5-31-2018

Implementation of Unmanned aerial vehicles (UAVs) for assessment of transportation infrastructure - Phase II

Colin Brooks
Michigan Technological University, cnbrooks@mtu.edu

Richard Dobson
Michigan Technological University, rjdobson@mtu.edu

David Banach
Michigan Technological University, dmbanach@mtu.edu

Thomas Oommen
Michigan Technological University, toommen@mtu.edu

Kuilin Zhang
Michigan Technological University, klzhang@mtu.edu

See next page for additional authors

Recommended Citation

Brooks, C., Dobson, R., Banach, D., Oommen, T., Zhang, K., Mukherjee, A., Havens, T. C., Ahlborn, T., Escobar-Wolf, R., Bhat, C., Zhao, S., Lyu, Q., & Marion, N. (2018). Implementation of Unmanned aerial vehicles (UAVs) for assessment of transportation infrastructure - Phase II.
Retrieved from: https://digitalcommons.mtu.edu/michigantech-p/4

Follow this and additional works at: https://digitalcommons.mtu.edu/michigantech-p
Part of the Engineering Commons

Authors

Colin Brooks, Richard Dobson, David Banach, Thomas Oommen, Kuilin Zhang, Amlan Mukherjee, Timothy C. Havens, Theresa Ahlborn, Rudiger Escobar-Wolf, Chaitanya Ganesh Bhat, S. Zhao, Q. Lyu, and Nicholas Marion

Michigan Tech Research Institute

MDOT Michigan Department of Transportation

www.mtri.org

Implementation of Unmanned Aerial Vehicles (UAVs) for Assessment of Transportation Infrastructure – Phase II

Final Report, No. SPR-1674

Prepared by:

C. Brooks[1], R. Dobson[1], D. Banach[1], T. Oommen[2], K. Zhang[2], A. Mukherjee[2], T. Havens[2], T. Ahborn[2], R. Escobar-Wolf[2], C. Bhat[2], S. Zhao[2], Q. Lyu[2], and N.Marion[1]

[1]Michigan Tech Research Institute (MTRI), Michigan Technological University, 3600 Green Ct., Ste. 100, Ann Arbor, Michigan 48105

[2] Michigan Technological University (MTU), 1400 Townsend Drive, Houghton, Michigan, 49931

Project Manager:

Steven J. Cook, P.E.
Michigan Department of Transportation
6333 Lansing Road
Lansing, MI 48917

Research Manager:

André Clover, P.E.
Michigan Department of Transportation
8885 Ricks Road
Lansing, MI 48917

Revised version - May 9, 2018

1. Report No. SPR-1674	**2. Government Accession No.** N/A	**3. MDOT Project Manager** Steven J. Cook, P.E

4. Title and Subtitle Implementation of Unmanned Aerial Vehicles (UAVs) for Assessment of Transportation Infrastructure - Phase II	**5. Report Date** May 31, 2018
	6. Performing Organization Code N/A
7. Author(s) C.Brooks, R. Dobson, D. Banach, T. Oommen, K. Zhang, A. Mukherjee, T. Havens, T. Ahlborn, R. Escobar-Wolf, C. Bhat, S. Zhao, Q. Lyu, and N.Marion	**8. Performing Org. Report No.** N/A
9. Performing Organization Name and Address Michigan Technological University 1400 Townsend Drive Houghton, MI 49931	**10. Work Unit No.** N/A
	11. Contract or Grant No. 2016-0067-Z1
12. Sponsoring Agency Name and Address Michigan Department of Transportation Research Administration 8885 Ricks Road P.O. Box 33049 Lansing, Michigan 48909	**13. Type of Report & Period Covered** Final Report, May 1, 2016 – May 31, 2018
	14. Sponsoring Agency Code N/A

15. Supplementary Notes

16. Abstract

Technological advances in unmanned aerial vehicle (UAV) technologies continue to enable these tools to become easier to use, more economical, and applicable for transportation-related operations, maintenance, and asset management while also increasing safety and decreasing cost. This Phase 2 project continued to test and evaluate five main UAV platforms with a combination of optical, thermal, and lidar sensors to determine how to implement them into MDOT workflows. Field demonstrations were completed at bridges, a construction site, road corridors, and along highways with data being processed and analyzed using customized algorithms and tools. Additionally, a cost-benefit analysis was conducted, comparing manual and UAV-based inspection methods. The project team also gave a series of technical demonstrations and conference presentations, enabling outreach to interested audiences who gained understanding of the potential implementation of this technology and the advanced research that MDOT is moving to implementation. The outreach efforts and research activities performed under this contract demonstrated how implementing UAV technologies into MDOT workflows can provide many benefits to MDOT and the motoring public; such as advantages in improved cost-effectiveness, operational management, and timely maintenance of Michigan's transportation infrastructure

17. Key Words Unmanned aerial vehicle, UAV, transportation, operations, maintenance, asset management, bridges, traffic, GIS, remote sensing	**18. Distribution Statement** No restrictions. This document is available to the public through the Michigan Department of Transportation.

19. Security Classification - report Unclassified	**20. Security Classification - page** Unclassified	**21. No. of Pages** 164	**22. Price** N/A

Acknowledgments

The project team would like to thank and acknowledge the MDOT program manager, Steve Cook, P.E. and the research manager, André Clover, P.E. for their advice and oversight during this project. The PI would like to thank his project team for their dedication to collecting quality, useful data safely as well as the help of MDOT staff in accessing and collecting data at field sites (Steve Katenhus and LeighAnn Mikesell), which was critical to the project as their help was greatly appreciated.

Research Report Disclaimer

This publication is disseminated in the interest of information exchange. The Michigan Department of Transportation (hereinafter referred to as MDOT) expressly disclaims any liability, of any kind, or for any reason, that might otherwise arise out of any use of this publication or the information or data provided in the publication. MDOT further disclaims any responsibility for typographical errors or accuracy of the information provided or contained within this information. MDOT makes no warranties or representations whatsoever regarding the quality, content, completeness, suitability, adequacy, sequence, accuracy or timeliness of the information and data provided, or that the contents represent standards, specifications, or Regulations.

Contents

List of Figures

List of Tables

1. Executive Summary

This report describes the accomplishments of the "Implementation of Unmanned Aerial Vehicles (UAVs) for Assessment of Transportation Infrastructure - Phase II" research project, contract number 2016-0067, project number Authorization No. 1, OR number OR15-139, as funded by the US DOT Federal Highway Administration (FHWA) and Michigan Department of Transportation (MDOT). Senior Research Scientist Colin Brooks of the Michigan Tech Research Institute (MTRI), a research center of Michigan Technological University ("MTU"), served as the Principal Investigator on the project, with a period of performance from May 1, 2016 to May 31, 2018. Co-Investigators were Dr. Thomas Oommen of the Michigan Tech Department of Geological/Mining Engineering and Sciences, Dr. Tim Havens of the Michigan Tech Electrical and Computer Engineering Department, Dr. Amlan Mukherjee, Dr. Kuilin Zhang, Dr. Theresa (Tess) Ahlborn all from the Michigan Tech Department of Civil and Environmental Engineering, and Research Scientist Richard Dobson of MTRI. A research project team comprised of Michigan Tech researchers and students provided critical support to performing and documenting all research tasks performed under this contract. The project builds directly from MDOT's Phase I project, called "Evaluating the Use of Unmanned Aerial Vehicles for Transportation Projects", as captured in the Phase I final report (Brooks et al. 2015, also http://www.michigan.gov/mdot/0,4616,7-151-9622_11045_24249-353767--,00.html) and in a recent Transportation Research Board (TRB) conference paper (Brooks et al. 2017).

This report describes background information on each of the UAV, (also unmanned aerial system / UAS, and colloquially "drone") platforms, sensors, and types of infrastructure that have been included in the project. Five main UAV platforms were tested and demonstrated, including the Bergen hexacopters manufactured in Michigan (a six-bladed multi-rotor UAV), a Bergen Quad 8 octocopter manufactured in Michigan (an eight-bladed multi-rotor UAV), a small DJI Phantom 3 Advanced (3A) quadcopter, a small DJI Mavic Pro quadcopter, and the Mariner 2 Splash quadcopter (a waterproof platform). The hexacopter platform, paired with optical and thermal sensors was determined to be most appropriate for high-resolution imaging of transportation infrastructure, such as creating the automated calculation of the locations and volumes of spalls as well as locations of likely bridge deck delaminations. Progress on the use of thermal data to find and map delaminations in concrete bridge decks has also been captured in a recent publication by the project team, Escobar-Wolf et al. (2017). The Quad-8 octocopter, paired with optical (digital camera) and light detection and ranging (lidar, also known as LiDAR) sensors was most appropriate for flying both sensors at the same time for testing purposes and along a road corridor. The DJI Phantom 3A quadcopter was shown to be useful for rapid overview imaging of bridge sites and road corridors. The DJI Mavic Pro proved most useful for collecting rapid overview imaging of a scene, traffic monitoring video due to its stability and high resolution, and close-up imagery of the side of bridge due to is forward collision avoidance technology. The Mariner 2 Splash proved most useful for collecting imagery underneath the bridge. The traffic monitoring blimp from Phase 1 of the project was reintroduced into Phase 2, but due to windy conditions at the data collection site, was not able to be successfully deployed.

The Mavic Pro was used for traffic video imaging instead, providing a stable, high-resolution platform for shorter-duration data collection. All of the output datasets have been stored on a data storage site provided by project partners, Surveying Solutions Inc. (SSI), that enabled sharing data with MDOT.

