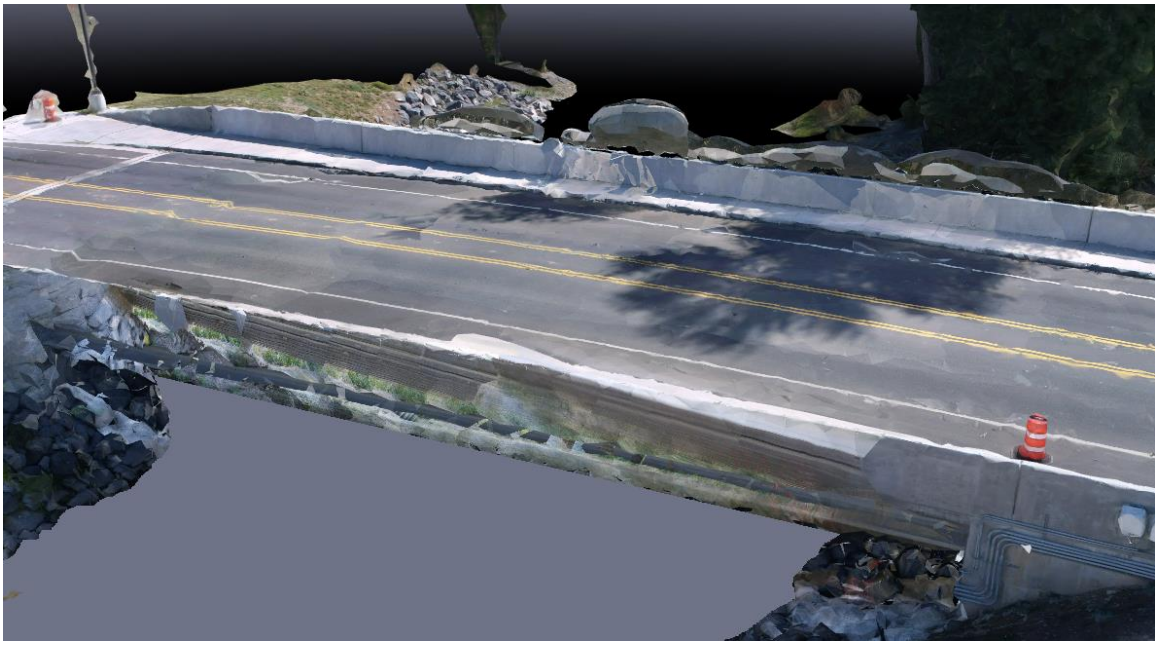

Figure 69. Incorrectly superimposed image on the Pix4D Nibley Bridge model.

The Pix4D software also struggled with creating planar surfaces. Where a smooth surface existed in real life, the modeling software would often create a surface that appeared choppy and disjointed. For example, the bridge deck in Figure 70 should appear relatively smooth. Instead, the model appears rough and uneven.

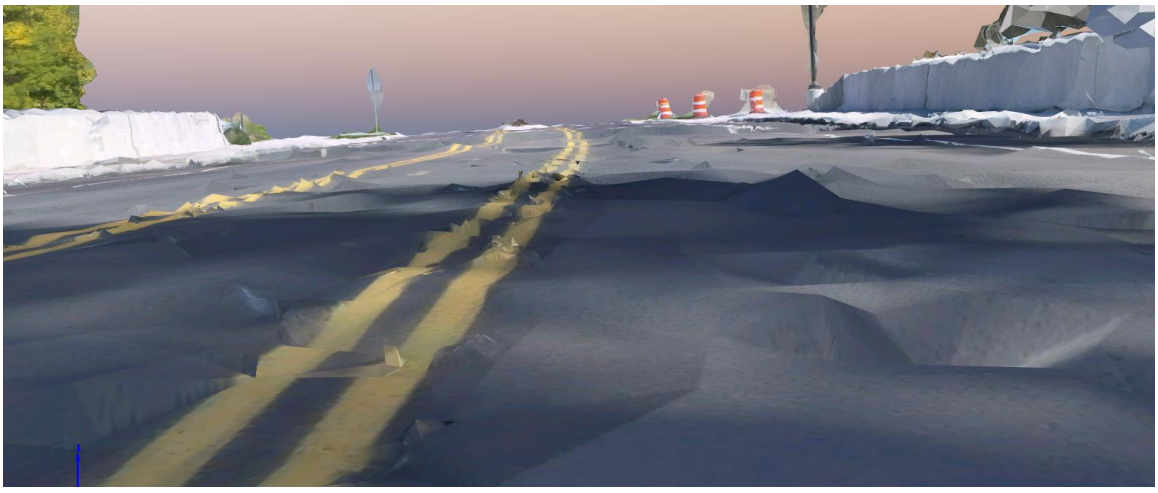

Figure 70. Deck Surface of the Pix4D Nibley Bridge model.

4.4.1.2 DroneDeploy

The DroneDeploy software, however, did particularly well modeling flat surfaces. Other software used often had difficulty when multiple points were coplanar. This would often lead to flat surfaces such as bridge decks being modeled appearing choppy and disjointed in the models. However, the DroneDeploy software did a relatively good job of modeling flat surfaces, as evidenced by its model of the Nibley Bridge deck in Figure 71.

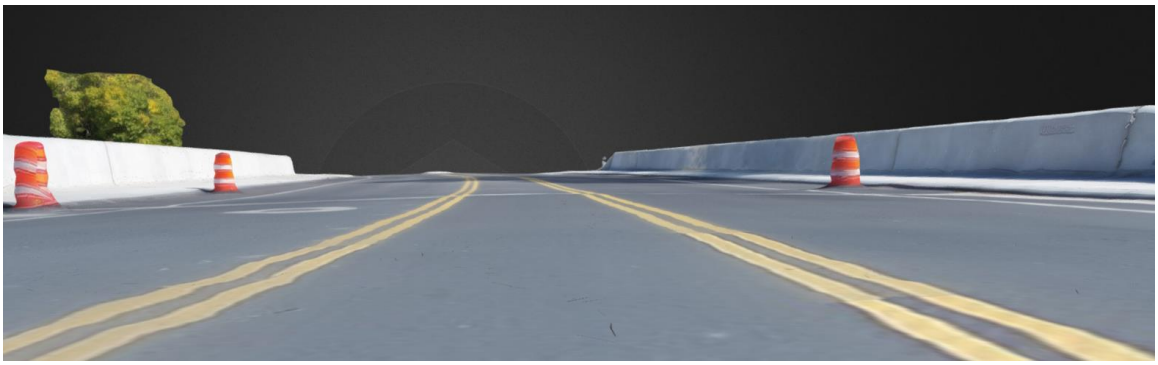

Figure 71. Flat Deck Surface of the DroneDeploy Nibley Bridge model.

However, DroneDeploy did have serious issues when it came to modeling skeletal structures. This included items such as metal traffic barriers, chain link fences, power poles, etc. If the item was less than about one foot (0.30 m) thick, the DroneDeploy software had difficulty creating a continuous member. Often, the software would simply eliminate the skeletal object from the model entirely.

This proved most problematic when modeling the Trenton Bridge. While the DroneDeploy software did better than expected modeling the lower truss members, it could not adequately model the skinnier lateral bracing and sway bracing at the top of the bridge. This is shown in Figure 72, which compares an actual photo of the Trenton Bridge on the left to the DroneDeploy model on the right.

Figure 72. Comparison of Trenton Bridge trusses to DroneDeploy model trusses.

Another area where the DroneDeploy software had problems was processing any water running beneath the bridge. Where water existed, the model would usually create disjointed blobs at various depths beneath the bridge (See Figure 73). This is likely due to the fluid nature of water, as well as the lack of identifiable points in water. If the water was exceptionally still, the software would also occasionally try to model an inverted bridge based off the bride's reflection in the water. While these issues alone do not hamper the ability to inspect a bridge using a model created from drone images, they do result in a messy-looking model.

Figure 73. Water underneath the DroneDeploy Elwood Bridge model.

One final modeling aspect that the DroneDeploy software handled well was recognizing that a gap existed between the underside of the bridge and the water below. Early attempts during this project to create models from drone photos often resulted in models where the sides of the bridge connected directly down into the water (See Figure 74). As the research progressed, experience piloting the drone combined with knowledge of best model photography practices resolved most of this issue. As shown later, the DroneDeploy software did the best job of recognizing the gap beneath the bridge and incorporating it into the model.

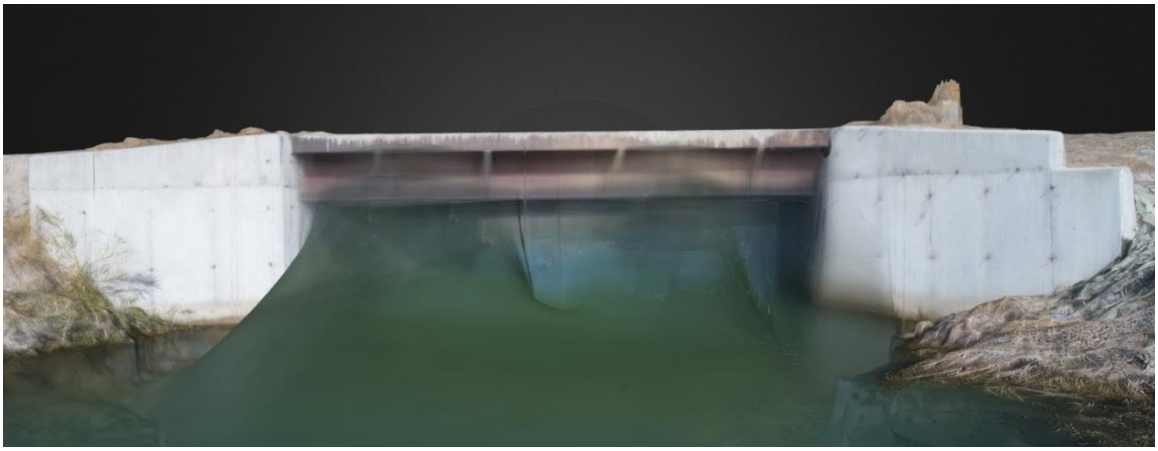

Figure 74. Early model of the Elwood Bridge.

4.4.1.3 Agisoft PhotoScan

The Agisoft PhotoScan software provided the highest surface resolution of the three modeling software programs. Whereas other programs created texture for their models by coloring the triangle mesh, Agisoft PhotoScan appears to have superimposed individual pixels onto the model. This gave the Agisoft PhotoScan software a crisp, high-resolution look. The increased resolution aided in creating lifelike models and was especially helpful when a specific area of the bridge needed to be closely inspected.

The Agisoft PhotoScan software struggled, however, at creating a successful gap underneath the bridge. This is likely due in part to the lack of easily identifiable points in the water beneath the bridge; any given point on the waters' surface is roughly identical to a nearby point. This theory is supported by Figure 75, which shows the Agisoft PhotoScan model of the Elwood Bridge.

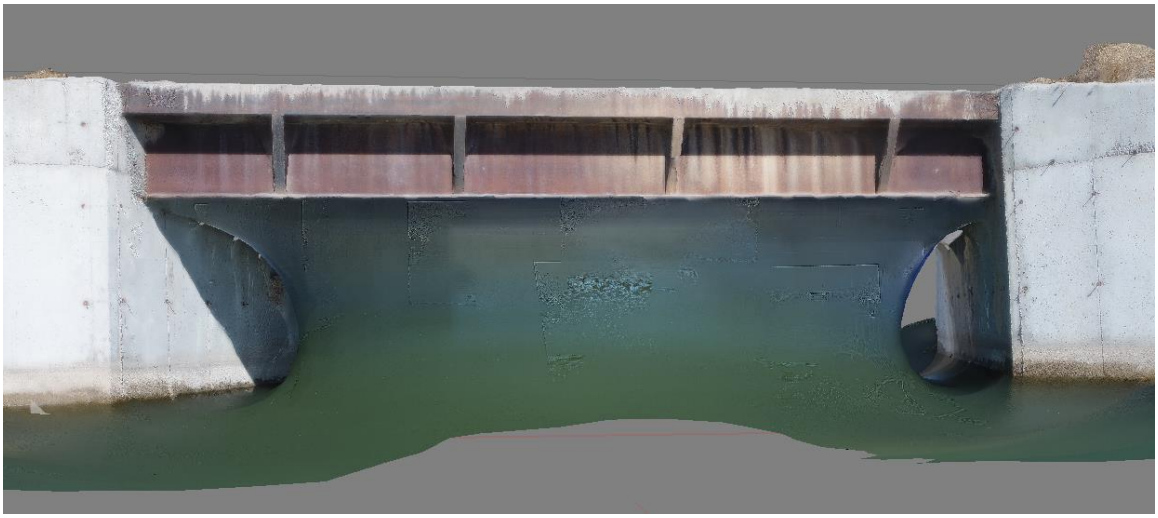

Figure 75. Water underneath the Agisoft PhotoScan Elwood Bridge model.

In Figure 75, a wall of water extends from the bottom of the bridge down to the canal's surface. The wall pulls away, however, from the bridge abutments. This is because the program is able to compare points across multiple photos to determine the location of abutments. However, the software has no way of telling how far away the water's surface is from the camera, because even from different angles all water points appear similar. Because the oblique photos show water directly adjacent to the bridge girder (See Figure 67 in Section 4.3.3), it assumes that the water must connect directly to the bridge girder. This creates the phenonemon shown in Figure 75.

Agisoft PhotoScan also had issues modeling flat water surfaces. However, unlike the DroneDeploy software, Agisoft PhotoScan was able to create a continuous water surface. Instead, the water surface was not planar; its elevation greatly varied. In Figure 76, the elevation of the water changes so that it forms a wave several feet high.