Five bridges, three road corridors, one construction site, and one testing site located outside of MTRI's office building were all evaluated through UAV-enabled imaging through this phase of the project. Appropriate lab testing was also completed ahead of the field testing and demonstrations to ensure safe and efficient operations. Four bridges were sensed while traffic was completely blocked off and rerouted around the sites, which provided the ability to collect data without interruption. The fifth bridge was part of a road corridor site and was imaged from an oblique angle to demonstrate UAV data collection capabilities without having to close traffic on a busy US Highway. At some locations, undersides, substructure elements, and fascia were also imaged. From the optical camera / imagery (a Nikon D810 with 50 millimeter prime lens), orthophotos with resolutions ranging between 0.007 ft (2 mm) to 0.026 ft (8 mm) were created, along with a three-dimensional (3D) profile (a digital elevation model or DEM), and a GIS layer of spalls that were located and characterized using an updated automated spall detection algorithm. Two thermal cameras, the FLIR Vue Pro and FLIR Vue Pro R (radiometric) enabled detection of potential delaminations from a UAV platform, with an updated automated delamination detection algorithm. A small FLIR Duo sensor (with both thermal and optical sensors) was tested. A lidar sensor was used to collect data at a road site near Muskegon, Michigan, but needed further development before it could be used to effectively collect data. Additionally, the lidar sensor was further tested to collect data outside of MTRI to demonstrate successful lidar data collection and processing of point cloud of a short stretch of roadway.

One demonstration was included in the project that were not part of the original objectives. This data collection was at a construction site near I-75 and Square Lake Road in Troy, Michigan, where the Bergen hexacopter and optical sensor were flown to collect aerial imagery of a aggregate mounds. Collected imagery was compared to lidar data collected via a commercial high-end drone being tested by project partners SSI. At the end of the project, the project team will participate in the 2018 Operations and Traffic Safety Conference in Mt. Pleasant, Michigan. During the conference, the project team will provide demonstrations and overviews of the capabilities of the UAVs and sensors for management and operations, including infrastructure assessment and corridor assessment.

Through these extensive testing, field demonstration, and outreach efforts, UAV technologies have continued to show how they are useful to meet the data and decision support needs of MDOT. Improved bridge deck assessment, traffic monitoring, roadway asset detection, lidar, and thermal data processing have all been advanced towards practical usage and implementation. As part of the project, a detailed cost-benefit analysis was conducted to compare manual and automatic bridge evaluation techniques. An Implementation Action Plan (IAP) is provided as a separate, shorter report derived from this report's content, with a focus on potential next steps for full implementation of UAV technologies as part of MDOT operations. The project team looks forward to the opportunity to continue helping MDOT take advantage of the increasing capabilities of rapidly advancing UAV technology through day-to-day deployment in the agency and its partners.

2. Background

2.1 Objectives

The objectives of this research project, as described in the project Statement of Work, were to:

1. Develop, deploy, and implement near-time data collection communication backhaul and data storage capabilities proof of concept for the most viable UAV platforms and sensing capabilities.
2. Develop, deploy, and implement (via pilot projects) UAV data uses, analysis, and processing systems delivered from on-board sensors for two (2) to three (3) specific business functions/activities identified by MDOT.
3. Demonstrate, deploy, and implement (via pilot projects) data quality protocols to ensure data collected is accurate and within tolerance requirements when compared to current data collection systems at MDOT for the same two (2) to three (2) specific business functions/activities identified by MDOT.
4. Demonstrate a proof of concept for data collection uses UAVs for transportation purposes, beyond those proven during Phase 1, from various highway assets.
5. Coordinate/leverage ongoing and past research of UAV sensing and data collection technologies. Provide device training and deployment/implementation plan, including a user/operation guidance document.
6. Determine the return on investment (benefit/cost analysis) performed on UAVs and sensory technologies deployed for pilot studies performed for this research project.
7. Secure a Federal Aviation Administration (FAA) Certificate of Authorization (COA) to complete the below tasks and deliverables. Note; Michigan Tech obtained a COA under the FAA Section 333 rules before the start of the project, but started operating under the newer Part 107 rules once they came into effect in August, 2016.

2.2 Tasks

- Task 1A – Collect data from the UAV platform using sensing technology in near-time (as real-time as can be achieved) demonstrating, developing, and implementing storage capabilities of large amounts, usage, and application development of data.
- Task 1B – Provide data collection from UAVs to the MDOT Data Use, Analysis, and Process (DUAP) project that meets the quality, low latency delivery and data format requirements.

- Task 1C – Provide a report that describes and recommends optimal methods to store and distribute potentially large imaging, point could and 3D surface datasets created through UAV-based data collection.
- Task 2 – Demonstrate, develop, and implement high-accuracy simultaneous thermal/photo/video/lidar measurement using a high-fidelity sensor-fused UAV positioning approach.
- Task 3 – Demonstrate the capabilities to complete aerial remote sensing data collections to meet MDOT mapping and construction monitoring needs. Coordinate with MDOT Survey Support to identify pilot projects and meet data delivery needs satisfying MDOT requirements for spatial data collections as it pertains to data density, absolute and relative 3D positional accuracy.
- Task 4 – Demonstrate, develop, and implement uses of data collection from UAV(s) and sensors for operations, maintenance, design, and asset management.
- Task 5A – Demonstrate, develop, and implement enhanced testing of UAV-based thermal imaging for bridge deck structural integrity.
- Task 5B – Compare data collected from UAV sensors to current data collected and systems used at MDOT for highway assets/operations.
- Task 6 – Demonstrate, develop, and implement systems management and operations uses.
- Task 7 – Provide a benefit/cost analysis and performance measures that define the return on investment as a result of deploying UAVs and related sensory technologies for transportation purposes.
- Task 8 - Secure a Federal Aviation Administration Certificate of Authorization to complete the below tasks and deliverables.

3. Methodology

An important component of the project's methodology was to use representative field data collection sites, selected with input from MDOT to enable how UAVs could provide critical data and results that meet MDOT's needs and objectives. The following section provides a brief overview of each location where these data were collected during the project. The locations were located around southern Michigan, including Ann Arbor, near Ovid, near Troy, near Saginaw, and near Muskegon / Whitehall, Michigan. Data from a Phase I data collection near Livonia was also used to test the enhanced capabilities of a bridge spall detection algorithm. Some sensor testing also occurred at the main campus of Michigan Tech in Houghton.

3.1 Data Collection Locations

1. South Warren Road (near Ovid, Michigan) – STR 9852
Date of Collection – July 26, 2016

The South Warren Road Bridge is 85 feet long, 31 feet wide and crosses the Maple River (Figures 1 and 2). A 2016 MDOT inspection indicated the bridge surface contained a chip seal over a hot mix asphalt (HMA) overlay with some delaminations and a surface rating of 5. The site was used to set up and test data collection platforms and sensors, and had previously been considered as a potential site based on a MDOT recommendation back in Phase 1 of the project. At this site, imagery using the Bergen hexacopter / Nikon D810 combination, DJI Phantom 3A, and FLIR Vue Pro sensors was collected. The project team demonstrated how GIS layers could be created including an orthophoto (Figure 2), DEM, hillshade and thermal layer of the site. SSI provided survey-grade quality global positioning system (GPS) positioning to help with accurate georeferencing of the GIS output layers.

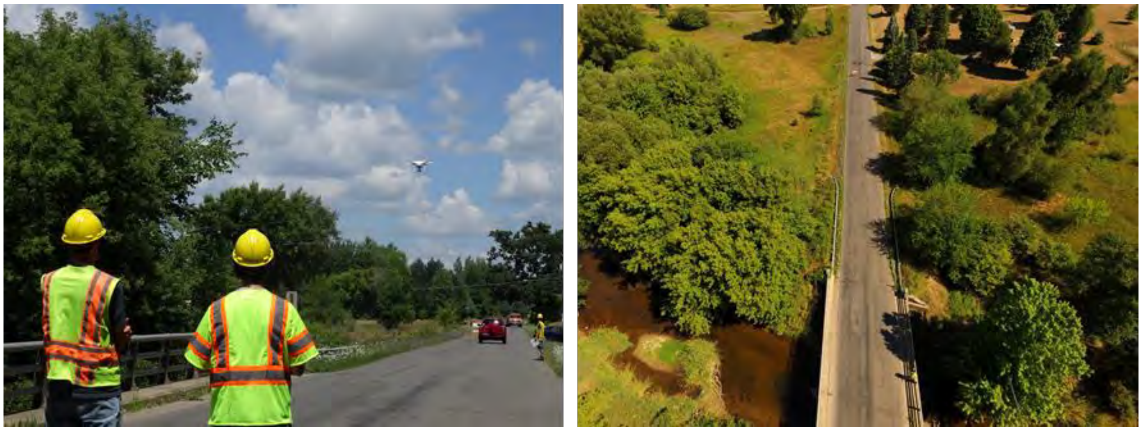


Figure 1: Collecting imagery data of the South Warren Road over Maple River bridge site (left) and example UAV image of the site (right) used to help create an overview image.

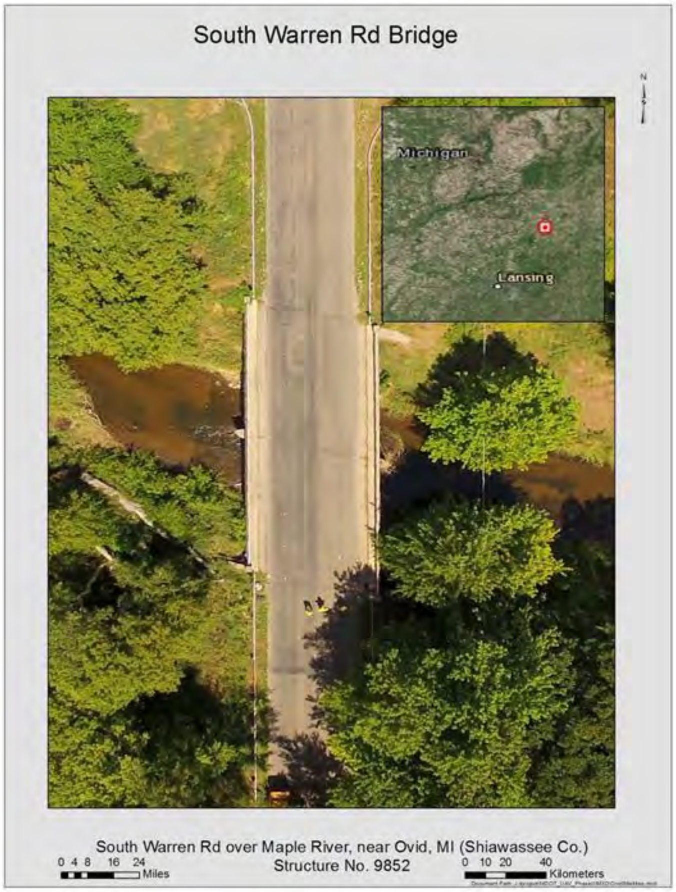


Figure 2: South Warren Rd Bridge, displayed with UAV imagery developed into an overhead orthophoto product.

2. Gordonville Road – STR 6999
Date of Collection – September 15, 2016

The Gordonville Road Bridge over Little Salt Creek measures 126 feet long and 35 feet wide (Figures 3, 4, and 5). The deck of this bridge is concrete. A 2015 MDOT bridge inspection report indicated the bridge had spalling and gave it a surface rating of 5. At this site, imagery using the Nikon D810, DJI Phantom 3A and Flir Vue Pro sensors was collected. GIS-based layers created include an orthophoto, DEM, hillshade, and thermal layer of the site.

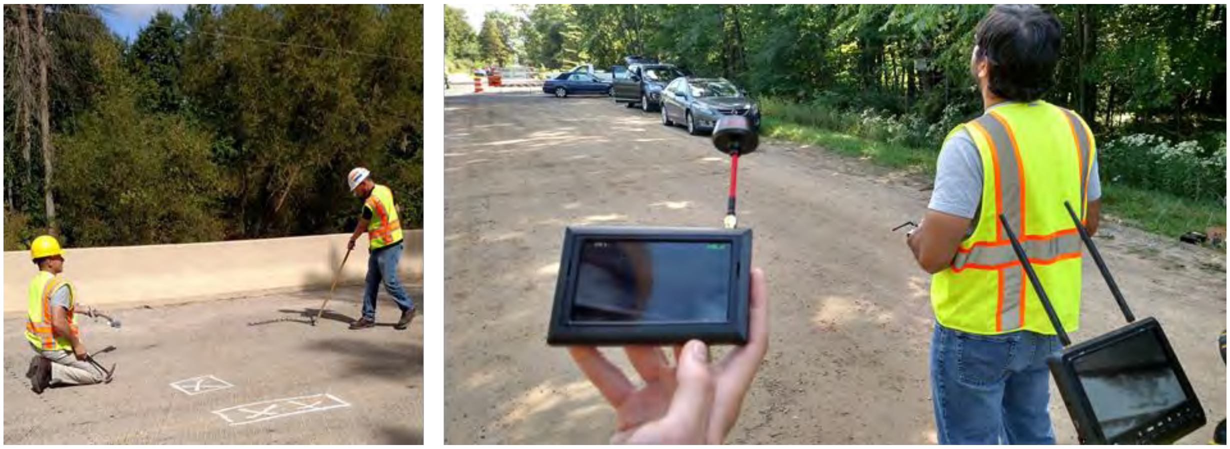


Figure 3: Gordonville Road data collection photos: MDOT staff collect ground truth delamination data with chain dragging (left); R. Dobson collects thermal data with the Bergen hexacopter and FLIR Vue Pro combination with real-time display of the thermal data visible in two first-person viewer screens (right).

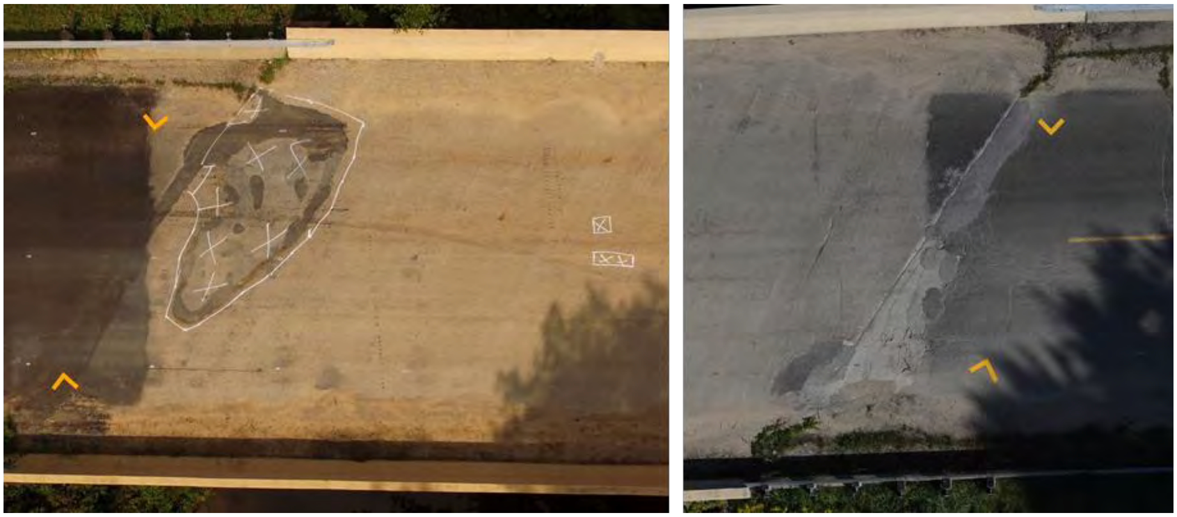


Figure 4: Example digital camera images from the Gordonville Road site: a DJI Phantom 3A image showing surface patching and sounding survey markings (left); a Nikon D810 image showing part of the bridge with distresses at the eastern bridge joint. Both photos have GPS ground control markers visible (yellow chevrons).

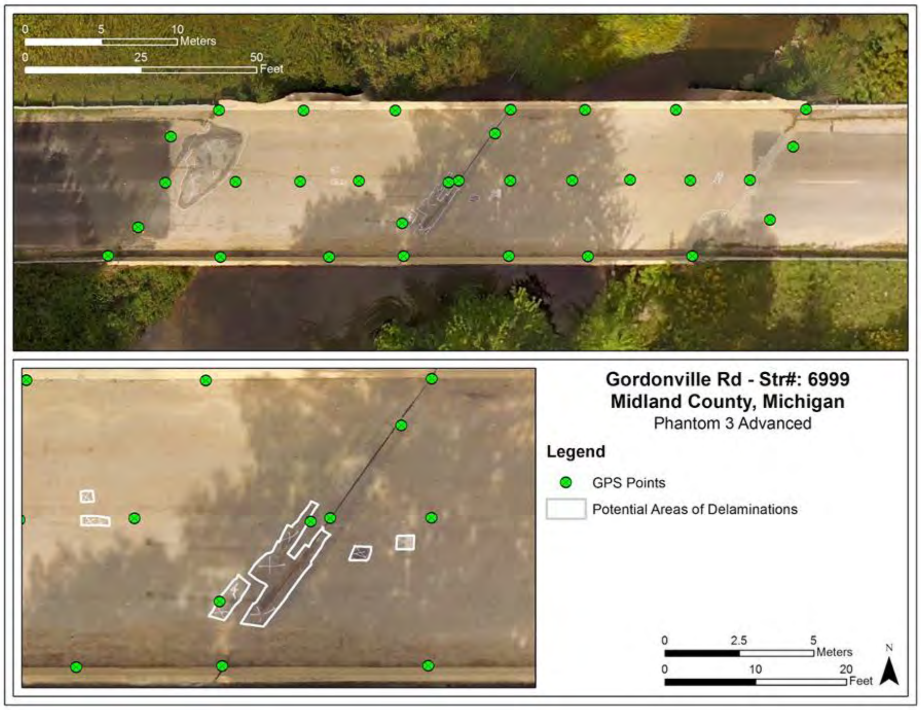


Figure 5: Gordonville Road Bridge, displayed with UAV imagery, high-resolution GPS points, and marking for manual soundings of potential delaminations by MDOT staff.

3. I-75 / Square Lake Road Construction Site

Date of Collection – October 10, 2016

As part of this project, SSI obtained temporary access to a Riegl Ri-COPTER (a high-end lidar UAV costing in the $400,000 range with platform and sensors) to use as a benchmark to compare with photo UAV data (Figure 6). The SSI Riegl RiCopter was flown at the I-75 Square Lake interchange construction site and the data from the flight was processed and used for comparison purposes. For comparison, Michigan Tech used a Bergen Hexacopter with the Nikon D810 (Figure 7) payload to create orthophotos (Figure 8), DEMs, and hillshade models of two separate aggregate mounds. The volumes of each aggregate mound using the DEM were also computed and was compared to data taken from the RiCopter lidar unit that surveyed the same sites. Additionally, the elevation difference between the RiCopter lidar and Bergen optical data was calculated, with the majority of the difference being between +/- one inch (Figure 9).