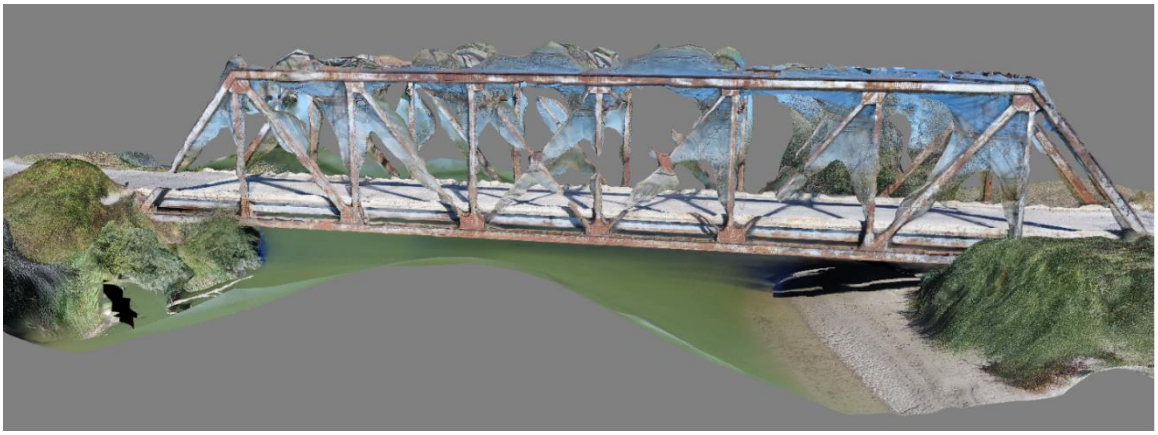

Figure 76. Agisoft PhotoScan Trenton Bridge model.

The Agisoft PhotoScan software also suffered from the same floating object problems as the Pix4D software. This included the incorporation of sky around skinny members (shown in Figure 76), as well as other types of floating objects. However, Agisoft PhotoScan's cropping feature allowed for the many of the floating objects to be removed. If the floating object was attached to the structure, however, cropping it out would leave a hole in the model where the floating object attached. For this reason, floating objects attached to the model were not cropped out.

4.4.2 Qualitative Comparison of Software Results

4.4.2.1 Nibley Bridge

The Nibley Bridge presented several unique challenges to the modeling software. The I-shape of the girders created a thin flange on the outside of the exterior of the girders for the software to model and made it near impossible to model the spaces between the interior girders. Meanwhile, the large shaded area underneath the bridge combined with the dense surrounding vegetation made it difficult to model the gap

between the underside of the bridge and the river below. Screenshots of the 3D models compared against a photo of the actual bridge are shown in Figure 77.

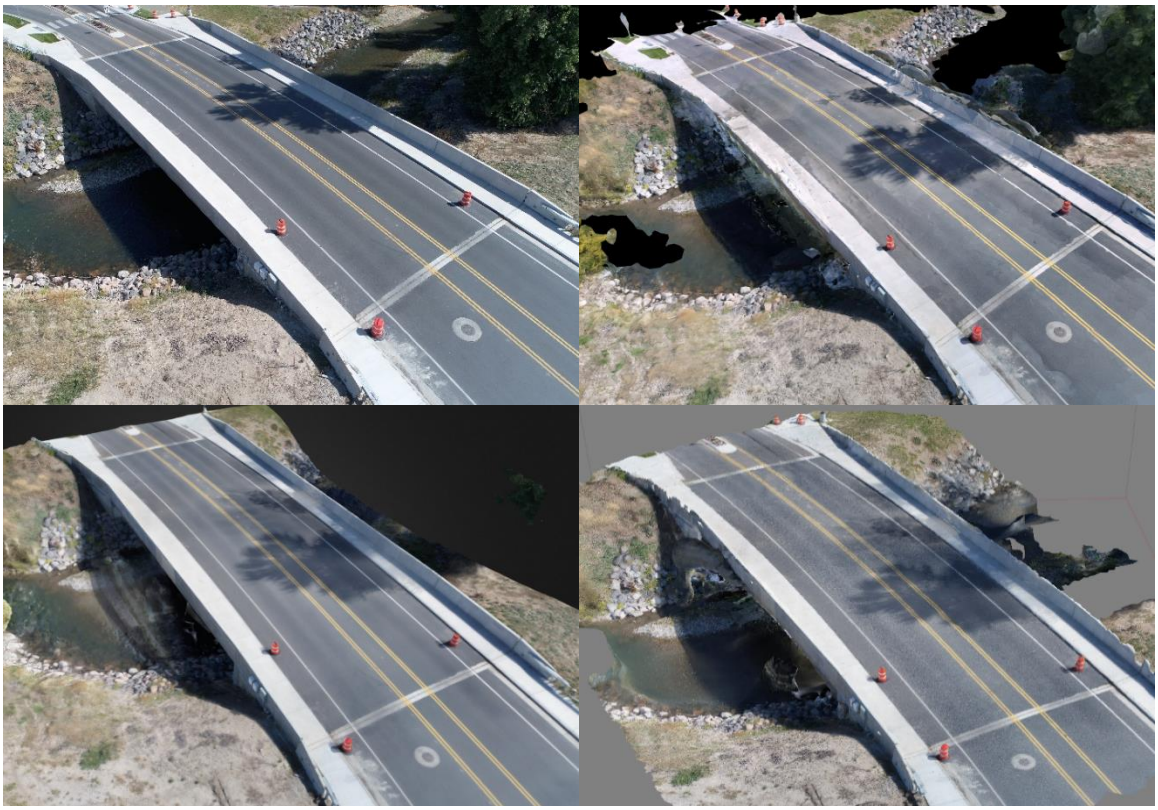

Figure 77. Photograph of the Nibley Bridge (top left) compared against screenshots of the Pix4D (top right), DroneDeploy (bottom left), and Agisoft PhotoScan (bottom right) models.

Each modeling software was able to recreate the bridge deck. However, the results of the underside of the bridge greatly varied. The DroneDeploy software was able to model the gap between the bridge and the river most successfully. The Pix4D and Agisoft PhotoScan software both struggled at successfully recreating this gap. Screenshots of the underside of the bridge from all three models are shown in Figure 78.

Figure 78. Screenshots of the underside of the Nibley Bridge Pix4D (top), DroneDeploy (center), and Agisoft PhotoScan (bottom) models.

As can be seen in Figure 78, the Pix4D model struggled significantly with floating objects, not only beneath the bridge but above the concrete parapets as well. It was discovered that the Pix4D software could adequately model the gap underneath the bridge if all pictures of the bottom of the bridge or of any portion of the sky were removed. However, the alternate model then contained no information about the underside of the bridge, defeating the purpose of creating the model.

The Agisoft PhotoScan software had issues with floating objects that extended from the underside of the bridge to the surface of the water. The DroneDeploy software, however, successfully recreated the basic shape of the underside of the bridge. While the DroneDeploy software did have some small floating objects, they were very minor compared to the other modeling software and did not impede inspection of the bottom of the bridge.

Because the drone camera could not look directly up into the gap between the individual girders, none of the models accurately recreated the shape of the gap shown in Figure 79. The Agisoft PhotoScan software was able to recognize that a space existed between the girders and attempted to recreate the gap. It was most successful with the girders closest to the edge, because the gap between these girders was documented best in drone photos. The DroneDeploy model simply modeled the underside of the bridge as a flat surface. The Pix4D software was able to recognize that gaps existed and attempted to recreate them; however, it was not as successful as the Agisoft PhotoScan software.

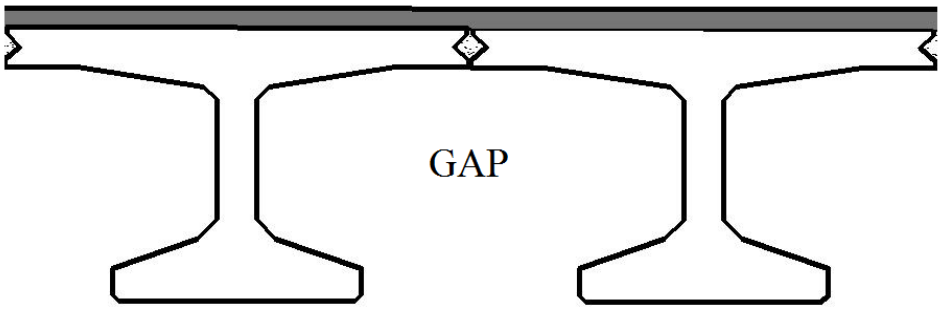

Figure 79. Gap between Nibley Bridge girders

Another qualitative metric was the recreation of the outside exterior girder. Figure 80 shows screenshots of Girder 10 (the southern exterior girder) from all three models.

While all three brands of software were able to recreate the basic shape of the flange, the quality of the finished product varied. DroneDeploy's ability to recreate smooth surfaces helped it create the most accurate model of the exterior girder, while the Agisoft PhotoScan software struggled to recreate the overhang of the top flange, giving it the least accurate exterior girder model.

Figure 80. Screenshots of Girder 10 from the Pix4D (top left), DroneDeploy (top right), and Agisoft PhotoScan (bottom left) Nibley Bridge models.

4.4.2.2 Trenton Bridge

The Trenton Bridge presented two main challenges to the modeling software. First, because the Trenton Bridge is a steel truss bridge, it is primarily composed of

skinny members. Thus, this bridge was a good test of the software's ability to model skinny members.

The second modeling challenge was human-caused. Because the Trenton Bridge was photographed relatively early in the research process, the majority of the oblique photos of the bridge contain large amounts of sky and/or horizon in them. It was learned later that photos of the sky cause large amounts of floating objects in the finished model.

Efforts to rephotograph the Trenton Bridge were hampered by winter. Because snow on the ground covers the identifiable points needed to create a model, a new model based on photos taken with snow on the ground would not be high quality. Thus, the decision was made to use the existing photographs to model the Trenton Bridge. Screenshots of all three models are shown in Figure 81.

Figure 81. Photograph of the Trenton Bridge (top left) compared against screenshots of the Pix4D (top right), DroneDeploy (bottom left), and Agisoft PhotoScan (bottom right) models.

For the Pix4D and DroneDeploy models, all pictures of the sky and horizon were removed prior to creating the model. However, it was discovered that the Agisoft PhotoScan software was better able to reconstruct the truss members if the photographs of the sky and horizon were included in the model. This is why the Agisoft PhotoScan model has more floating objects than models created using other software programs.

Removing photos containing the horizon resulted in fewer photographs available to create the Pix4D and DroneDeploy models. The number of photos used was already limited due to the short drone battery life of the Trenton Bridge flight. The limited battery life during this flight also meant that no pictures were taken of the underside of this bridge using the drone. Thus, none of the models contained an adequate reconstruction of the underside of the Trenton Bridge.

The skinniest members of the Trenton Bridge are the sway bracing and top lateral bracing members. Figure 82 shows a side-by-side comparison of the sway bracing, while Figure 83 shows a side-by-side comparison of the top lateral bracing. In both figures, it is apparent that each modeling software struggled modeling skinny members. However, each software modeled the skinny members in a different way.

Figure 82. Trenton Bridge sway bracing from a drone photograph (top) the Pix4D (top center), DroneDeploy (bottom center), and Agisoft PhotoScan (bottom) models.

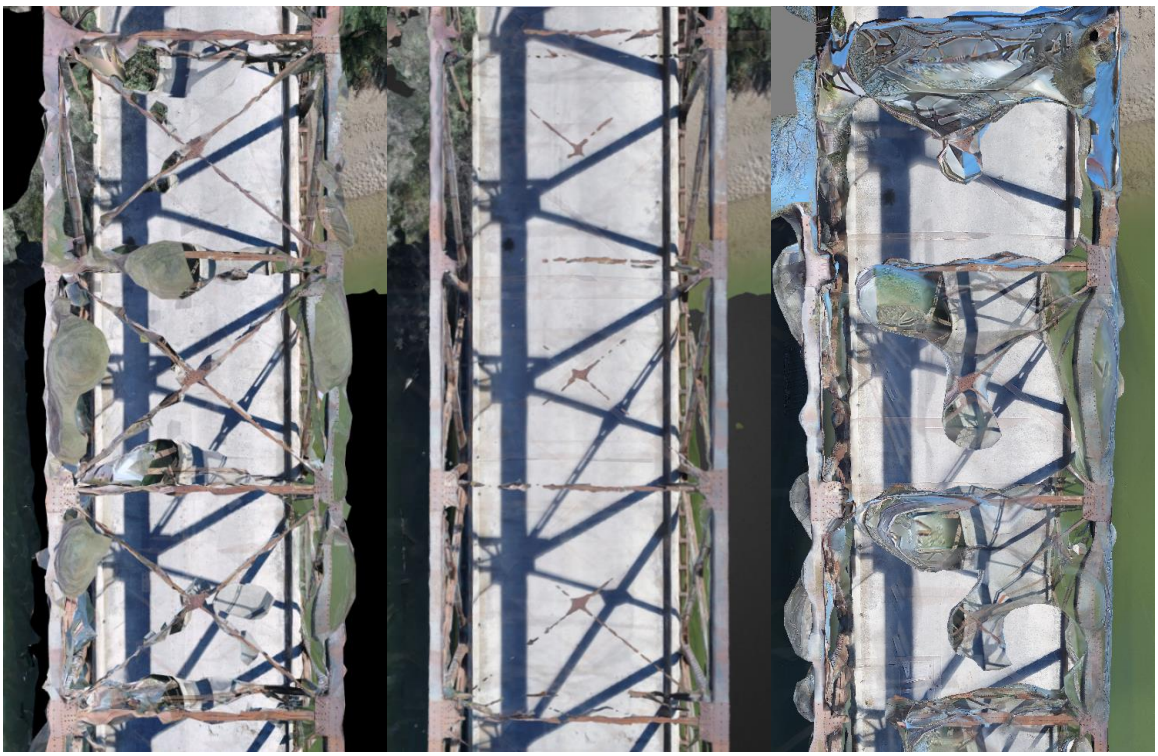

Figure 83. Trenton Bridge top lateral bracing from the Pix4D (left), DroneDeploy (center), and Agisoft PhotoScan (right) models.

The Pix4D software was able to recreate the basic shape of the sway and the top lateral bracing with the greatest accuracy. There was a limited quantity of floating objects encasing the members, and a cross-section of any given recreated member would not geometrically resemble the actual cross-section of the member. However, the program generally recognized that the truss consisted of several independent skinny members.

The DroneDeploy software, as noted in Section 4.4.1.2, struggled significantly to recreate continuous skinny members. In fact, it almost completely failed to recreate any portion of the interior sway brace shown in Figure 82. The software did better at recreating the bracing joints, where multiple members connected at a plate. Because the joints were wider than the truss members, the program was able to detect and model the bracing joints.