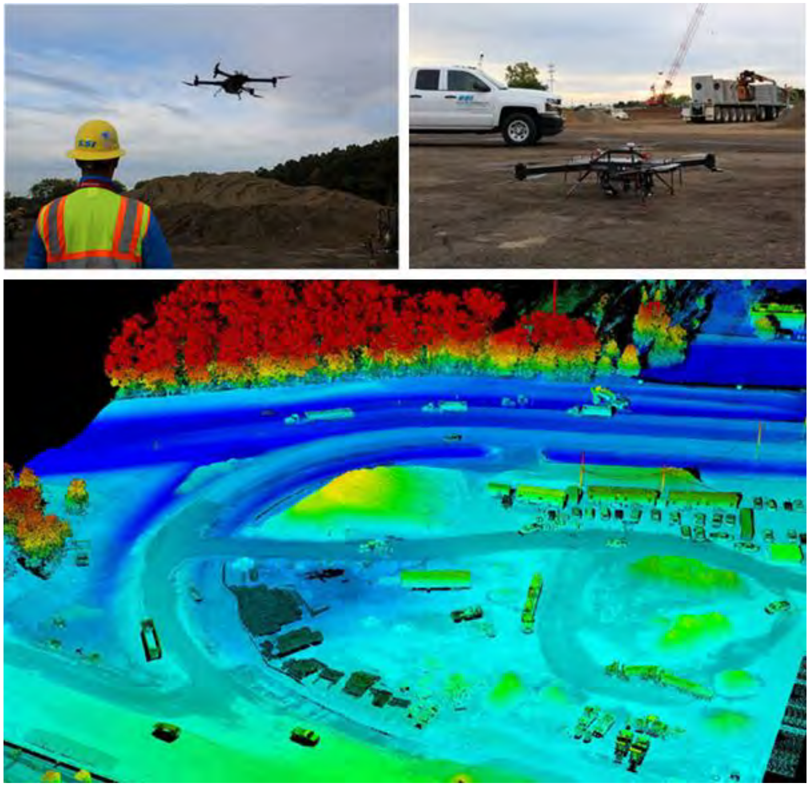


Figure 6: UAV-based Ricopter lidar platform and resulting point cloud data demonstrated at I-75 / Square Lake construction site.

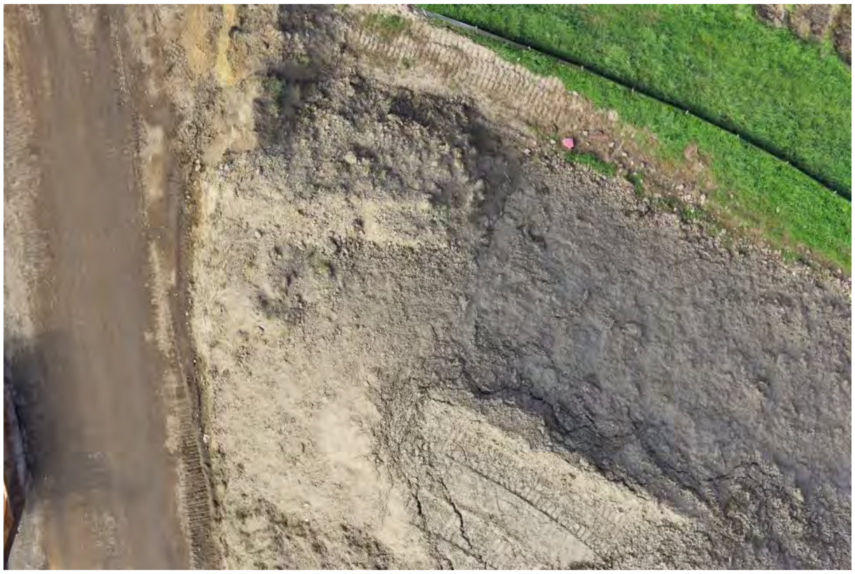
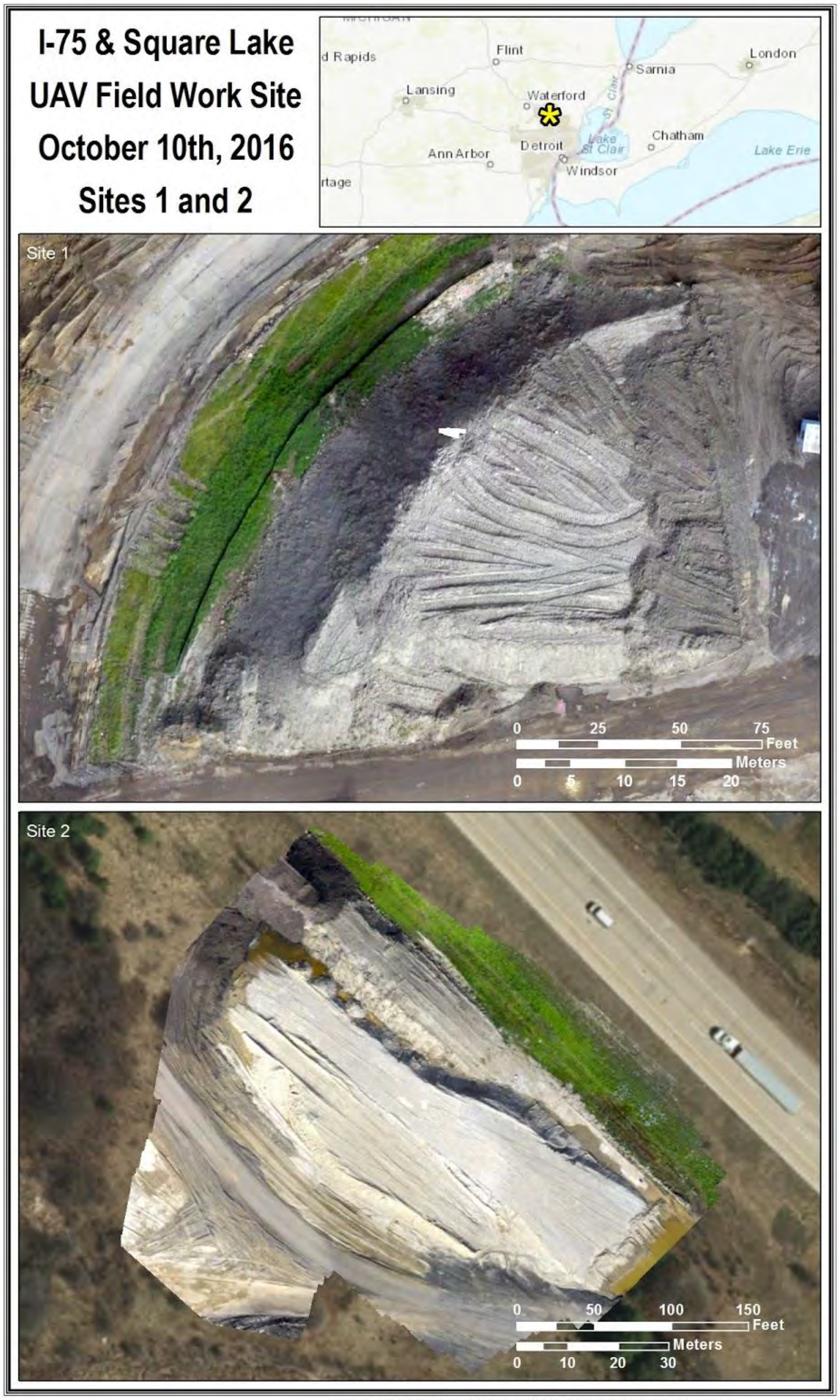


Figure 7: Nikon D810 36 MP image of one of the construction aggregate mounds as collected with the Bergen Hexacopter; it was used to create an orthophoto and DEM of the construction site area.



Document Path: J:\project\MDOT_UAV_PhaseII\MXD\I75_SquareLake_20161011_SiteComparison.mxd

Figure 8: Orthophotos of two construction aggregate mounds at the I-75 Square Lake construction site collected with Bergen Hexacopter / Nikon D810 sensing combination.

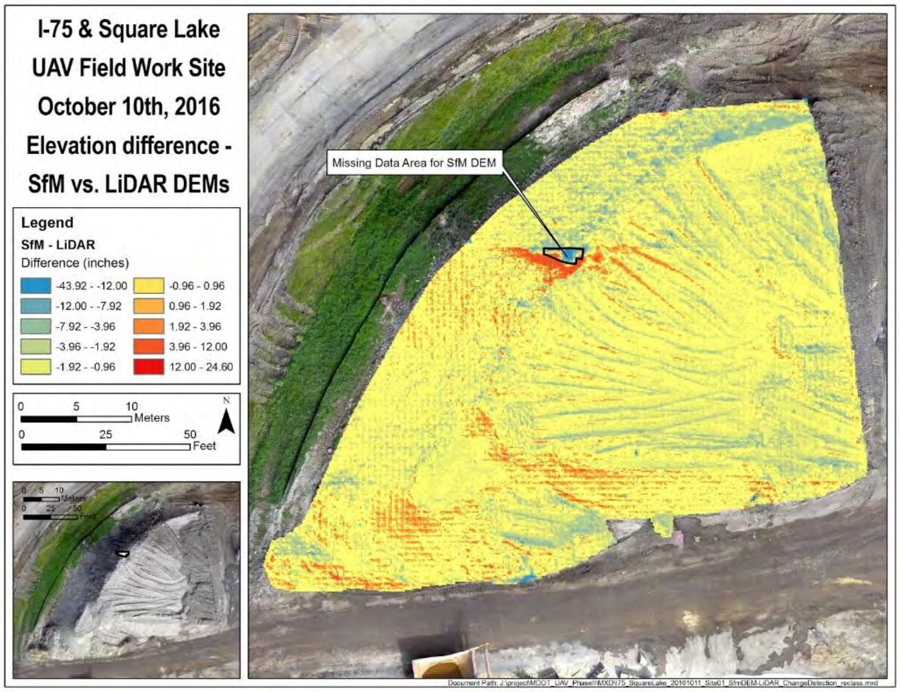


Figure 9: Elevation difference between the RiCopter lidar and Bergen optical data for a construction aggregate mound. Most values are +/- 1 inch (2.5 cm)

4. Uncle Henry Road – STR 9289

Dates of Collection – November 14, 2016 and December 30, 2016

Uncle Henry Road Bridge over the Uncle Henry Drain is 60 feet long, 26 feet wide and has a concrete deck (Figures 10, 11, and 12). A 2015 MDOT bridge inspection states that there were several spalls (12 sq. ft. in five locations) and delaminations (15 sq. ft in five locations) present on the bridge deck, which was assigned a surface rating of 5. Nikon D810 and Phantom 3A imagery were collected at this site and processed to create an orthophoto, DEM, and hillshade model of the bridge. Thermal imagery was collected from a FLIR Vue Pro flying aboard a Bergen Hexacopter.