The Agisoft PhotoScan had an interesting approach to model the sway bracing. While it may appear from Figure 82 that the software has created a near-perfect replica of the sway bracing, the software did not actually model the individual truss members. Instead, the entire sway brace is a thin, rectangular slice spanning the width of the bridge. An image of the truss is then superimposed over the slice.

Similarly, the Agisoft PhotoScan software was not able to recreate the lateral truss bracing. Where it was able to detect the bracing, the software usually created an irregularly-shaped floating object. This is due partially because the Agisoft PhotoScan model incorporated pictures of the horizon, which can contribute to the creation of floating objects in the model.

Interestingly, all three versions modeling software overlaid “phantom” images of the bracing onto the concrete deck below. This is likely due to the fact that the bracing was very close to the drone in relation to the bridge deck. If the drone were to take a photo of the deck while hovering over a brace and move even a short distance, the brace would no longer be in the picture. Repetition of this phenomenon over time would lead to “disappearing” braces and result in the phantom image shown in Figure 84.

Figure 84. “Phantom” image of the top lateral bracing from the DroneDeploy Trenton Bridge model.

Finally, the vertical truss members of the Trenton Bridge provided an opportunity to evaluate each software’s ability to model larger skinny members. The vertical truss members of the Trenton Bridge are built-up members consisting of two steel channels connected by a lattice. A photograph of the vertical truss member is shown in Figure 85, alongside screenshots of vertical truss members from each modeling software.

Figure 85. Photograph of a Trenton Bridge vertical truss member (left) compared against screenshots of the Pix4D (center left), DroneDeploy (center right), and Agisoft PhotoScan (right) models.

The Pix4D software was able to recreate the vertical truss member most accurately. The resulting member looks correct in both shape and image overlay. The DroneDeploy software was able to model the vertical truss member more successfully than the skinner members in Figures 82 and 83. However, a cross-section of the resulting member would not match a cross-section of the actual member. The Agisoft PhotoScan software had the most difficulties recreating the vertical truss members. This is due in part to the images of the horizon included in the model. None of the software programs were able to recognize that the vertical truss members were in fact hollow and made of multiple connected steel components.

4.4.2.3 Elwood Bridge

The models of the Elwood Bridge were of particular interest for this research. Because it was the last bridge to be photographed, the best drone photography practices were used to capture the images that went into these models. Additionally, because the Elwood Bridge is much smaller than the other bridges, it had a significantly higher number of photos of each component of the bridge. It was therefore expected that the Elwood Bridge would have the highest-quality models of the three bridges selected. Figure 86 shows screenshots of each of the models compared to an actual photograph of the bridge.

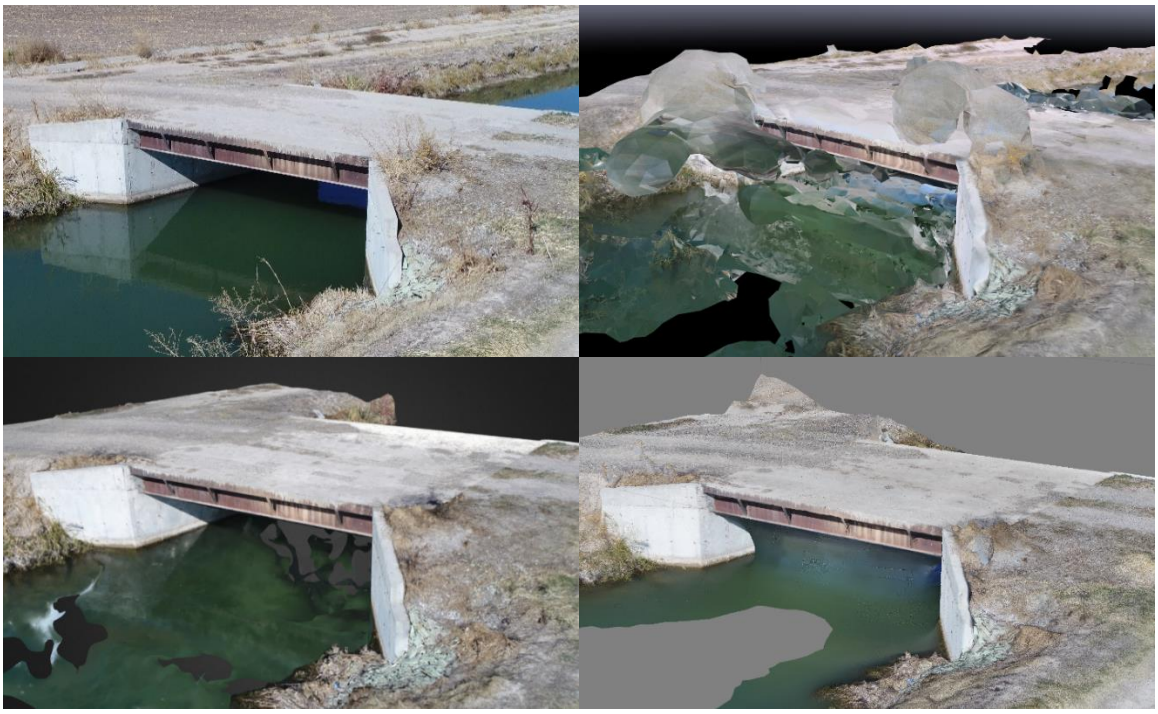

Figure 86. Photograph of the Elwood Bridge (top left) compared against screenshots of the Pix4D (top right), DroneDeploy (bottom left), and Agisoft PhotoScan (bottom right) models.

Because the Elwood Bridge was chosen specifically so that the drone could fly as close to the deck as possible, of specific interest was whether, in ideal conditions, the

modeling software would successfully retain the resolution of the drone photographs in the finished model. Figure 87 compares a portion of the deck from each model to a photograph taken while the drone hovered approximately three feet above the same area of the deck. The photograph shows a section of the concrete deck covered in patches of dirt and gravel and marked with tire tracks.

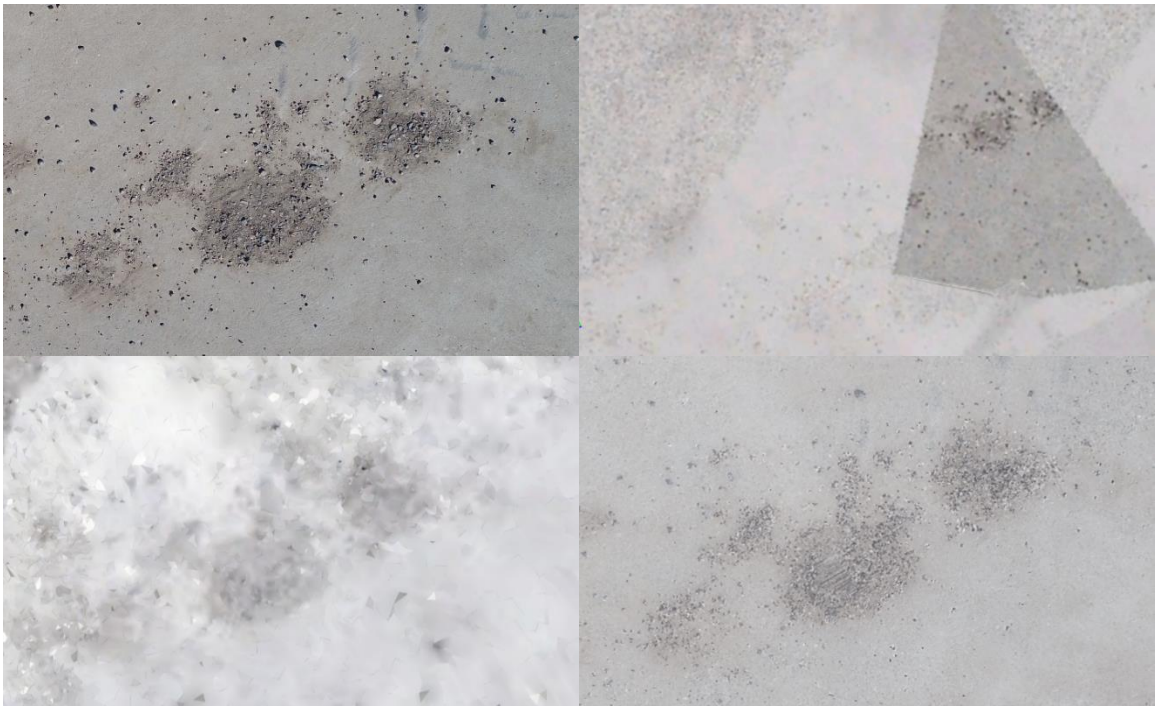

Figure 87. Photograph of the Elwood Bridge deck (top left) compared against screenshots of the deck from the Pix4D (top right), DroneDeploy (bottom left), and Agisoft PhotoScan (bottom right) models.

Figure 87 shows that there is an appreciable downgrade in resolution from photograph to model, even in the best model. As noted in Section 4.4.1, the Agisoft PhotoScan software was able to preserve more of the original resolution and texture of the drone images than other software programs. The DroneDeploy model was able to retain the basic shape of the original markings, but the image was segmented into fragments.

The Pix4D model experienced a total loss of resolution/texture because the software incorrectly superimposed an image of a different segment of deck over the analyzed location (a phenomenon described in Section 4.4.1). However, in other areas the Pix4D model had an intermediate resolution between the DroneDeploy and Agisoft PhotoScan models. When compared to models of the other bridges, flying the drone closer to the bridge deck did result in an appreciable upgrade in the surface resolution of the model.

Models of the underside of the Elwood Bridge had varied results, which are shown in Figure 88. The Pix4D software modeled the underside of the bridge as a rough, uneven surface with texturing that roughly approximated the actual bottom of the bridge. Limitations on the number of photos that could be uploaded to the DroneDeploy software meant that only the best pictures could be used in the model. This led to the discarding of the majority of the pictures of the underside of the bridge for this specific model. Because of this, the DroneDeploy software did not have sufficient data to reconstruct the underside of the Elwood Bridge. The Agisoft PhotoScan software was unable to recreate the bottom of the bridge for reasons noted in Section 4.4.1.3.

Figure 88. Screenshots of the underside of the Elwood Bridge Pix4D (top), DroneDeploy (center), and Agisoft PhotoScan (bottom) models.

The difficulty modeling the underside of the Elwood Bridge is thought to stem from two causes. First, the bridge superstructure is located only a few feet above the

water surface of the canal. This does not leave very much space for light to penetrate and illuminate the underside of the bridge girders. In the majority of pictures taken of the Nibley Bridge, the contrast between the shadow and bright sun makes the underside of the bridge appear extremely dark. This makes it difficult for the software to distinguish between individual points to create the 3D model.

Second, because the drone's camera could not look higher than a horizontal plane, virtually every photo of the underside of the bridge girders also contained shots of the horizon. These photos had to either not be used in the model or included in the model, where they would cause floating objects.

Finally, Figure 89 compares the models of the flange bracing along the exterior bridge girders. As shown by the top left image in Figure 89, the outer flanges on each edge of the bridge were periodically braced by thin strips of steel.

Figure 89. Photograph of the Elwood Bridge flange bracing (top left) compared against screenshots of the flange bracing from the Pix4D (top right), DroneDeploy (bottom left), and Agisoft PhotoScan (bottom right) models.

The Agisoft PhotoScan model was unable to recreate the gap between the bracing in the girder and instead inserted triangular-shaped wedges where they should be placed. The DroneDeploy model was able to recreate the flange bracing most accurately, consistently recreating the gap between the girder and brace. This allowed for the unique point of view shown in Figure 89. The Pix4D software achieved an intermediary result. All programs did better at modeling the flange bracing on the south side of the bridge, which was better illuminated than the flange bracing on the north side.

4.4.3 Quantitative Comparison of Software Results

4.4.3.1 Defects

As noted at the beginning of Section 4.3, one actual bridge defect was identified prior to photographing each bridge. Three artificial "cracks" were also created at each bridge by applying masking tape to the bridge surface and drawing cracks of various lengths and thicknesses onto the tape. Measurements were taken of the dimensions of both the real and artificial defects.

Once the models were finished, each model was inspected to determine whether the individual defects were visible in the finished model. It was found that the majority of the defects (69.4%) were either not present in the finished model or were too low-resolution to measure. The software did best at modeling defects if one of two conditions were met: 1) the drone flew exceptionally close to the bridge (i.e. the Elwood Bridge) or 2) the defect was exceptionally large (i.e. the pothole in the Trenton Bridge). No defects appeared in the models of the Nibley Bridge.

All six models with visible defects were exported into object (.obj) files. The object files were then loaded into MeshLab, a free three-dimensional modeling program. Loading the models into MeshLab allowed the models to be scaled to the correct size and measured. (While Pix4D has the ability to measure model dimensions in-program, DroneDeploy and Agisoft PhotoScan do not).

It was discovered that while each model was proportionally correct, the initially-measured distances in each model did not match the real-life dimensions of each bridge. To correct this, multiple real-life measurements from each bridge (typically the bridge

length) were compared against the corresponding measurements from each model. The ratios from the measurements were averaged for each model and used to scale the bridge to the appropriate size. These scale factors are shown in Table 6.

Table 6. Scale factors for each model.

Bridge/Software	Scale Factor
Trenton Bridge	
Pix4D	3.318
DroneDeploy	3.309
Agisoft PhotoScan	3.660
Elwood Bridge	
Pix4D	3.326
DroneDeploy	3.256
Agisoft PhotoScan	2.109

A total of five different defects were visible between the six models, one defect from the Trenton Bridge and all four defects from the Elwood Bridge. The Trenton Bridge's defect was a large pothole in the concrete deck located 48.46 feet (14.77 m) from the west abutment. The Elwood Bridge's defects consisted of 1) a numeral "1" etched into the concrete deck (the last numeral in the date), 2) an artificial crack drawn with a pen and taped to the bridge deck, 3) an artificial crack drawn with a marker and taped to the bridge deck, and 4) an artificial crack drawn with a marker and taped to the southeast wing wall. Photographs of all five defects are shown in Figure 90.

Figure 90. Defects in the Trenton and Elwood Bridges; Defect 1: a pothole (top left); Defect 2: a numeral 1 (top right); Defect 3: a pen-drawn crack (bottom left); Defect 4: a marker-drawn crack (bottom center); Defect 5: a marker-drawn crack (bottom right).