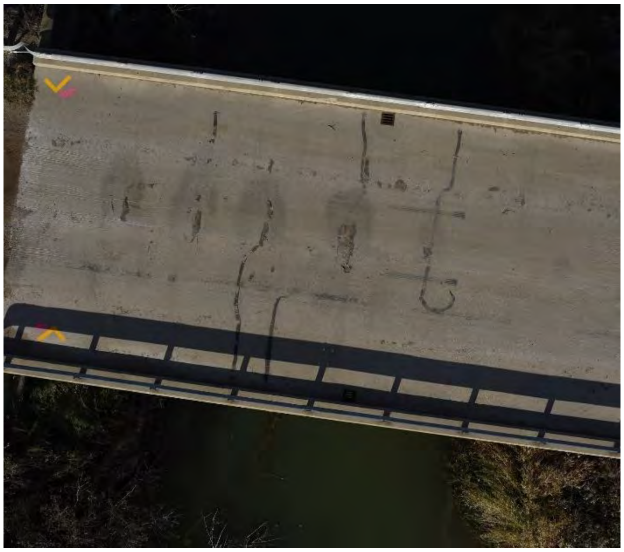


Figure 10: Nikon D810 image of the Uncle Henry Road bridge deck.

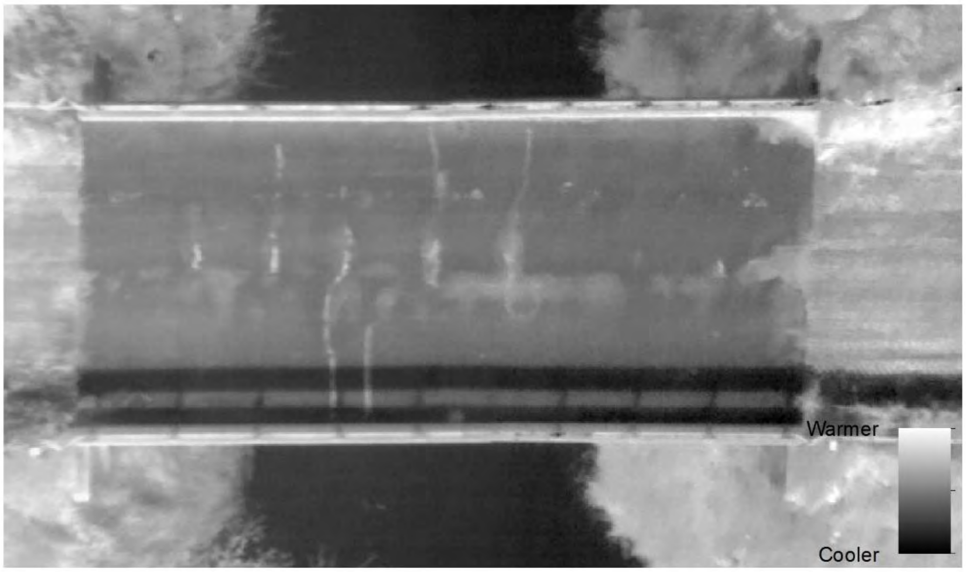


Figure 11: Thermal image of relative temperature for the Uncle Henry Road bridge deck and surrounding areas collected by the FLIR Vue Pro from the Bergen Hexacopter; lighter colors are warmer, darker colors are cooler.

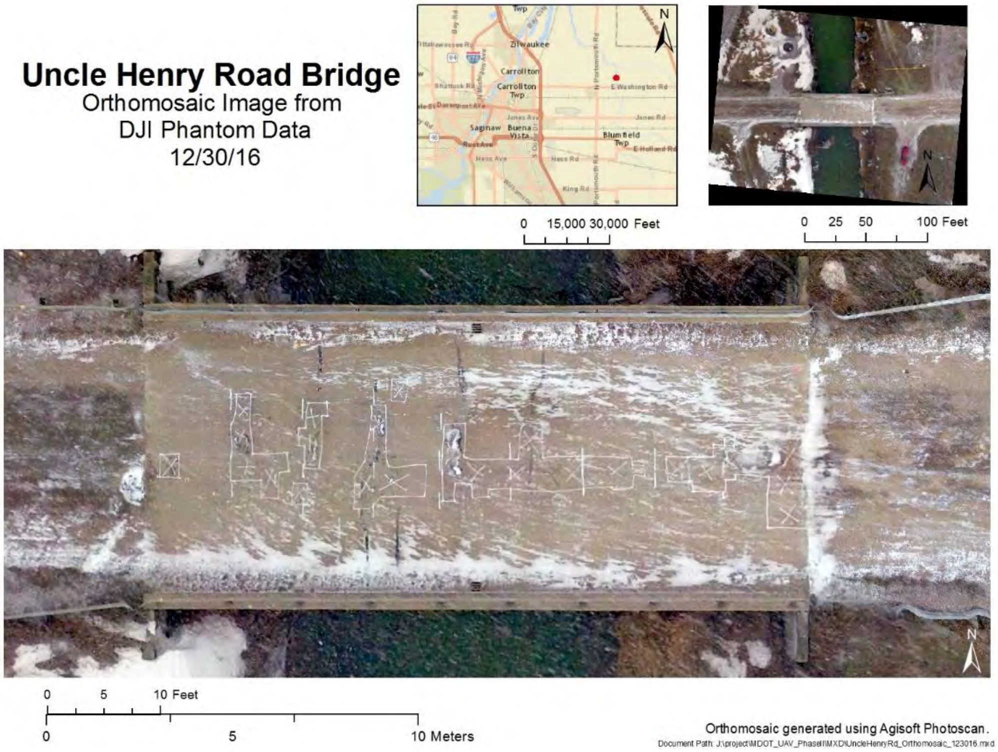


Figure 12: Uncle Henry Road Bridge, as imaged by Phantom 3A UAV platform, with delamination markings from a MDOT sounding survey used to compare to thermal analysis results.

5. Beyer Road – STR 9293

Dates of Collection – November 14, 2016 and December 30, 2016

The bridge on Beyer Road over Cheboyganing Creek is 67 feet long, 26 feet wide and has a concrete surface (Figures 13, 14, 15, and 16). A 2016 MDOT bridge report indicated that there were four spalled and one delaminated region on the bridge deck at the time, which was assigned a surface rating of 5 (Fair condition). At this site, imagery using the Nikon D810, DJI Phantom 3A, and Flir Vue Pro sensors were collected. GIS-based layers created include an orthophoto, DEM, hillshade and thermal layer of the site and SSI provided survey-ground GPS ground control points for georeferencing outputs.

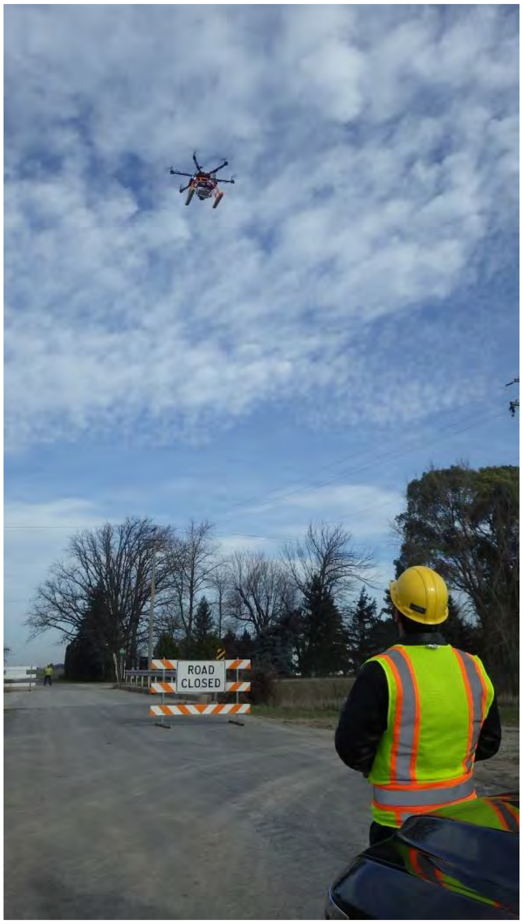


Figure 13: Collecting Nikon D810 imagery of the Beyer Road bridge site.

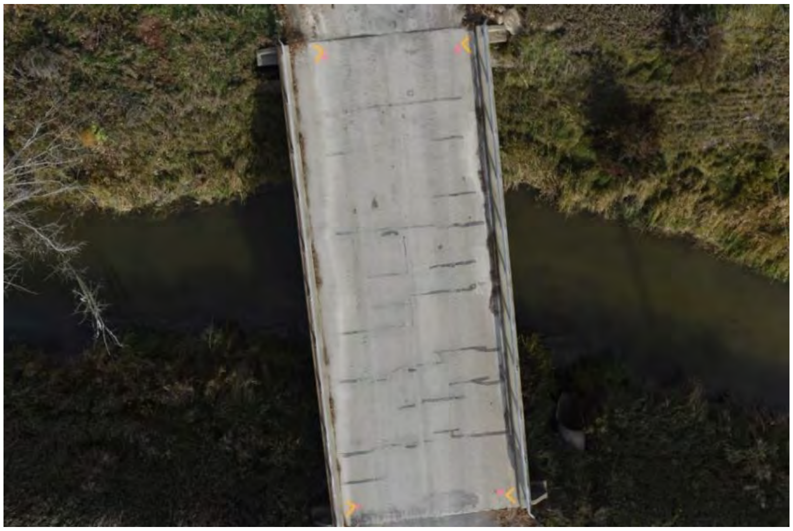


Figure 14: Nikon D810 image used for orthophoto and DEM creation for the Beyer Rd bridge.