The length of each defect was measured in each model. The results are given in Table 7. However, not all defects were present in every model. Where Table 7 reads "N/A", the defect was not present in the finished model.

Table 7. Comparison of bridge defect measurements

Defect	Actual Length	Pix4D	DroneDeploy	Agisoft PhotoScan
Trenton Bridge				
Defect 1 (Pothole)	27 inch (68.58 cm)	27.67 in (70.29 cm)	27.79 inch (70.59 cm)	27.52 inch (69.89 cm)
Elwood Bridge				
Defect 2 (Numeral)	3.37 inch (8.56 cm)	2.37 inch (6.02 cm)	3.4 inch (8.64 cm)	3.18 inch (8.07 cm)
Defect 3 (Crack)	1.92 inch (4.88 cm)	N/A	N/A	1.91 inch (4.86 cm)
Defect 4 (Crack)	2.23 inch (5.66 cm)	N/A	1.68 inch (4.26 cm)	2.24 inch (5.7 cm)
Defect 5 (Crack)	2.1 inch (5.33 cm)	2.17 inch (5.52 cm)	N/A	2.1 inch (5.34 cm)

The wider the defect, the better the software was at recreating that defect in the three-dimensional model. This is demonstrated by the fact that all three software models were able to recreate Defects 1 and 2, the largest defects. Only two software models contained Defects 4 and 5, which were only 0.05 inches (0.13 cm) thick. Only one software model contained Defect 3, which was only 0.02 inches (0.04 cm) thick.

It should be noted that because 5.5 months passed between taking the drone photographs and measuring the finished model, there was some ambiguity as to where the pothole's (Defect 1's) measurements were originally taken from. However, all three models were measured from the same points. Given the precision of the measurements for Defect 1 in Table 7, it is probable that they are accurate.

Another issue to note is that the Pix4D software modeled the pothole (Defect 1) as a smooth surface without any depth. While the depth of the pothole was not measured as part of this research, a successful recreation of this defect would include a representation of its depth. The DroneDeploy and Agisoft PhotoScan software both modeled the pothole with depth. The width of the pothole was not measured on the models.

It should also be noted that Defects 1 and 4 were partially obscured in the DroneDeploy models. The location of the beginning of the pothole was approximated for Defect 1. For Defect 4, the measurement was taken as it appeared on the model. This is why the DroneDeploy measurement for Figure 71 is shorter than the real-life measurement. It is unknown why the Pix4D measurement of Defect 2 is shorter than the real-life measurement.

Table 7 clearly demonstrates that, so long as the model is able to recreate a defect discernably, it is possible to get accurate measurements from a model created using drone imagery. In the majority of circumstances, the measurement was off by less than 0.1 inches (0.25 cm). On average, the model measurement only differed from the actual measurement by 2.06%.

However, there are several factors that could potentially impede accurate measurements on a model outside of a controlled research setting. First, the two major outliers in the data set, Pix4D's measurement of Defect 2 and DroneDeploy's measurement of Defect 4, differed from their respective real-life values by 29.62% and 24.81%. For this research, real-world measurements were taken and compared against the model measurements to test whether they were accurate. In practice, there would be no way of knowing the accuracy of a model measurement.

Second, the same researcher who took the original measurements also measured the models. This could have potentially skewed the bridge measurements towards being more accurate, as all of the data was in one place and being compared against each other.

Third, even when the defects could be measured, they were usually very low-resolution. Determining exactly where the defect began and ended was extremely difficult. This is shown in Figure 91, which contains screenshots of some of the defects measured in various models.

Figure 91. Screenshots of defects from each model; Defect 3, Agisoft PhotoScan model (left); Defect 4, DroneDeploy model (center); Defect 5, Pix4D model (right).

As noted when discussing the Elwood Bridge models in Section 4.4.2 and demonstrated by Figure 91, there was a noticeable loss of resolution when the software converted still images into a three-dimensional model. Often the shape of the crack was fragmented into a small cluster of pixels.

Usually, the skinniest defects were only discernable because it was known beforehand that the defects existed and where they were at on the bridge. It is questionable if even an experienced bridge inspector could have located the defects on the model without prior knowledge of their location. The loss of image resolutions remains one of the chief roadblocks to recreating and accurately measuring defects on a three-dimensional model.

4.4.3.2 Coins

A dime, a penny, a nickel, and a quarter were also placed face up on the bridge deck prior to photographing the bridge. This allowed for a side-by-side evaluation of whether the software could recreate specific details of the bridge, and how high the resolution of the recreated details would be. The modeled coins are shown in Figures 92, 93, and 94, where they are compared against a drone photograph of the same coins. The

coins appear in the order stated above. If a software's results are not included in a figure, that software did not recognizably recreate the coins in its model.

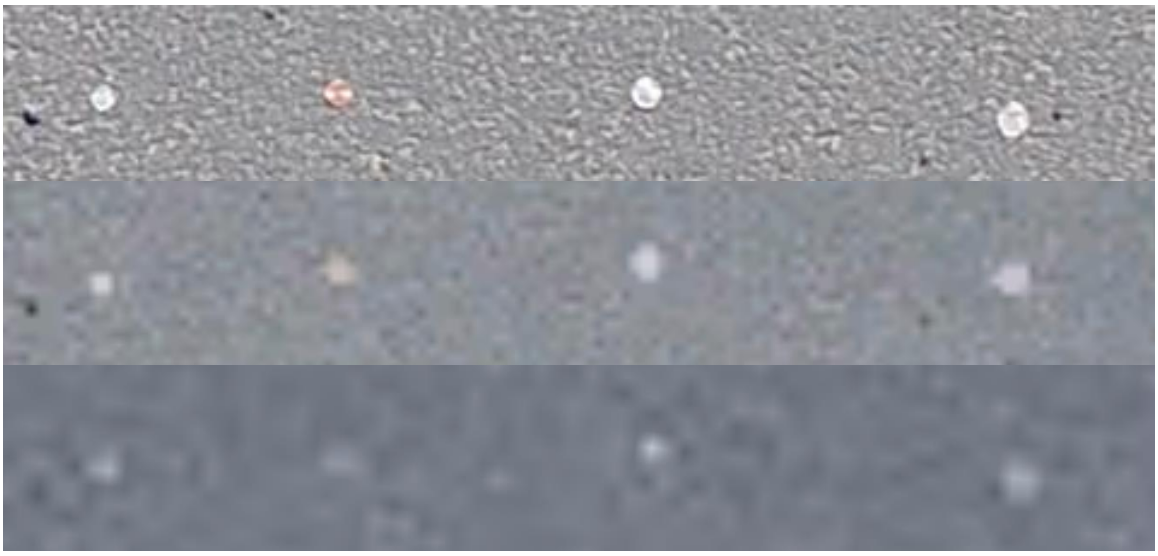

Figure 92. Drone photograph of coins placed on the Nibley Bridge (top); coins from the Pix4D model (center) and DroneDeploy model (bottom).

Figure 93. Drone photograph of coins placed on the Trenton Bridge (top); coins from the Pix4D model (center) and DroneDeploy model (bottom).

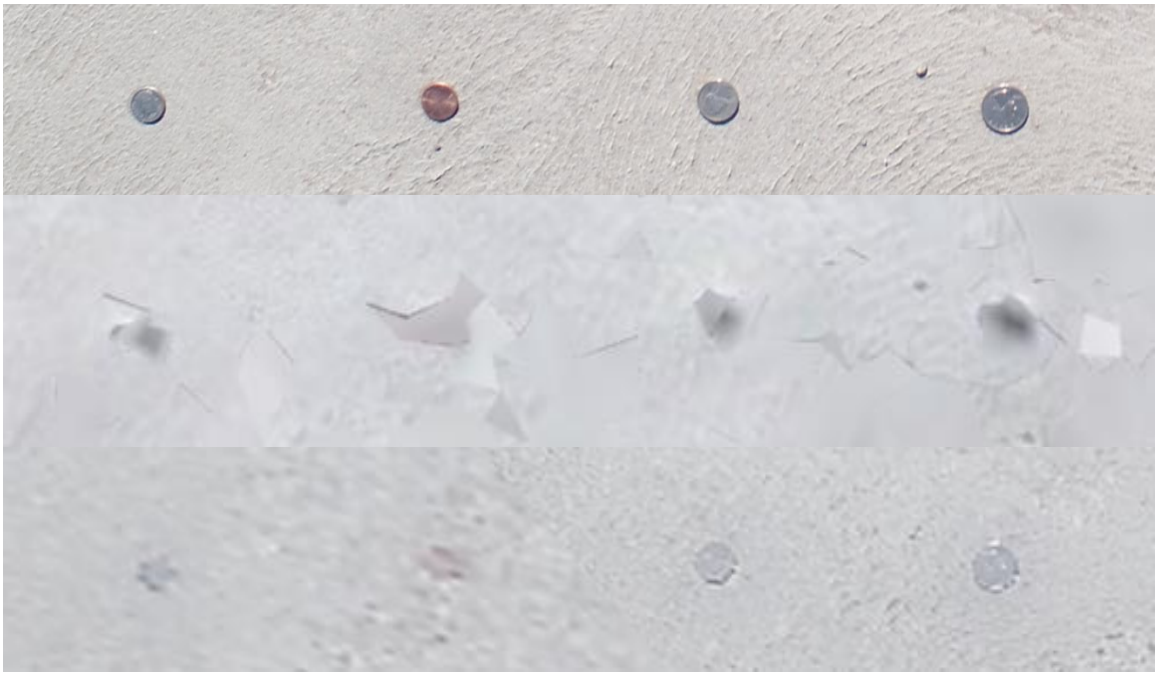

Figure 94. Drone photograph of coins placed on the Elwood Bridge (top); coins from the DroneDeploy model (center) and Agisoft PhotoScan model (bottom).

As can be seen in the figures, the coin models are most accurate and detailed in the Elwood Bridge models and least accurate in the Trenton Bridge models. Recall from Section 4.3 that the drone flew 25 feet (7.62 m) above the Nibley Bridge deck, 35 feet (10.67 m) above the Trenton Bridge deck, and 1 foot (0.30 m) above the Elwood Bridge deck (the photograph in Figure 94 was taken at 3 feet (0.91 m)). Flying the drone closer to the deck not only resulted in a better quality drone photograph; it also allowed the models to recreate the coins at a higher resolution as well. Hence, the distance the drone took pictures above the deck significantly influenced the resolution of the coins in the model.

4.5 Drone Modeling Conclusions and Recommendations

4.5.1 Conclusions

The primary conclusion of this research is that models based off of images captured by standard, commercially available drones are not yet suitable for bridge inspections. This is based off of the models' inability to recreate the majority of identified defects and the significant loss of resolution between the drone images and the finished models. Until these issues are resolved, three-dimensional models will be unable to attain the level of measurement accuracy necessary to conduct a bridge inspection.

However, this research also concludes that the existing technology is heading in the right direction. Currently, three-dimensional modeling of bridges is a new field, one that is growing and changing rapidly. Much of the technology used in this research was not available a decade ago. As the industry grows to meet this niche field, the price of drones built specifically to conduct bridge inspections is expected to drop. Modeling software can also be developed that is specifically tailored to model bridges. This will address and resolve several of the difficulties encountered in this research.

There are two primary roadblocks that modeling software must overcome before three-dimensional models can be used to aid bridge inspections. First, the modeling software must be able to accurately recreate the shape of bridge components. This not only includes accurately recreating members' shapes (See Figure 85 in Section 4.4.2.2); the software needs to stop creating floating objects. This is because floating objects make viewing the bridge difficult and warp bridge members to the point they can no longer be inspected.

Second, the software must be able to recreate 3D-model textures accurately and at higher resolutions than at present. Because the model texture is how the drone photographs are superimposed onto the blank model, if the model texture is wrong than it can result in a defect being the wrong size or being omitted from the model entirely. If the model texture is too low-resolution, it makes it extremely difficult to locate defects on the model and obtain accurate measurements.

Until modeling technology reaches the capability to better process drone imagery of bridges, drones can still be used independently of modeling software to aid in bridge inspections. While the screenshots of the bridge models were usually low-resolution, the raw drone images usually clearly documented bridge defects.

It is anticipated that drones will be most helpful when used in conjunction with other bridge inspection methods. They are particularly well suited for situations where one or more bridge components are difficult to access. This is especially true on large bridges with long spans (as opposed to the relatively short bridges considered in this study). Consider, for example, the Perrine Bridge near Twin Falls, Idaho, shown in Figure 95.

Figure 95. Perrine Bridge (Owens, n.d.).

The Perrine Bridge is approximately 1,500 feet long (457.2 m) long and stands 486 feet (148.13 m) above the Snake River. Its main arch is too far below the deck for under bridge inspection vehicles to access. The arch could be examined by lowering inspectors down on ropes; however, this carries human risk. A drone could inspect components of the bridge arch both easily and safely. For this reason, drones are particularly well suited to inspect these types of large bridges.

The primary difficulty of using pure drone imagery is that it is difficult to measure defect dimensions from a single photograph. To calculate defect dimensions from a single photograph, one must know the distance between the drone's camera and the defect. Alternately, the ratio between the defect and an object in the photograph of known dimensions could be used. Because neither the drone-to-defect distance nor a ratio is typically known, at present drones are best used in conjunction with other established bridge inspection methods.

4.5.2 Recommendations

4.5.2.1 Recommendations for Drone Selection

The following recommendations are intended to aid in the selection of a drone for a bridge inspection project.

- Having a drone camera that can look directly upwards should be considered paramount for a drone bridge inspection project. Without it, it is near impossible to collect good data about the underside of the bridge's superstructure.
- Consideration should be given into the quality of the camera attached to the drone. It should be ascertained that the drone camera will capture images at high-enough resolution to adequately display bridge defects.
- Consideration should be given into purchasing extra batteries for the drone. More batteries allow the operator more flight time at each location. More flight time allows the drone to take more photos, which can upgrade the quality and accuracy of the model.
- Having a drone with sensors that prevent it from crashing into obstacles is very useful. This is because it allows the drone operator to take close-up photos of the bridge while minimizing the chance that the drone will fly to close and crash into the structure. It also can be cost-effective, as it helps prevent costly drone replacements.
- Some drone models will automatically return to home if they lose GPS signal. If this feature is included in a drone, it should be verified that the feature can be

turned off to prevent a drone from flying directly upwards into an obstacle (such as the bottom of a bridge deck).