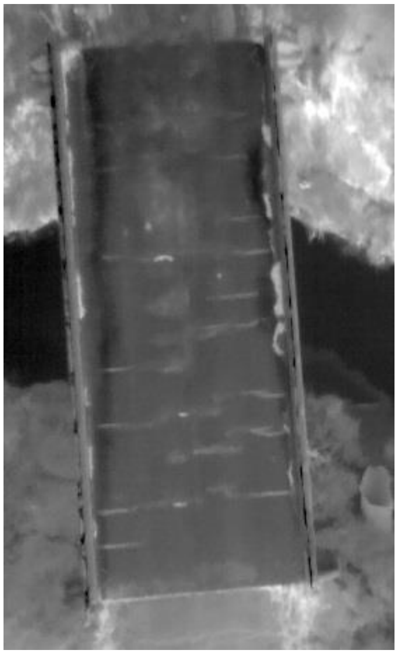


Figure 15: UAV-collected thermal image of the Beyer Rd bridge and surroundings; lighter colors are warmer, darker colors are cooler.

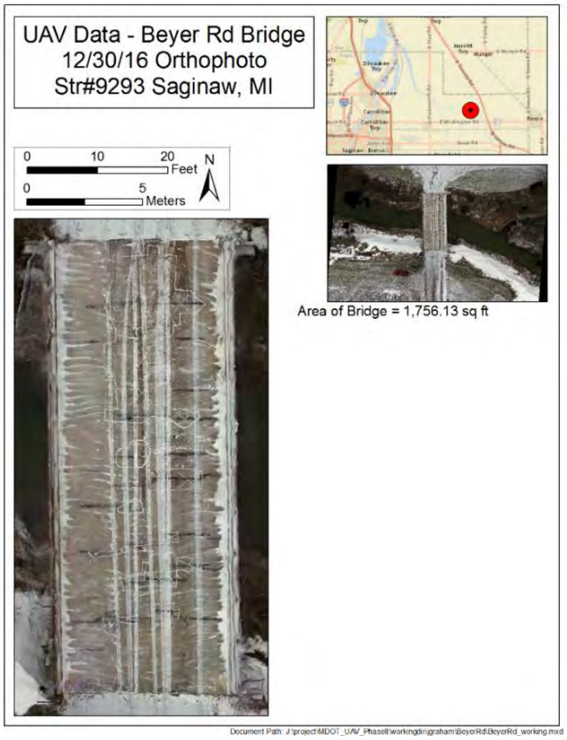

UAV Data - Beyer Rd Bridge
12/30/16 Orthophoto
Str#9293 Saginaw, MI
0
10
20
Feet
0
5
Meters
N
Merritt Twp
Reese
Carrollton
Carrollton Twp
Area of Bridge = 1,756.13 sq ft
Document Path: J:\project\MDOT_UAV_PhaseII\workingdir\graham\BeyerRd\BeyerRd_working.mxd

Figure 16: Beyer Road Bridge, with delamination markings from a MDOT sounding survey.

6. M-120 / Holton Road

Date of Collection – June 14, 2017

At the M-120/Holton Road site, data were collected along a 0.23 mile (0.37 km) road corridor (Figures 17, 18, 19, 20, and 21) used to represent the potential for inventory and assessment for longer road corridors surveys via UAV. Traffic control was provided by LeighAnn Mikesell (MDOT - Muskegon Transportation Service Center). At this site, imagery using the Nikon D810, Phantom 3A, Mavic Pro, FLIR Vue Pro, and FLIR Vue Pro R were collected. GIS-based layers created include orthophotos, DEMs, hillshade models, and traffic video at the site. Survey grade GPS data were collected by SSI. Furthermore, orthomosaic thermal imagery that displayed relative temperature values in addition to radiometrically calibrated per-pixel temperature values were also created. The site was selected in part because SSI had collected mobile-based lidar data in a previous project for this corridor.

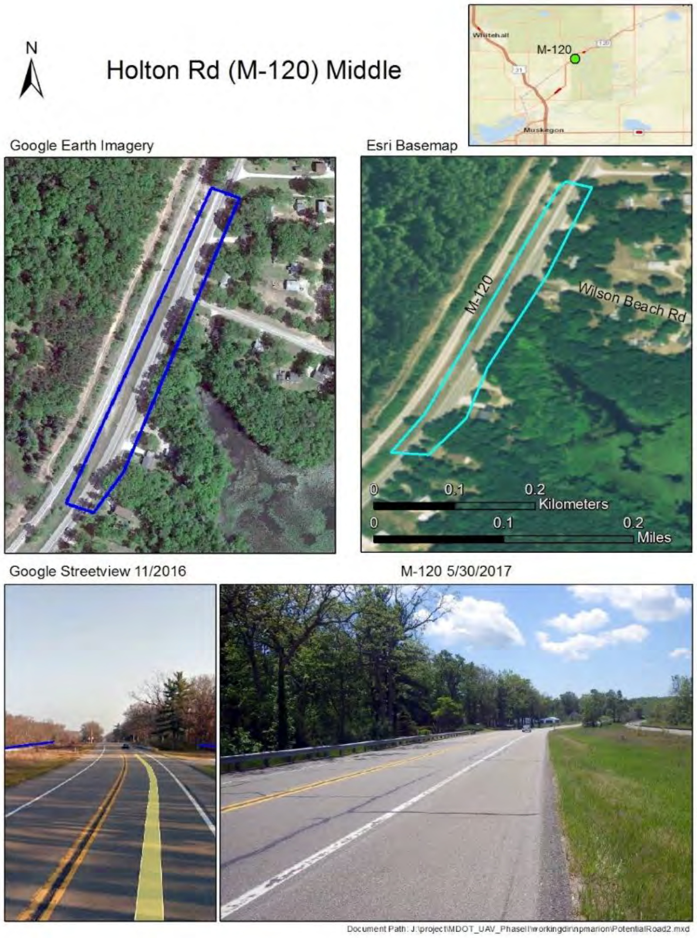


Figure 17: Holton Road Corridor map created for planning the UAV data collection at this site.

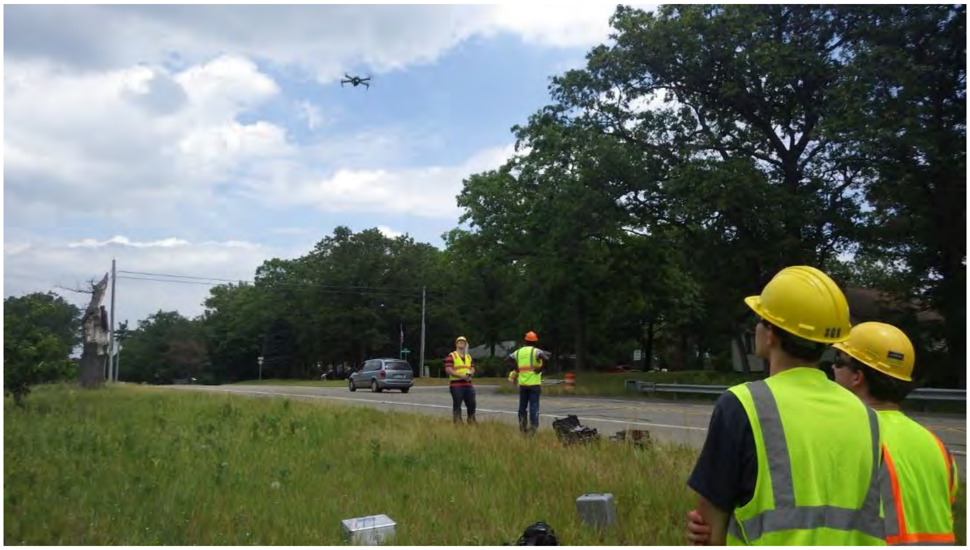


Figure 18: Collecting oblique imagery of the roadway for an overview map of the M-120/Holton Road corridor site with the Mavic Pro UAV.

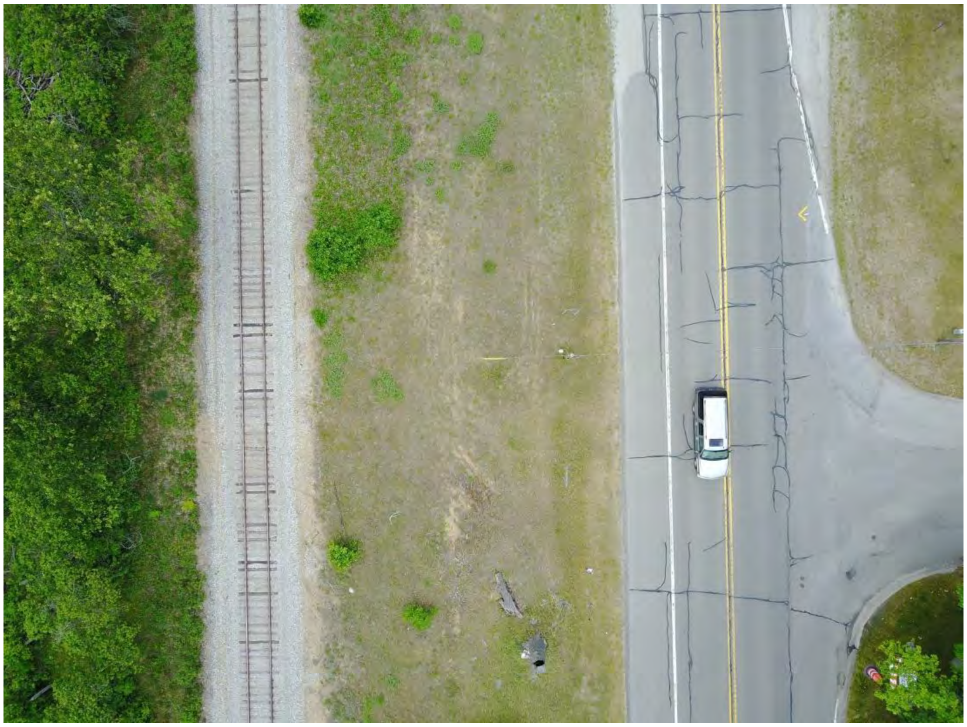


Figure 19: Photo collected of the Holton Road site using the Mavic Pro UAV while flying over the MDOT right-of-way between the highway and a rail line, showing the wide field of view possible for site mapping.