4.5.2.2 Recommendations for Bridge Suitability

Before beginning a drone bridge inspection project, the suitability of the bridge for drone image capture should be determined. This includes evaluations of whether a drone will be able to adequately capture images of bridge components, as well as whether the software will be able to create a quality model of the bridge. The following recommendations are intended to help in this evaluation.

- The minimum altitude of the drone flight above the bridge deck should be considered. The higher the drone flies above the bridge deck, the lower resolution photographs of the deck will be. The minimum altitude above the bridge deck is typically dictated by the obstacles surrounding the bridge, including vehicular traffic, overhead bridge trusses, vegetation, power poles, and buildings. If the minimum altitude of the flight is too high above the bridge deck to provide an adequate photograph resolution, the bridge is not suitable for drone image capture.
- If heavy traffic regularly crosses the bridge, the bridge will need to be closed while the drone flies over the deck in order to comply with federal aviation law. This is because federal aviation law does not permit drones to fly directly above vehicles in motion or pedestrians not directly involved in the drone's flight.
- If vegetation exists near the bridge, it will be difficult to maneuver the drone to capture pictures. It may be impossible to capture photos from some angles. If the

vegetation is flush with the bridge, the drone will be unable to capture up-close images of that area. Where vegetation is present, caution should be taken when allowing an app to control the drone and automatically take pictures of the bridge. This recommendation also applies to areas with overhead obstacles (power lines, street lamps, etc.).

- If the drone images will be included in a model, the area surrounding the bridge should be free of large patches of snow. Photographs containing large areas of snow will cause several of the same problems that pictures of the water or sky cause in 3D models.
- Bridges made of large, basic shapes (e.g. box girder bridges) will generally have better models than bridges with skinny or complicated members (e.g. truss bridges or girder bridges).

4.5.2.3 Recommendations for Image Capture

The following recommendations apply to drone operators and the end-users of 3D-modeling software. The quality of the model is most often a direct function of the drone photography practices that captured the images included in the model. If followed, these recommendations are intended to result in the best-quality model of a given bridge.

- As noted in Section 4.5.1, the quality of the photos taken by the drone is far more important than the quantity of pictures taken. Including sub-standard photos in a 3D model usually had a greater negative impact than including more photos had a positive impact. Using too many photographs in a model noticeably increased the model processing time as well. The photos to be included in a model should be

reviewed prior to beginning the modeling process, and low-quality photos should be removed.

- Pictures taken with a drone should only contain stationary objects. If an object moves while photos are being taken, the modeling software will identify the object in one photo but will be unable to find the object in the same location in subsequent photos. This can confuse the software and lead to errors in the model. Pictures containing moving objects (including vehicles, pedestrians, and the drone operator) should be removed prior to modeling.
- Care should be taken to avoid taking photographs that include the sky and/or horizon in the background. As defined earlier, the "horizon" consists of the horizon line and objects that are a considerable distance away from the bridge. Including photos containing the sky or horizon in a model greatly increases the number of floating objects. Except where absolutely necessary, images containing shots of the sky or horizon should be removed prior to modeling the bridge.
- Photos that are overexposed or contain sun glare (see Figure 96) should be removed prior to starting the modeling process. The differences in shade and contrast between overexposed and regular pictures can confuse the modeling software in a similar manner as moving objects. This also applies to images containing sun glare.

Figure 96. Overexposed photograph of the Elwood Bridge containing sun glare.

- Drone pictures should be taken from the proper distance away from the object. If the drone is too far away, the resolution of the finished model will be low. If the drone is too close to the object, there may not be enough identifiable points in the photograph to help orient the picture in the model.
- When flying the drone manually, it is possible (using DJI's in-flight control app) to have the drone automatically take a photo every two seconds. It is highly recommended that the drone operator use this capability because it speeds up the image capture process, allows the drone operator to focus on flying the drone, and captures more photos per flight.
- Oblique images are extremely important to creating a successful model of a bridge. These images allow the modeling software to more accurately recreate the sides of the bridge, model the gap underneath the bridge, and recreate the area underneath an overhang. While flying the drone, care should be taken to capture oblique pictures from multiple viewpoints and angles.

- If a bridge defect is readily apparent, the drone operator should capture as many detailed photos of the defect as possible. This includes close-up photos of the defect, as well as photos that show where the defect is located in relation to other bridge components.
- The drone operator should take care to capture photos of every part of the bridge. This includes photos of the underside of the bridge. If too few photos exist of a bridge component, it will not be included in the model. Capturing photos of bridge components from different angles is helpful as well.
- When using the DroneDeploy app to automatically capture overhead photos, increasing the overlap between the photos can result in a higher-quality model. However, increasing the overlap too high can result in the drone essentially hovering over the same spot, taking photos without moving.
- It was observed that the three software programs rarely used pictures taken with the Nikon D90 camera in the model, even when they were included in the model. This was because the photos taken using the Nikon D90 camera were taken at variable focal lengths and were not geotagged. If supplemental pictures are taken with a camera, a camera with a constant focal length (no zoom lens) and geotagging capabilities should be used.

4.5.2.4 Recommendations for Modeling Software Improvement

The following recommendations are intended to be ways in which the modeling software itself could be improved. They are intended for companies producing the software rather than the end-user.

- The two largest areas for improvement, eliminating floating objects and improving model resolution, were evaluated in detail in Section 4.5.1. These conclusions will not be reexamined in this section. However, these remain the two areas in need of greatest improvement.
- While images of many bridge components can be taken without including the horizon in the model, for some components including the horizon in photographs is unavoidable (e.g. when capturing images of the underside of the bridge). Because of this, the software could be further improved to process images containing the sky and horizon in the background without creating as many errors in the model.
- Another area for improvement is the software programs' ability to model water. Whether the modeling software connected the water to the underside of the bridge, created floating objects where the water should be, or failed to recreate the water altogether, none of the software programs used in this research were able to accurately recreate water surfaces if the water was more than a few feet deep. While measures can be taken to avoid including sky in the background of the model, it is inevitable that pictures of bridges spanning water will contain images of the water. Thus, modeling software for bridges needs to be capable of handling water in a model.
- The software could do better at recognizing that a gap exists between the superstructure of the bridge and the river beneath. Often, especially in early models, the software was unable to recognize that the gap existed (see Figures 74

and 75 in Section 4.4.1). Software designed for bridge modeling could begin the model with the assumption that a gap exists, allowing for a better model.

- The software could be improved to better recognize and model skinny bridge members (for example, trusses).
- Another area where the software was often deficient was creating smooth surfaces in the model. If a smooth surface was choppy (as in Figure 70 of Section 4.4.1.1), the software was already starting with an inaccurate base model before the texture was ever applied. This could potentially result in distortion of defects and bridge geometry.
- One useful feature would be the ability to annotate defects and other problem areas in-software. This feature already exists in third-party software. Figure 97 shows an image from an archaeology site model uploaded to Sketchfab, an online software for sharing 3D models. Important areas are easily identified with numbers. Clicking on the numbers zooms in on the area and provides a short, user-provided description.

Figure 97. Screenshot of Ortli excavation site (low-poly) from Sketchfab (agisoft 2014).

This feature would be tremendously useful for bridge inspections. The ability to annotate where defects are located would allow notes to be maintained about the location and extent of the defects. It would also allow for easier monitoring of defects over time as new models are created. The annotation could also be linked to all of the images in which that defect appears for evaluation of the higher-quality drone images.

- Another useful feature would be the ability to measure three-dimensional models in the software that they were originally produced. This would mean that models would not need to be exported to MeshLab in order to be measured, simplifying and expediting the measuring process.
- The ability to crop a finished model was one of the most useful features of the Agisoft PhotoScan software. A good bridge modeling software would include this feature, as it allows for non-essential features such as vegetation to be removed

from the model. Removing non-essential features allows for easier inspection of the bridge model because the removed features do not block views bridge components.

- As explained in Section 4.1.3, the DroneDeploy Android app was used to map the bridge deck looking directly down from above. With the advent of sensors that can tell how close the drone is to the ground, drone mapping apps could also include flying patterns that vary the drone's height, flying the drone up-and-down to map the side of a bridge while looking horizontally.

4.5.2.5 Recommendations for Regulatory Improvement

As noted in the Minnesota Department of Transportation's (MnDOT) 2015 research project, Federal Aviation Administration (FAA) regulations on drones can be restrictive. While the process to become registered to fly a drone has been simplified over the last several years, several obstacles remain to the widespread use of drones to inspect bridges. This includes the following:

- Flying drones for research purposes is treated the same as commercial use and is regulated by 104 CFR Part 107. This can be overly restrictive because it requires researchers to obtain a remote pilot certificate before they can fly a drone. The remote pilot certification process can take a significant amount of time and includes a test, application, and a background check. The amount of time spent trying to become certified can be problematic, especially if a research project is on a tight schedule. In the case of this project, more time was spent trying to get

the certificate (3 weeks) than was actually spent flying the drone (about 12 hours including practice flights).

- 104 CFR 107.39 forbids flying drones over pedestrians and moving vehicles. While it is possible to receive a Part 107 Waiver, only seven companies in the United States have been able to obtain one. Without a waiver, flying a drone above a bridge deck necessitates that no traffic be on the bridge. For bridges with regular use, this necessitates either a partial or a full bridge-closure, which negatively affects the economic advantages of using drones for bridge inspection. For bridges with less traffic, the drone operator must halt the drone flight and move the drone away from the bridge whenever a vehicle approaches. This can interrupt the automatic flight program and unnecessarily uses battery life. It can also be difficult to fly the drone to a safe distance in the space of time between when a vehicle comes into view and when the vehicle crosses the bridge.

CHAPTER 5

CONCLUSION

5.1 Summary of Conclusions

5.1.1 Summary of Conclusions for Short-Term Strain Monitoring Research

Based on the measured data, the traffic on the Nibley Bridge regularly exceeded HS-20 truck loading, with recorded strains of up to 59.26 με. The largest loading events approached (but did not exceed) the HL-93 design loads. Because the monitoring took place over a relatively narrow span of time, it was theorized that the maximum strains experienced by the bridge girders could be even higher than the measured values.

It was discovered that the largest strains were typically concentrated into a small number of days per week. On any given day, the largest strains were of similar magnitudes, and the shape of the strain events indicated that the strains were caused by vehicles with similar axle configurations (See Figure 32). Because of these similarities, it was theorized that the largest strain events on any given day were likely caused by the same vehicle crossing the Nibley Bridge multiple times that day.

It was found that the girders immediately under the traffic lanes experienced the highest magnitude of strains. Adjacent girders experienced progressively less strain the farther they were placed from the loaded traffic lane. The girders farthest from the loaded traffic lane (the exterior girders and girders underneath in the opposite lane) typically experienced little to no strain.

As expected, the distribution of strain events followed established traffic patterns, with the number of strain events picking up throughout the day and decreasing at night.

The highest number of strain events occurred just before school started and right after school let out. During these periods, multiple vehicles used the bridge at once, resulting in higher strains in the bridge girders.

The majority of vehicles (95.62%) caused less than 5.0 με of strain in any girder on the bridge. These values reflect smaller vehicles crossing the Nibley Bridge; vehicles smaller than a pickup truck typically caused a strain of 2.42–3.45 με. School buses typically caused a strain of around 13.65 με, while heavily laden construction equipment caused the largest strains but had highly variable strain magnitudes.

5.1.2 Summary of Conclusions for Drone Modeling Research

The drone modeling area of the research concluded that standard, commercially available drones can be an effective tool to aid in bridge inspections. The effectiveness of the drone depends on the size and length of the bridge, as well as how difficult it is to inspect bridge members using other methods. Other factors such as heavy vegetation and high traffic levels also affect whether using a drone would be appropriate for a bridge inspection project.

It was also concluded that three-dimensional models created using images captured using a non-specialized drone are not yet suitable for bridge inspections. This conclusion is primarily due to the low resolution of the models created for this project. The low model resolution made it difficult to locate and measure bridge defects. The low resolution also often meant that defects were either not present or were indiscernible in the finished model.

The conclusion about the suitability of bridge models for bridge inspections was also due to the programs' inability to accurately model some bridge members. The creation of "floating objects" added geometry to the bridge that did not actually exist. It also distorted some bridge members and obscured others from view. Some software failed to model flat surfaces accurately, while others could not accurately recreate skinny members.

The final conclusion of this area of research is that both drone and 3D modeling technology is heading in the right direction. The rapid growth of this industry may address several of the issues raised in this research over the next several years. As the market grows, drones and software specializing in inspecting and modeling bridges should be developed. Future developments in this field could allow for the accurate creation and measurement of bridge models.

5.2 Recommendations for Additional Research

5.2.1 Recommendations for Additional Short-Term Strain Monitoring Research

- The data used in this study was collected over a five-week span. Future research could collect data over a longer period in order to confirm and expand upon this research.
- Continuous video monitoring could be used to increase the sample sizes of large vehicles shown in Table 3 in Section 3.3.5.5.
- The live load test data collected in Section 3.3.5.4 could be analyzed using a finite-element model. The results could be used to verify the conclusions drawn in

Section 3.3.5.4 and improve the accuracy of the girder distribution factors presented in Section 3.3.5.7.

- Future research could place additional strain transducers on Girder 4 in order to verify the readings of Sensor B1978.

5.2.2 Recommendations for Additional Drone Modeling Research

- The majority of this research focused on using drone imagery to create a bridge model, then conducting a bridge inspection on said model. Future research could instead focus on identifying and measuring defects from raw drone imagery.
- Only three bridges were modeled as part of this research. Future research could examine a greater variety of bridge types, span lengths, etc.
- A licensed bridge inspector could inspect a bridge prior to capturing drone imagery. Once a 3D model has been created, a separate licensed bridge inspector could inspect the bridge model. The results of the two inspections could then be compared.
- Future research could attempt to model larger bridges (such as the Perrine Bridge) in order to evaluate the feasibility of using drones to inspect and model larger bridges.
- This research focused on inspecting bridges using a standard, non-specialized, commercially available drone. Tests could be conducted using a drone specifically designed for bridge inspections and compared against this research.