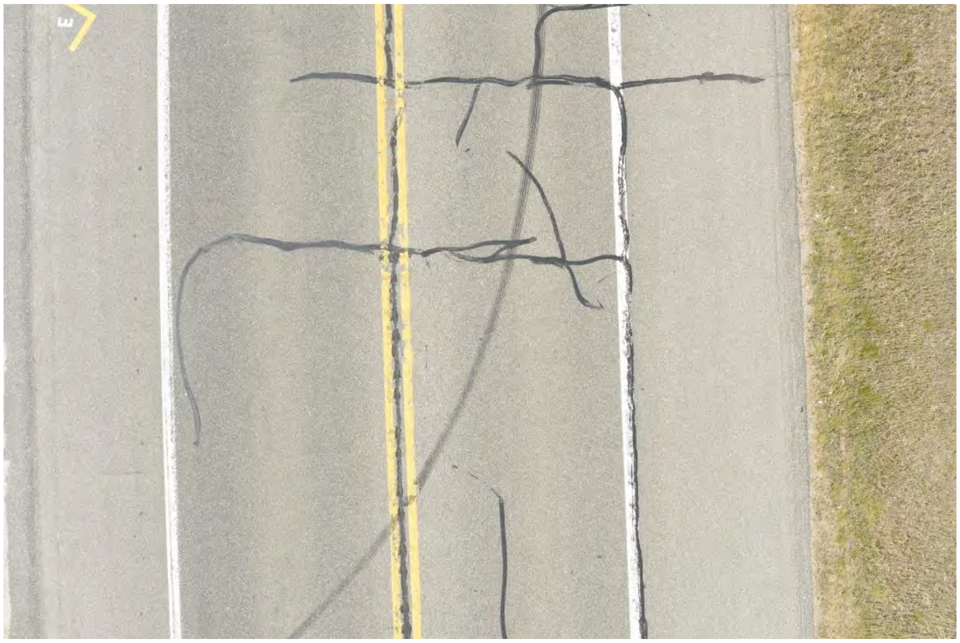


Figure 20: Nikon D810 image of Holton Rd collected via UAV showing sealed crack details.

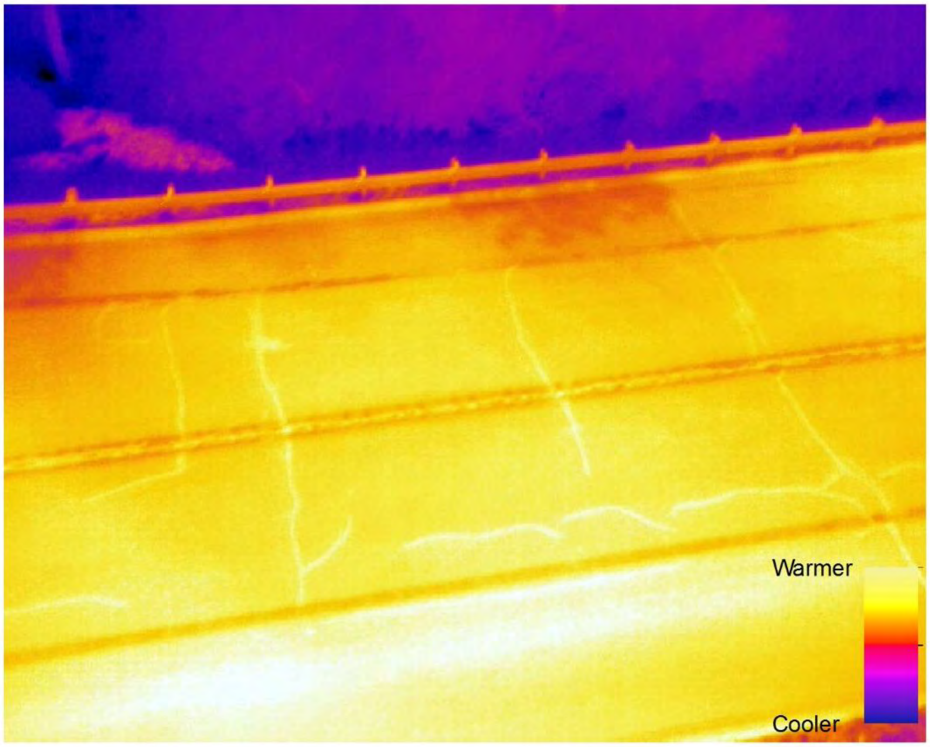


Figure 21: Thermal image of Holton Rd collected with the Bergen Hexacopter; warmer areas (such as sealed cracks) are in brighter yellow colors while cooler areas are in purple (such as vegetation in the background).

7. US-31 / White River Corridor – STR 7587
Date of Collection – June 15, 2017

At this site, data along a road corridor of US-31, 14 miles north of Muskegon (near Whitehall, MI), were collected including a bridge over the White River (Figure 22). By providing a right-of-way along a road corridor, and a bridge, this site helped combine a demonstration of longer-distance monitoring with a bridge survey. The bridge has a length of 169 feet, a width of 43 feet and a concrete deck. A 2014 detailed MDOT bridge scoping report indicates that 13.6% of the bridge has delaminations and approximately 300 square feet of concrete patching; for a bridge deck rating of 4 (Poor condition). For this site, imagery using the Nikon D810, DJI Phantom 3A, DJI Mavic, Flir Vue Pro, Flir Vue Pro R, Flir Duo, and GoPro Hero3 were collected, including traffic video. Figures 23, 24, 25, 26, 27, and 28 show some of these UAV-enabled data collection capabilities. GIS-based layers created include an optical orthophoto, DEM, and hillshade were created. Thermal orthophotos with both relative temperature values and radiometrically calibrated per-pixel values and an orthomosaic of the underside of the bridge were also created.

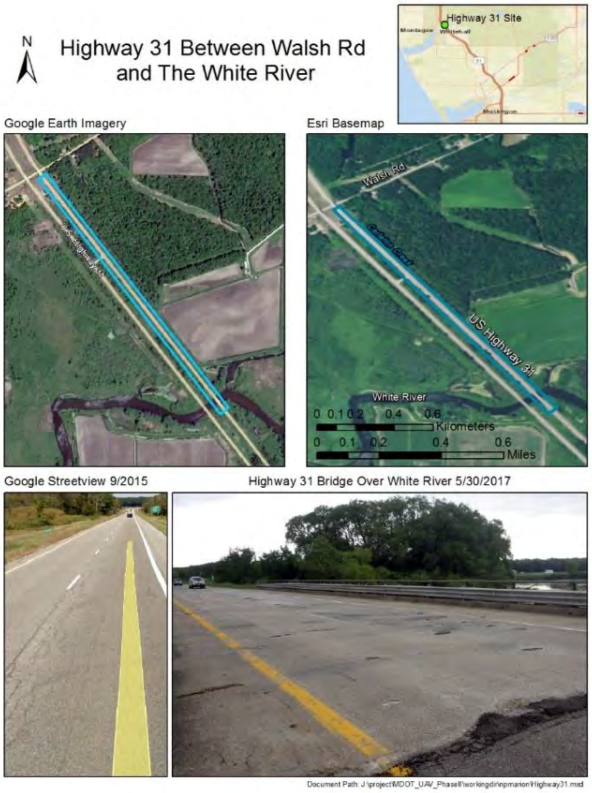

N
Highway 31 Between Walsh Rd and The White River
Highway 31 Site
Montague
Whitehall
Muskegon
Google Earth Imagery
Esri Basemap
Walsh Rd
US Highway 31
White River
0 0.1 0.2 0.4 0.6 Kilometers
0 0.1 0.2 0.4 0.6 Miles
Google Streetview 9/2015
Highway 31 Bridge Over White River 5/30/2017
Document Path: J:\project\MDOT_UAV_PhaseII\workingdir\npmarion\Highway31.mxd

Figure 22: US-31 Road Corridor near Muskegon and Whitehall, MI (US-31 / White River bridge) map created for planning the UAV data collection at this site.