- Future research could place “artificial” easily identifiable points around the bridge in the form of colored cones in order to examine whether this would improve the quality of the 3D model.
- The drone imagery from this research was captured around midday. Future research could examine whether capturing drone imagery closer to dawn/dusk would reduce the contrast underneath the bridge and allow for easier modeling of the underside of the bridge.
- Future research could determine recommended limits for the maximum/minimum number of photos per model.
- Because this field is advancing rapidly, this research should be revisited periodically to determine whether new technology has resolved the problems encountered in this research.

REFERENCES

agisoft. (2014). “Ortli excavation site (low-poly).” Sketchfab, New York, NY.

Dà-Jiāng Innovations. (2017). Phantom 4 Pro. *www.dji.com, photograph,* Dà-Jiāng Innovations Science and Technology Co., Ltd, Shenzhen, China.

Dà-Jiāng Innovations. (2017). Remote Controller with Built-in Screen. *www.dji.com, photograph,* Dà-Jiāng Innovations Science and Technology Co., Ltd, Shenzhen, China.

Dà-Jiāng Innovations. (2018). “DJI Phantom 4 Pro – Specs, FAQ, Tutorials and Downloads.” DJI Official Website, Dà-Jiāng Innovations, <https://www.dji.com/phantom-4-pro/info#specs> (Mar. 28, 2018).

DroneDeploy Labs. (2017). “DroneDeploy.” *DroneDeploy*, computer software, Infatics, Inc.

Hodson, D. J., Barr, P. J., and Halling, M. W. (2012). “Live-Load Analysis of Posttensioned Box-Girder Bridges.” *Journal of Bridge Engineering*, 17(4), 644–651.

Lattanzi, D., and Miller, G. R. (2015). “3D Scene Reconstruction for Robotic Bridge Inspection.” *Journal of Infrastructure Systems*, 21(2), 04014041–1-04014041–12.

Ma, Z. (J.), Chaudhury, S., Millam, J. L., and Hulsey, J. L. (2007). “Field Test and 3D FE Modeling of Decked Bulb-Tee Bridges.” *Journal of Bridge Engineering*, 12(3), 306–314.

Nowak, A. S., and Collins, K. R. (2013). Reliability of Structures. CRC Press, Boca Raton, FL.

Owens, P. (n.d.). *Perrine Bridge. https://visitidaho.org/,* photograph, Idaho Department of Commerce - Tourism, Boise, ID.

Parrot S.A. (2017). Parrot AR Drone GPS Edition. photograph.

Pickett, E. (2017). “The Short and Long-Term Effects of Temperature and Strain on a Concrete Bulb-Tee Girder Bridge.” thesis.

Schwarz, M., and Laman, J. A. (2001). “Response of Prestressed Concrete I-Girder Bridges to Live Load.” *Journal of Bridge Engineering*, 6(1), 1–8.

Zink, J., and Lovelace, B. (2015). *Unmanned Aerial Vehicle Bridge Inspection Demonstration Project.* Minnesota Department of Transportation, Research Services & Library, St. Paul, MN.

APPENDICES

APPENDIX A

Event Extraction VBA Program

```
Sub ExtractEvents()
      Dim Threshold As Double
      Dim MaxStallTime As Double
      Dim NormalDataPath As String
      Dim EventDataPath As String
      Dim TestSpecificationName As String
      Dim PointsPerSecond As String
      Dim FileLength As Double

      Dim mybook As Workbook, BaseWks As Worksheet, CalcMode As Long
      Dim MyFiles() As String, FilesInPath As String

      Dim TimeFormula As String
      Dim NumberEvents As Double
      Dim FNum As Double
      Dim StartCell As Double
      Dim StopCell As Double
      Dim StartTime As Double
      Dim FirstEvent As Double
      Dim EventRow As Double
      Dim Filename As String
      Dim BeforeEventTime As Double
      Dim AfterEventTime As Double
      Dim CompareTime As Double
      Dim IgnoreTime As Double


'*****************************************************************
'*****************************************************************

      Threshold = Worksheets("Main").Range("D16").Cells(1, 1).Value
      MaxStallTime = Worksheets("Main").Range("D17").Cells(1, 1).Value
      BeforeEventTime = Worksheets("Main").Range("D18").Cells(1,
            1).Value
      AfterEventTime = Worksheets("Main").Range("D19").Cells(1,
            1).Value
      CompareTime = Worksheets("Main").Range("D20").Cells(1, 1).Value
      IgnoreTime = Worksheets("Main").Range("D21").Cells(1, 1).Value
      NormalDataPath = Worksheets("Main").Range("D24").Cells(1,
            1).Value
      EventDataPath = Worksheets("Main").Range("D25").Cells(1, 1).Value
      TestSpecificationName = Worksheets("Main").Range("D31").Cells(1,
            1).Value
      PointsPerSecond = Worksheets("Main").Range("D32").Cells(1,
            1).Value
      FileLength = Worksheets("Main").Range("D33").Cells(1, 1).Value
      TimeofYear = Worksheets("Main").Range("D35").Cells(1, 1).Value
```

```
'*****************************************************************
'*****************************************************************

      FileLength = PointsPerSecond * 3600 * FileLength + 1
      MaxStallTime = MaxStallTime * PointsPerSecond
      BeforeEventTime = BeforeEventTime * PointsPerSecond
      AfterEventTime = AfterEventTime * PointsPerSecond
      CompareTime = CompareTime * PointsPerSecond
      IgnoreTime = IgnoreTime * PointsPerSecond

      ' Add a slash after path names if needed.
      If Right(NormalDataPath, 1) <> "\" Then
            NormalDataPath = NormalDataPath & "\"
      End If
      If Right(EventDataPath, 1) <> "\" Then
            EventDataPath = EventDataPath & "\"
      End If

      ' If there are no TDMS files in the folder, stop program.
      FilesInPath = Dir(NormalDataPath & "*.tdms*")
      If FilesInPath = "" Then
            MsgBox "No TDMS files found"
            Exit Sub
      End If

      ' Fill the myFiles array with the list of TDMS files in the
            folder.
      FNum = 0
      Do While FilesInPath <> ""
            FNum = FNum + 1
            ReDim Preserve MyFiles(1 To FNum)
            MyFiles(FNum) = FilesInPath
            FilesInPath = Dir()
      Loop

      ' Change application properties.
      With Application
            CalcMode = .Calculation
            .Calculation = xlCalculationManual
            .ScreenUpdating = False
            .EnableEvents = False
            .CalculateBeforeSave = False
      End With

      ' Loop through all files in the myFiles array.
      If FNum > 0 Then
            For FNum = LBound(MyFiles) To UBound(MyFiles)
                  Set mybook = Nothing
                  On Error Resume Next
                  On Error GoTo 0

                  '**********************Open TDMS File
                        Code**********************
                  'Get TDM Excel Add-In
```

```
Dim obj As COMAddIn
Set obj =
      Application.COMAddIns.Item("ExcelTDM.TDMAddin")

'Confirm only importing "Description" properties for
      Root
Call obj.Object.Config.RootProperties.DeselectAll
Call
      obj.Object.Config.RootProperties.Select("Descri
      ption")

'Show the group count as property
Call
      obj.Object.Config.RootProperties.Select("Groups
      ")

'Select all the available properties for Group
Call obj.Object.Config.GroupProperties.SelectAll

'Import custom properties
obj.Object.Config.RootProperties.SelectCustomProperti
      es = True
obj.Object.Config.GroupProperties.SelectCustomPropert
      ies = True
obj.Object.Config.ChannelProperties.SelectCustomPrope
      rties = True

'Import the selected file
Call obj.Object.ImportFile(NormalDataPath &
      MyFiles(FNum))

'Record down the current workbook & Select Correct
      Sheet
Set mybook = ActiveWorkbook
'Range("D3") = (Left(Range("D2").Cells(1, 1).Value,
      Len(Range("D2")) - 4))
Range("A3") = (Left(Range("A2").Cells(1, 1).Value,
      Len(Range("A2")) - 29))
Range("A3") = (Right(Range("A3").Cells(1, 1).Value,
      Len(Range("A3")) - 14))
Range("A3") = Replace(Range("A3").Cells(1, 1).Value,
      "_", "/", , 2)
Range("A3") = Replace(Range("A3").Cells(1, 1).Value,
      "_", " ", , 1)
Range("A3") = Replace(Range("A3").Cells(1, 1).Value,
      "_", ":", , 2)
Range("A3").Select
Selection.NumberFormat = "0.00"

StartTime = Range("A3")
ActiveSheet.Next.Select
StartCell = 1
StopCell = 1
```

```
'*************************************************
      ************

'Prepare Times
TimeFormula = "=M2+1/3600/24/" & PointsPerSecond
Worksheets(TestSpecificationName).Range("M2").Cells(1
      , 1).Value = StartTime
Worksheets(TestSpecificationName).Range("M3").Cells(1
      , 1).Value = TimeFormula
Range("M3").Select
Selection.AutoFill Destination:=Range(Cells(3, 13),
      Cells(FileLength, 13)), Type:=xlFillDefault


'Insert Event Detection Formula & Accidental Trigger
      Prevention
Worksheets(TestSpecificationName).Range(Cells(FileLen
      gth + 1, 2), Cells(FileLength + 1 + IgnoreTime,
      9)).Value = -999
ActualCell = 2 + CompareTime

'Start Formula
Worksheets(TestSpecificationName).Range("S1").Cells(A
      ctualCell, 1).Value = "=MIN(B2:B" & ActualCell
      & ")"
Range("S1").Cells(ActualCell, 1).Select
Selection.AutoFill
      Destination:=Range(Cells(ActualCell, 19),
      Cells(ActualCell, 26)), Type:=xlFillDefault
Range(Cells(ActualCell, 19), Cells(ActualCell,
      26)).Select
Selection.AutoFill
      Destination:=Range(Cells(ActualCell, 19),
      Cells(FileLength, 26)), Type:=xlFillDefault

Worksheets(TestSpecificationName).Range("AB1").Cells(
      ActualCell, 1).Value = "=B" & ActualCell & "-S"
      & ActualCell
Range("AB1").Cells(ActualCell, 1).Select
Selection.AutoFill
      Destination:=Range(Cells(ActualCell, 28),
      Cells(ActualCell, 35)), Type:=xlFillDefault
Range(Cells(ActualCell, 28), Cells(ActualCell,
      35)).Select
Selection.AutoFill
      Destination:=Range(Cells(ActualCell, 28),
      Cells(FileLength, 35)), Type:=xlFillDefault

Worksheets(TestSpecificationName).Range("Q1").Cells(A
      ctualCell, 1).Value = "=IF(OR(AB" & ActualCell
      & ">" & Threshold & ",AC" & ActualCell & ">" &
      Threshold & ",AD" & ActualCell & ">" &
      Threshold & ",AE" & ActualCell & ">" &
      Threshold & ",AF" & ActualCell & ">" &
```

```
      Threshold & ",AG" & ActualCell & ">" &
      Threshold & ",AH" & ActualCell & ">" &
      Threshold & ",AI" & ActualCell & ">" &
      Threshold & "),1,0)"
Worksheets(TestSpecificationName).Range("K1").Cells(A
      ctualCell, 1).Value = "=IF(AND(Q" & ActualCell
      & "=1,SUM(Q" & ActualCell - IgnoreTime & ":Q" &
      ActualCell & ")=1),1,0)"


Range("Q1").Cells(ActualCell, 1).Select
Selection.AutoFill
      Destination:=Range(Cells(ActualCell, 17),
      Cells(FileLength, 17)), Type:=xlFillDefault
Range("K1").Cells(ActualCell, 1).Select
Selection.AutoFill
      Destination:=Range(Cells(ActualCell, 11),
      Cells(FileLength, 11)), Type:=xlFillDefault

Range("K1").Cells(1 + CompareTime, 1).Value = "Start
      Triggers (Filtered)"
Range("L1").Cells(1 + CompareTime, 1).Value = "Stop
      Trigger"
Range("Q1").Cells(1 + CompareTime, 1).Value = "All
      Start Triggers"
Range("R1").Cells(1 + CompareTime, 1).Value = "
      Minimums"
Range("AB1").Cells(1 + CompareTime, 1).Value =
      "Differences"


'Stop Formula
Worksheets(TestSpecificationName).Range("L1").Cells(A
      ctualCell, 1).Value = "=IF(OR(B2-B" & 2 +
      CompareTime & ">" & Threshold - 0.25 & ",C2-C"
      & 2 + CompareTime & ">" & Threshold - 0.25 &
      ",D2-D" & 2 + CompareTime & ">" & Threshold -
      0.25 & ",E2-E" & 2 + CompareTime & ">" &
      Threshold - 0.25 & ",F2-F" & 2 + CompareTime &
      ">" & Threshold - 0.25 & ",G2-G" & 2 +
      CompareTime & ">" & Threshold - 0.25 & ",H2-H"
      & 2 + CompareTime & ">" & Threshold - 0.25 &
      ",I2-I" & 2 + CompareTime & ">" & Threshold -
      0.25 & "),-1,0)"
Worksheets(TestSpecificationName).Range("L1").Cells(A
      ctualCell, 1).Select
Selection.AutoFill
      Destination:=Range(Cells(ActualCell, 12),
      Cells(FileLength, 12)), Type:=xlFillDefault
Worksheets(TestSpecificationName).Range("A1").Cells(F
      ileLength, 12).Value = -1

Worksheets(TestSpecificationName).Range("N1").Cells(1
      , 1).Value = "# Events"
```

```
                    Worksheets(TestSpecificationName).Range("N2").Cells(1
                          , 1).Value = "=SUM(K:K)-N5"
                    Worksheets(TestSpecificationName).Range("N4").Cells(1
                          , 1).Value = "# Duplicate Triggers"
                    Calculate

                    NumberEvents =
                          Worksheets(TestSpecificationName).Range("N2").C
                          ells(1, 1).Value
                    'MsgBox ("Initial Calculations Complete for File " &
                          FNum & ". " & vbCrLf & "Approximately " &
                          NumberEvents & " Events Found.")

                    'Finalize Times
                    Range(Cells(2, 1), Cells(FileLength, 1)) =
                          Range(Cells(2, 13), Cells(FileLength,
                          13)).Value
                    Range(Cells(2, 13), Cells(FileLength, 13)).Clear

                    'Find an Event Start
                    Range("K1").Cells(StartCell, 1).Select
                    Columns("K:K").Select
                    Columns("K:K").Find(What:="1", After:=ActiveCell,
                          LookIn:=xlValues, LookAt:=xlPart,
                          SearchOrder:=xlByRows, SearchDirection:=xlNext,
                          MatchCase:=False, SearchFormat:=False).Activate
                    StartCell = ActiveCell.Row
                    Range("L1").Cells(StartCell, 1).Select
                    'Columns("L:L").Select
Repeat:
                    Columns("L:L").Find(What:="-1", After:=ActiveCell,
                          LookIn:=xlValues, LookAt:=xlPart,
                          SearchOrder:=xlByRows, SearchDirection:=xlNext,
                          MatchCase:=False, SearchFormat:=False).Activate
                    StopCell = ActiveCell.Row
                    If StopCell < StartCell Then GoTo Repeat

                    FakeEventCounter = 0
                    FirstEvent = StartCell
                    EventRow = StartCell
                    BaseValue1978 = Cells(StartCell - CompareTime,
                          2).Value
                    If TimeofYear = "Summer" Then
                          BaseValue1985 = Cells(StartCell - CompareTime,
                                6).Value
                    Else
                          BaseValue1985 = Cells(StartCell - CompareTime,
                                8).Value
                    End If
                    BaseValue1988 = Cells(StartCell - CompareTime,
                          9).Value
                    mybook.Sheets(TestSpecificationName).Activate

                    'Copy Event Data into New Files
```

```
Do
                'Copy Event Data
                MyRange = 0
                If StopCell - StartCell + BeforeEventTime
                      < MaxStallTime Then
                      MyRange = Range(Cells(StartCell -
                            BeforeEventTime, 1),
                            Cells(StopCell +
                            AfterEventTime, 9))
                      MinValueRange =
                            Range(Cells(StartCell, 19),
                            Cells(StartCell, 26))
                Else
                      MyRange = Range(Cells(StartCell -
                            BeforeEventTime, 1),
                            Cells(StartCell +
                            MaxStallTime -
                            BeforeEventTime, 9))
                      MinValueRange =
                            Range(Cells(StartCell, 19),
                            Cells(StartCell, 26))
                End If

                'Create & Set Up New Workbook
                Set BaseWks =
                      Workbooks.Add(xlWBATWorksheet).Work
                      sheets(1)
                Worksheets("Sheet1").Range("A1").Cells(1,
                      1).Value = "Time"
                Worksheets("Sheet1").Range("A1").Cells(1,
                      2).Value = "B1978"
                Worksheets("Sheet1").Range("A1").Cells(1,
                      3).Value = "B1979"
                Worksheets("Sheet1").Range("A1").Cells(1,
                      4).Value = "B1980"
                Worksheets("Sheet1").Range("A1").Cells(1,
                      5).Value = "B1984"
                Worksheets("Sheet1").Range("A1").Cells(1,
                      6).Value = "B1985"
                Worksheets("Sheet1").Range("A1").Cells(1,
                      7).Value = "B1986"
                Worksheets("Sheet1").Range("A1").Cells(1,
                      8).Value = "B1987"
                Worksheets("Sheet1").Range("A1").Cells(1,
                      9).Value = "B1988"
                Worksheets("Sheet1").Range(Cells(1, 1),
                      Cells(1, 9)).Font.Bold = True
                Worksheets("Sheet1").Range("K1").Cells(1,
                      1).Value = "Mins:"
                Worksheets("Sheet1").Range("K2").Cells(1,
                      1).Value = "=B2-K$1"
                Worksheets("Sheet1").Range(Cells(1, 11),
                      Cells(1, 18)).Font.Italic = True
```

```
                        Worksheets("Sheet1").Range("K1").Cells(1,
                              1).HorizontalAlignment = xlRight

                        'Paste Data Into New Workbook
                        If StopCell - StartCell < MaxStallTime
                              Then
                              BaseWks.Range(Cells(2, 1),
                                    Cells(StopCell - StartCell +
                                    BeforeEventTime +
                                    AfterEventTime + 1,
                                    9)).Select
                              BaseWks.Range(Cells(2, 1),
                                    Cells(StopCell - StartCell +
                                    BeforeEventTime +
                                    AfterEventTime + 1, 9)).Value
                                    = MyRange
                        Else
                              BaseWks.Range(Cells(2, 1),
                                    Cells(MaxStallTime + 1,
                                    9)).Select
                              BaseWks.Range(Cells(2, 1),
                                    Cells(MaxStallTime + 1,
                                    9)).Value = MyRange
                        End If
                        BaseWks.Range(Cells(1, 11), Cells(1,
                              18)).Value = MinValueRange
                        BaseWks.Range("A1").Cells(BeforeEventTime
                              + 2, 1).NumberFormat = "mm/dd/yyyy
                              hh:mm:ss.ss;@"
                        If Filename =
                              Replace(Replace(CStr(BaseWks.Range(
                              "A1").Cells(BeforeEventTime + 2,
                              1)), "/", "-"), ":", ";") Then
                              Filename =
                                    Replace(Replace(CStr(BaseWks.
                                    Range("A1").Cells(BeforeEvent
                                    Time + 2, 1)), "/", "-"),
                                    ":", ";") & " (2)"
                        Else
                              Filename =
                                    Replace(Replace(CStr(BaseWks.
                                    Range("A1").Cells(BeforeEvent
                                    Time + 2, 1)), "/", "-"),
                                    ":", ";")
                        End If

                        'Save And Exit
                        BaseWks.SaveAs Filename:=EventDataPath &
                              Filename,
                              FileFormat:=xlOpenXMLWorkbook
                        BaseWks.Activate
                        ActiveWorkbook.Close savechanges:=False

0:                  'Next Workbook
```

```
mybook.Sheets(TestSpecificationName).Acti
      vate
Range("K1").Cells(StartCell, 1).Select
Columns("K:K").Find(What:="1",
      After:=ActiveCell,
      LookIn:=xlValues, LookAt:=xlPart,
      SearchOrder:=xlByRows,
      SearchDirection:=xlNext,
      MatchCase:=False,
      SearchFormat:=False).Activate
StartCell = ActiveCell.Row
If StartCell - EventRow <=
      PointsPerSecond * 10 And StartCell
      - EventRow > -1000 Then
      ComparisonValue1978 =
            Cells(StartCell, 2).Value
      If TimeofYear = "Summer" Then
            ComparisonValue1985 =
                  Cells(StartCell,
                  6).Value
      Else
            ComparisonValue1985 =
                  Cells(StartCell,
                  8).Value
      End If
      ComparisonValue1988 =
            Cells(StartCell, 9).Value
      BaseValue1978 = Cells(StartCell,
            2).Value
      If TimeofYear = "Summer" Then
            BaseValue1985 =
                  Cells(StartCell, 6 +
                  17).Value
      Else
            BaseValue1985 =
                  Cells(StartCell, 8 +
                  17).Value
      End If
      BaseValue1985 = Cells(StartCell, 9
            + 17).Value

      If ComparisonValue1978 -
            BaseValue1978 > Threshold *
            1.6 Or ComparisonValue1985 -
            BaseValue1985 > Threshold *
            1.6 Then  'Megaevent
            Protection
            FakeEventCounter =
                  FakeEventCounter + 1
            GoTo 0
            ElseIf ComparisonValue1978 -
                  BaseValue1978 + 0.2 <
                  ComparisonValue1988 -
                  BaseValue1988 And
```

```
                                    ComparisonValue1985 -
                                    BaseValue1985 + 0.2 <
                                    ComparisonValue1988 -
                                    BaseValue1988 Then
                                    'Sidewalk Sensor
                              FakeEventCounter =
                                    FakeEventCounter + 1
                              GoTo 0
                        'ElseIf
                              WorksheetFunction.Max(Cells(E
                              ventRow, 2).Value,
                              ComparisonValue1978) -
                              WorksheetFunction.Min(Range(C
                              ells(EventRow, 2),
                              Cells(StartCell, 2))) < 0.5
                              And
                              WorksheetFunction.Max(Cells(E
                              ventRow, 7).Value,
                              ComparisonValue1985) -
                              WorksheetFunction.Min(Range(C
                              ells(EventRow, 7),
                              Cells(StartCell, 7))) < 0.5
                              Then    'Prevents the tail
                              end of the event being
                              classified as an event
                              'FakeEventCounter =
                                    FakeEventCounter + 1
                              'GoTo 0
                        End If
                  End If
                  Range("L1").Cells(StartCell, 1).Select
                  Columns("L:L").Find(What:="-1",
                        After:=ActiveCell,
                        LookIn:=xlValues, LookAt:=xlPart,
                        SearchOrder:=xlByRows,
                        SearchDirection:=xlNext,
                        MatchCase:=False,
                        SearchFormat:=False).Activate
                  StopCell = ActiveCell.Row
                  EventRow = StartCell
                  If EventRow > 73000 Then
                  qwerty = 1
                  End If

Loop Until EventRow = FirstEvent


' Save & Close TDMS File
Range("A2").NumberFormat = "mm/dd/yyyy hh:mm:ss;@"
Worksheets(TestSpecificationName).Range("N5").Cells(1
      , 1).Value = FakeEventCounter
Filename = Replace(Replace(CStr(Range("A2")), "/", "-
      "), ":", ";")
```

```
                mybook.SaveAs Filename:=EventDataPath & "Normal
                      Recording File " & Filename,
                      FileFormat:=xlOpenXMLWorkbook
                mybook.Close savechanges:=False

                ' Open The Next Normal Recording File.
            Next FNum
      End If


      ' Restore the application properties.
      With Application
            .ScreenUpdating = True
            .EnableEvents = True
            .Calculation = xlAutomatic
            .CalculateBeforeSave = True
      End With
      MsgBox "Events Have Been Extracted."

End Sub
```

Strain Compilation VBA Program

```
Sub Compile_Maximum_Event_Strains()
      Dim MyPath As String, FilesInPath As String
      Dim MyFiles() As String
      Dim FNum As Long
      Dim mybook As Workbook, BaseWks As Worksheet
      Dim CalcMode As Long
      Dim MyDate As String

'*****************************************************************
'*****************************************************************

      MyPath = Worksheets("Main").Range("D16").Cells(1, 1).Value
      PointsPerSecond = Worksheets("Main").Range("D23").Cells(1,
            1).Value
      BeforeTime = Worksheets("Main").Range("D24").Cells(1, 1).Value

'*****************************************************************
'*****************************************************************
      BeforeTime = BeforeTime * PointsPerSecond

      ' Add a slash after MyPath if needed.
      If Right(MyPath, 1) <> "\" Then
            MyPath = MyPath & "\"
      End If

      ' If there are no xlsx files in the folder, stop program.
      FilesInPath = Dir(MyPath & "*.xlsx*")
      If FilesInPath = "" Then
            MsgBox "No xlsx files found"
            Exit Sub
      End If

      ' Fill the myFiles array with the list of Excel files in the
            folder.
      FNum = 0
      Do While FilesInPath <> ""
            FNum = FNum + 1
            ReDim Preserve MyFiles(1 To FNum)
            MyFiles(FNum) = FilesInPath
            FilesInPath = Dir()
      Loop

      xsx = LBound(MyFiles)
      yxy = UBound(MyFiles)

      ' Change application properties.
      With Application
            CalcMode = .Calculation
            .Calculation = xlCalculationManual
            .ScreenUpdating = False
            .EnableEvents = False
      End With
```

```
' Add a new workbook with one sheet; set up headers.
Set BaseWks = Workbooks.Add(xlWBATWorksheet).Worksheets(1)
Worksheets("Sheet1").Range("A1").Cells(1, 1).Value = "Time of
      Trigger (i.e. when the first sensor hits 2 με)"
Worksheets("Sheet1").Range("A1").Cells(1, 2).Value = "B1978"
Worksheets("Sheet1").Range("A1").Cells(1, 3).Value = "B1979"
Worksheets("Sheet1").Range("A1").Cells(1, 4).Value = "B1980"
Worksheets("Sheet1").Range("A1").Cells(1, 5).Value = "B1984"
Worksheets("Sheet1").Range("A1").Cells(1, 6).Value = "B1985"
Worksheets("Sheet1").Range("A1").Cells(1, 7).Value = "B1986"
Worksheets("Sheet1").Range("A1").Cells(1, 8).Value = "B1987"
Worksheets("Sheet1").Range("A1").Cells(1, 9).Value = "B1988"
Worksheets("Sheet1").Range("A1").Cells(1, 11).Value = "Drift"
'Worksheets("Sheet1").Range("A1").Cells(1, 13).Value = 0
'Worksheets("Sheet1").Range("A1").Cells(1, 14).Value = 2
'Worksheets("Sheet1").Range("A1").Cells(1, 15).Value = 4
'Worksheets("Sheet1").Range("A1").Cells(1, 16).Value = 6
'Worksheets("Sheet1").Range("A1").Cells(1, 17).Value = 8
'Worksheets("Sheet1").Range("A1").Cells(1, 18).Value = 10
'Worksheets("Sheet1").Range("A1").Cells(1, 19).Value = 12
'Worksheets("Sheet1").Range("A1").Cells(1, 20).Value = 14
'Worksheets("Sheet1").Range("A1").Cells(1, 21).Value = 16
'Worksheets("Sheet1").Range("A1").Cells(1, 22).Value = 18
'Worksheets("Sheet1").Range("A1").Cells(1, 23).Value = 20
'Worksheets("Sheet1").Range("A1").Cells(1, 24).Value = "Infinity"

Worksheets("Sheet1").Range(Cells(1, 1), Cells(1, 24)).Font.Bold =
      True
BaseWks.SaveAs fileName:="C:\Users\jake\Desktop\Master File",
      FileFormat:=xlOpenXMLWorkbook

' Loop through all files in the myFiles array.
If FNum > 0 Then
      For FNum = LBound(MyFiles) To UBound(MyFiles)
            Set mybook = Nothing
            On Error Resume Next
            Set mybook = Workbooks.Open(MyPath & MyFiles(FNum))
            On Error GoTo 0

            'Select Correct Sheet & Copy Data Out
            MyDate = Range("A1").Cells(2 + BeforeTime, 1)
                  Value1 = WorksheetFunction.Max(Range(Cells(2 +
                        BeforeTime, 2), Cells(450, 2))) -
                        Range("K1")
                  Value2 = WorksheetFunction.Max(Range(Cells(2 +
                        BeforeTime, 3), Cells(450, 3))) -
                        Range("L1")
                  Value3 = WorksheetFunction.Max(Range(Cells(2 +
                        BeforeTime, 4), Cells(450, 4))) -
                        Range("M1")
                  Value4 = WorksheetFunction.Max(Range(Cells(2 +
                        BeforeTime, 5), Cells(450, 5))) -
                        Range("N1")
```

```
                  Value5 = WorksheetFunction.Max(Range(Cells(2 +
                        BeforeTime, 6), Cells(450, 6))) -
                        Range("O1")
                  Value6 = WorksheetFunction.Max(Range(Cells(2 +
                        BeforeTime, 7), Cells(450, 7))) -
                        Range("P1")
                  Value7 = WorksheetFunction.Max(Range(Cells(2 +
                        BeforeTime, 8), Cells(450, 8))) -
                        Range("Q1")
                  Value8 = WorksheetFunction.Max(Range(Cells(2 +
                        BeforeTime, 9), Cells(450, 9))) -
                        Range("R1")

            'Paste Data into Master Sheet
            Windows("Master File").Activate
            Worksheets("Sheet1").Range("A2").Cells(FNum, 1).Value
                  = MyDate
            Worksheets("Sheet1").Range("A2").Cells(FNum, 2).Value
                  = Value1
            Worksheets("Sheet1").Range("A2").Cells(FNum, 3).Value
                  = Value2
            Worksheets("Sheet1").Range("A2").Cells(FNum, 4).Value
                  = Value3
            Worksheets("Sheet1").Range("A2").Cells(FNum, 5).Value
                  = Value4
            Worksheets("Sheet1").Range("A2").Cells(FNum, 6).Value
                  = Value5
            Worksheets("Sheet1").Range("A2").Cells(FNum, 7).Value
                  = Value6
            Worksheets("Sheet1").Range("A2").Cells(FNum, 8).Value
                  = Value7
            Worksheets("Sheet1").Range("A2").Cells(FNum, 9).Value
                  = Value8

            ' Close TDMS File
            mybook.Close savechanges:=False

            ' Open the next workbook.
      Next FNum
End If


'Drift Column
Worksheets("Sheet1").Range("K2").Cells(1, 1).Value =
      "=IF(AND(B2<2,C2<2,D2<2,E2<2,F2<2,G2<2,H2<2,I2<2),1,0)"
Range("K2").Select
Selection.AutoFill Destination:=Range(Cells(2, 11), Cells(FNum,
      11)), Type:=xlFillDefault

'Create tally table
'Worksheets("Sheet1").Range("N2").Cells(1, 1).Value =
      "=IF(AND($B2>=M$1,$B2<N$1,$K2<>1),1,0)"
'Range("N2").Select
```

```
'Selection.AutoFill Destination:=Range("N2:X2"),
      Type:=xlFillDefault
'Range("N2:X2").Select
'Selection.AutoFill Destination:=Range(Cells(2, 14), Cells(FNum,
      24)), Type:=xlFillDefault
'Worksheets("Sheet1").Range("A1").Cells(1, 26).Value = "0-2"
'Worksheets("Sheet1").Range("A1").Cells(1, 27).Value = "2-4"
'Worksheets("Sheet1").Range("A1").Cells(1, 28).Value = "4-6"
'Worksheets("Sheet1").Range("A1").Cells(1, 29).Value = "6-8"
'Worksheets("Sheet1").Range("A1").Cells(1, 30).Value = "8-10"
'Worksheets("Sheet1").Range("A1").Cells(1, 31).Value = "10-12"
'Worksheets("Sheet1").Range("A1").Cells(1, 32).Value = "12-14"
'Worksheets("Sheet1").Range("A1").Cells(1, 33).Value = "14-16"
'Worksheets("Sheet1").Range("A1").Cells(1, 34).Value = "16-18"
'Worksheets("Sheet1").Range("A1").Cells(1, 35).Value = "18-20"
'Worksheets("Sheet1").Range("A1").Cells(1, 36).Value = "20+"
'Worksheets("Sheet1").Range("A1").Cells(1, 37).Value = "Sum:"
Worksheets("Sheet1").Range("A1").Cells(1, 39).Value = "# of
      Drifts"
'Worksheets("Sheet1").Range("A1").Cells(2, 26).Value =
      "=SUM(N:N)-N1"
'Worksheets("Sheet1").Range("A1").Cells(2, 27).Value =
      "=SUM(O:O)-O1"
'Worksheets("Sheet1").Range("A1").Cells(2, 28).Value =
      "=SUM(P:P)-P1"
'Worksheets("Sheet1").Range("A1").Cells(2, 29).Value =
      "=SUM(Q:Q)-Q1"
'Worksheets("Sheet1").Range("A1").Cells(2, 30).Value =
      "=SUM(R:R)-R1"
'Worksheets("Sheet1").Range("A1").Cells(2, 31).Value =
      "=SUM(S:S)-S1"
'Worksheets("Sheet1").Range("A1").Cells(2, 32).Value =
      "=SUM(T:T)-T1"
'Worksheets("Sheet1").Range("A1").Cells(2, 33).Value =
      "=SUM(U:U)-U1"
'Worksheets("Sheet1").Range("A1").Cells(2, 34).Value =
      "=SUM(V:V)-V1"
'Worksheets("Sheet1").Range("A1").Cells(2, 35).Value =
      "=SUM(W:W)-W1"
'Worksheets("Sheet1").Range("A1").Cells(2, 36).Value =
      "=SUM(X:X)"
'Worksheets("Sheet1").Range("A1").Cells(2, 37).Value =
      "=SUM(Z2:AJ2)"
Worksheets("Sheet1").Range("A1").Cells(2, 39).Value = "=SUM(K:K)"
Worksheets("Sheet1").Range(Cells(1, 26), Cells(1, 39)).Font.Bold
      = True
Worksheets("Sheet1").Range("Y5").Cells(1, 1).Value = "'<—Values
      set up for Sensor B1978"

' Restore the application properties.
With Application
      .ScreenUpdating = True
      .EnableEvents = True
      .Calculation = xlAutomatic
```

```
        End With

        'Freeze Header Row/Timestamp Column
        Range("B2").Select
        ActiveWindow.FreezePanes = True

        'Clean up time data
        'Columns("A:A").Select
        'Selection.Replace What:="-", Replacement:="/", LookAt _
              :=xlPart, SearchOrder:=xlByRows, MatchCase:=False,
                    SearchFormat:=False, _
              ReplaceFormat:=False
        'Selection.Replace What:=".", Replacement:=":", LookAt _
              :=xlPart, SearchOrder:=xlByRows, MatchCase:=False,
                    SearchFormat:=False, _
              ReplaceFormat:=False
        'Range("A1").Select

        'Set the column width in the new workbook.
        Windows("Master File").Activate
        ActiveWorkbook.Save
        BaseWks.Columns.AutoFit
        Columns("A:A").ColumnWidth = 25
        Columns("A:A").HorizontalAlignment = xlLeft
        Columns("Y:Y").ColumnWidth = 8.43
        MsgBox "Data Extraction Complete."
        Worksheets("Sheet1").Range(Cells(2, 1), Cells(FNum,
              1)).NumberFormat = "m/d/yyyy h:mm:ss AM/PM"
End Sub
```

APPENDIX B

The following specifications are taken from DJI's website (Dà-Jiāng Innovations 2018):

Aircraft

Weight (Battery & Propellers Included)	1388 g
Diagonal Size (Propellers Excluded)	350 mm
Max Ascent Speed	S-mode: 6 m/s P-mode: 5 m/s
Max Descent Speed	S-mode: 4 m/s P-mode: 3 m/s
Max Speed	S-mode: 45 mph (72 kph) A-mode: 36 mph (58 kph) P-mode: 31 mph (50 kph)
Max Tilt Angle	S-mode: 42° A-mode: 35° P-mode: 25°
Max Angular Speed	S-mode: 250°/s A-mode: 150°/s
Max Service Ceiling Above Sea Level	19685 feet (6000 m)
Max Wind Speed Resistance	10 m/s
Max Flight Time	Approx. 30 minutes
Operating Temperature Range	32° to 104°F (0° to 40°C)
Satellite Positioning Systems	GPS/GLONASS
Hover Accuracy Range	Vertical: ±0.1 m (with Vision Positioning) ±0.5 m (with GPS Positioning) Horizontal: ±0.3 m (with Vision Positioning) ±1.5 m (with GPS Positioning)

Vision System

Vision System	Forward Vision System Backward Vision System Downward Vision System
Velocity Range	≤31 mph (50 kph) at 6.6 feet (2 m) above ground
Altitude Range	0 - 33 feet (0 - 10 m)
Operating Range	0 - 33 feet (0 - 10 m)

Obstacle Sensory Range	2 - 98 feet (0.7 - 30 m)
FOV	Forward: 60°(Horizontal), ±27°(Vertical) Backward: 60°(Horizontal), ±27°(Vertical) Downward: 70°(Front and Rear), 50°(Left and Right)
Measuring Frequency	Forward: 10 Hz Backward: 10 Hz Downward: 20 Hz
Operating Environment	Surface with clear pattern and adequate lighting (lux>15)

Camera

Sensor	1’’ CMOS Effective pixels: 20M
Lens	FOV 84° 8.8 mm/24 mm (35 mm format equivalent) f/2.8 - f/11 auto focus at 1 m - ∞
ISO Range	Video: 100 - 3200 (Auto) 100 - 6400 (Manual) Photo: 100 - 3200 (Auto) 100- 12800 (Manual)
Mechanical Shutter Speed	8 - 1/2000 s
Electronic Shutter Speed	8 - 1/8000 s
Image Size	3:2 Aspect Ratio: 5472 × 3648 4:3 Aspect Ratio: 4864 × 3648 16:9 Aspect Ratio: 5472 × 3078
PIV Image Size	4096×2160(4096×2160 24/25/30/48/50p) 3840×2160(3840×2160 24/25/30/48/50/60p) 2720×1530(2720×1530 24/25/30/48/50/60p) 1920×1080(1920×1080 24/25/30/48/50/60/120p) 1280×720(1280×720 24/25/30/48/50/60/120p)
Still Photography Modes	Single Shot Burst Shooting: 3/5/7/10/14 frames Auto Exposure Bracketing (AEB): 3/5 bracketed frames at 0.7 EV Bias Interval: 2/3/5/7/10/15/20/30/60 s
Video Recording Modes	H.265 C4K:4096×2160 24/25/30p @100Mbps 4K:3840×2160 24/25/30p @100Mbps

	2.7K:2720×1530 24/25/30p @65Mbps 2.7K:2720×1530 48/50/60p @80Mbps FHD:1920×1080 24/25/30p @50Mbps FHD:1920×1080 48/50/60p @65Mbps FHD:1920×1080 120p @100Mbps HD:1280×720 24/25/30p @25Mbps HD:1280×720 48/50/60p @35Mbps HD:1280×720 120p @60Mbps H.264 C4K:4096×2160 24/25/30/48/50/60p @100Mbps 4K:3840×2160 24/25/30/48/50/60p @100Mbps 2.7K:2720×1530 24/25/30p @80Mbps 2.7K:2720×1530 48/50/60p @100Mbps FHD:1920×1080 24/25/30p @60Mbps FHD:1920×1080 48/50/60 @80Mbps FHD:1920×1080 120p @100Mbps HD:1280×720 24/25/30p @30Mbps HD:1280×720 48/50/60p @45Mbps HD:1280×720 120p @80Mbps
Max Video Bitrate	100 Mbps
Supported File Systems	FAT32 (≤32 GB); exFAT (>32 GB)
Photo	JPEG, DNG (RAW), JPEG + DNG
Video	MP4/MOV (AVC/H.264; HEVC/H.265)
Supported SD Cards	Micro SD Max Capacity: 128GB Write speed ≥15MB/s, Class 10 or UHS-1 rating required
Operating Temperature Range	32° to 104°F (0° to 40°C)

Charger

Voltage	17.4 V
Rated Power	100 W

App / Live View

Mobile App	DJI GO 4
Live View Working Frequency	2.4 GHz ISM, 5.8 GHz ISM
Live View Quality	720P @ 30fps

Latency	Phantom 4 Pro: 220 ms (depending on conditions and mobile device) Phantom 4 Pro $^{+}$: 160 - 180 ms
Required Operating Systems	iOS 9.0 or later Android 4.4.0 or later

Gimbal

Stabilization	3-axis (pitch, roll, yaw)
Controllable Range	Pitch: -90° to +30°
Max Controllable Angular Speed	Pitch: 90°/s
Angular Vibration Range	±0.02°

Infrared Sensing System

Obstacle Sensory Range	0.6 - 23 feet (0.2 - 7 m)
FOV	70° (Horizontal), ±10° (Vertical)
Measuring Frequency	10 Hz
Operating Environment	Surface with diffuse reflection material, and reflectivity > 8% (such as wall, trees, humans, etc.)

Remote Controller

Operating Frequency	2.400 - 2.483 GHz and 5.725 - 5.825 GHz
Max Transmission Distance	2.400 - 2.483 GHz (Unobstructed, free of interference) FCC: 4.3 mi (7 km) CE: 2.2 mi (3.5 km) SRRC: 2.5 mi (4 km) 5.725 - 5.825 GHz (Unobstructed, free of interference) FCC: 4.3 mi (7 km) CE: 1.2 mi (2 km) SRRC: 3.1 mi (5 km)
Operating Temperature Range	32° to 104°F (0° to 40°C)
Battery	6000 mAh LiPo 2S
Transmitter Power (EIRP)	2.400 - 2.483 GHz FCC: 26 dBm CE: 17 dBm SRRC: 20 dBm MIC: 17 dBm 5.725 - 5.825 GHz FCC: 28 dBm

	CE: 14 dBm SRRC: 20 dBm MIC: -
Operating Current/Voltage	1.2 A@7.4 V
Video Output Port	GL300E: HDMI GL300F: USB
Mobile Device Holder	GL300E: Built-in display device (5.5 inch screen, 1920×1080, 1000 cd/m^2, Android system, 4 GB RAM ＋16 GB ROM) GL300F: Tablets and smart phones

Intelligent Flight Battery

Capacity	5870 mAh
Voltage	15.2 V
Battery Type	LiPo 4S
Energy	89.2 Wh
Net Weight	468 g
Charging Temperature Range	41° to 104°F (5° to 40°C)
Max Charging Power	160 W