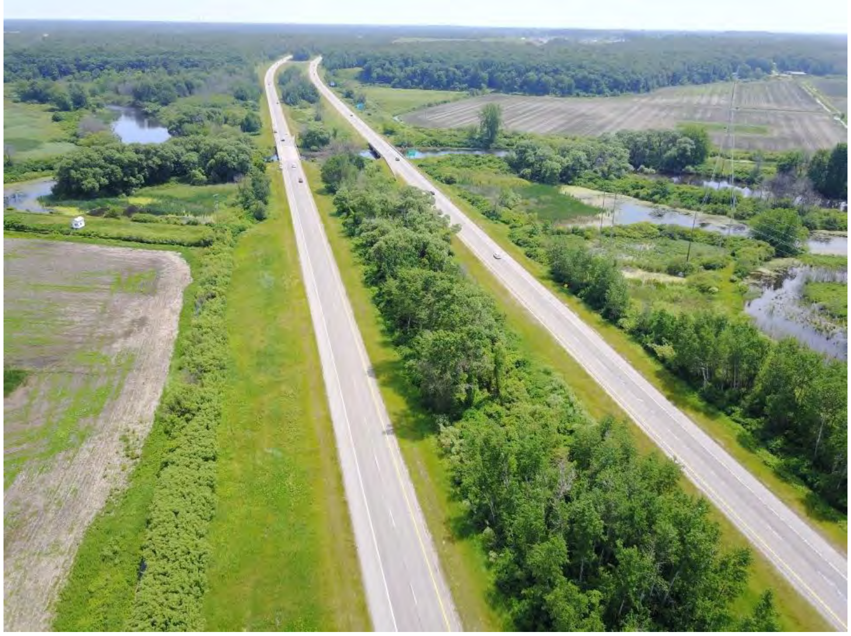


Figure 23: The main corridor area captured through use of US31 / White River site, as imaged by the Mavic Pro quadcopter. The left grassy area was used for UAV deployment, and the assessed bridge is visible over the White River in the background.

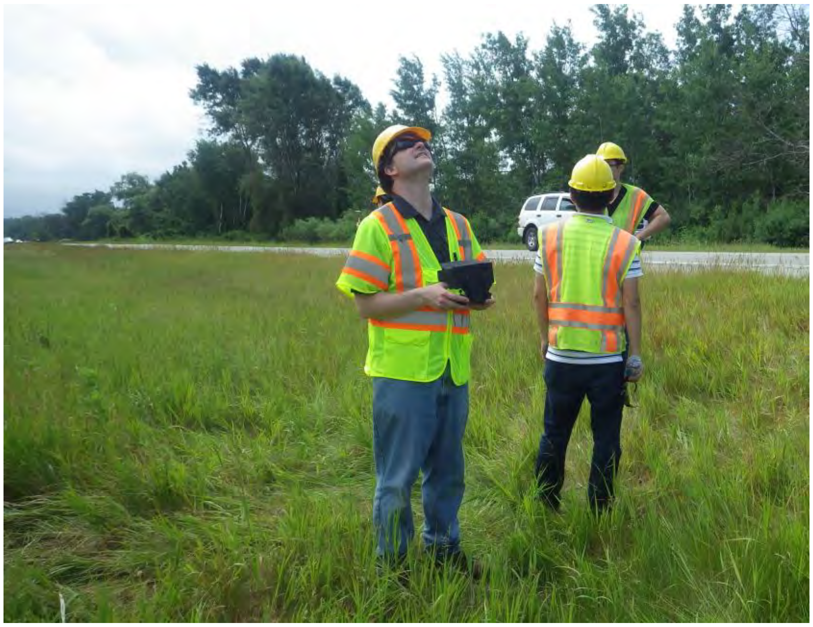


Figure 24: Example of operating the Mavic Pro safely from the US-31 right-of-way; the operator is looking towards the UAV while also using the FPV screen to enable live viewing of what the UAV is seeing while it collects imagery and video data.

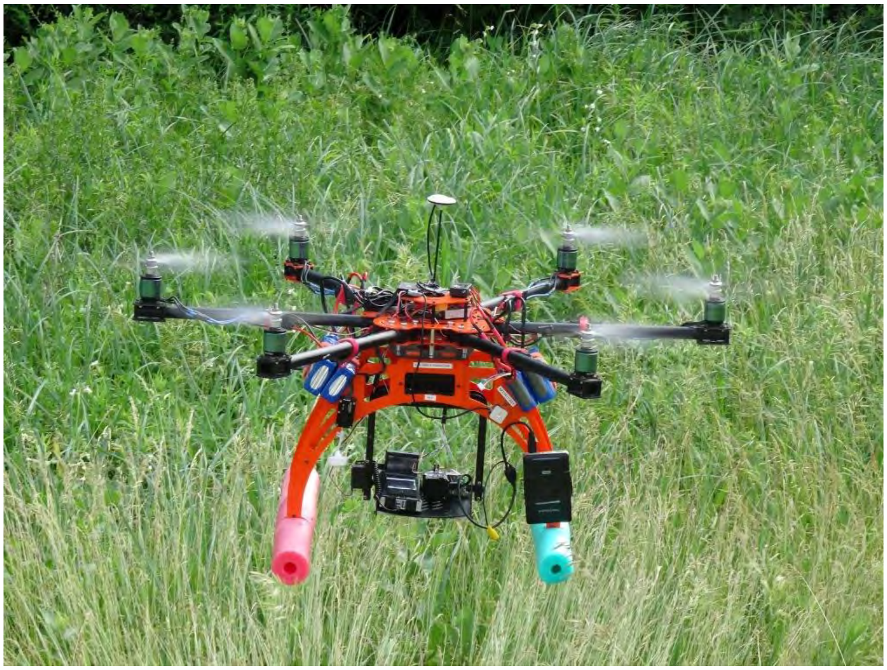


Figure 25: A Bergen Hexacopter shortly after take-off, about to collect thermal imagery of the US-31 road corridor with dual FLIR cameras.

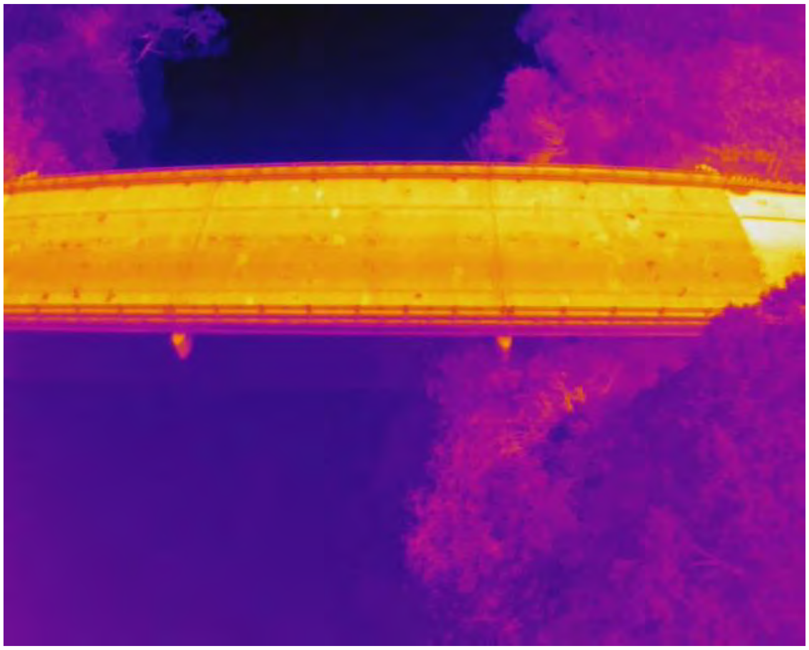


Figure 26: Thermal image of the US-31 / White River north-bound bridge collected with the FLIR Vue Pro R sensor, with different temperatures on the bridge deck potentially indicating defects. Brighter yellow colors are warmer while darker purple colors are cooler. With the Pro R version, these colors can be turned into per-pixel temperature values.

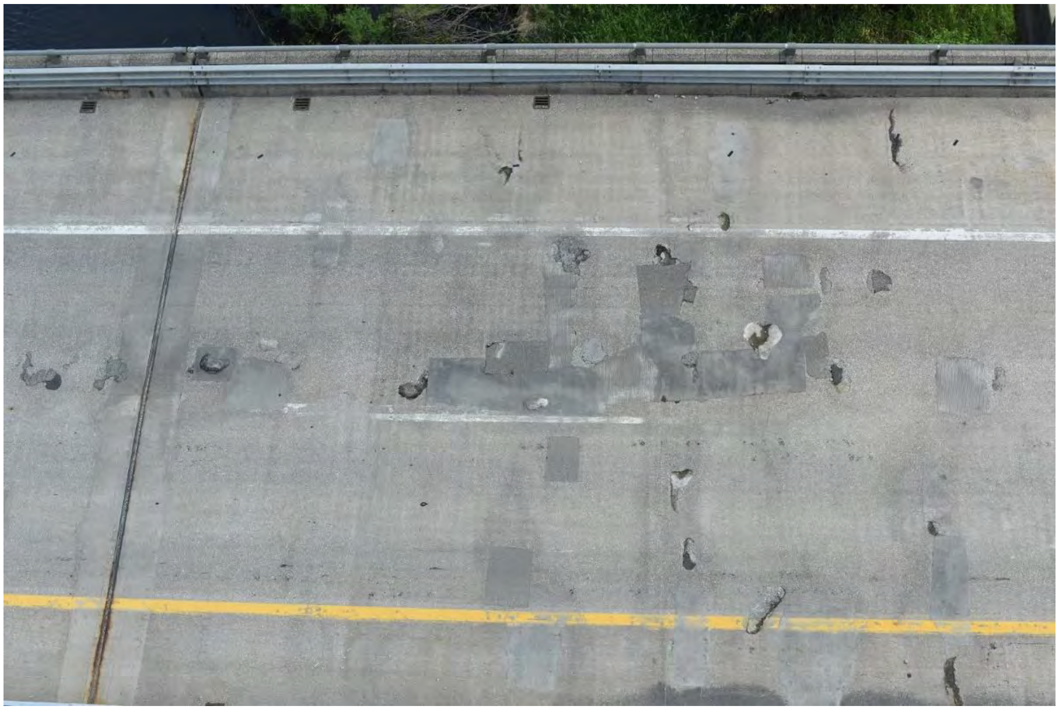


Figure 27: Bridge deck photo collected with Nikon D810 from the Bergen Hexacopter, used to create a detailed orthophoto and DEM of the bridge surface.

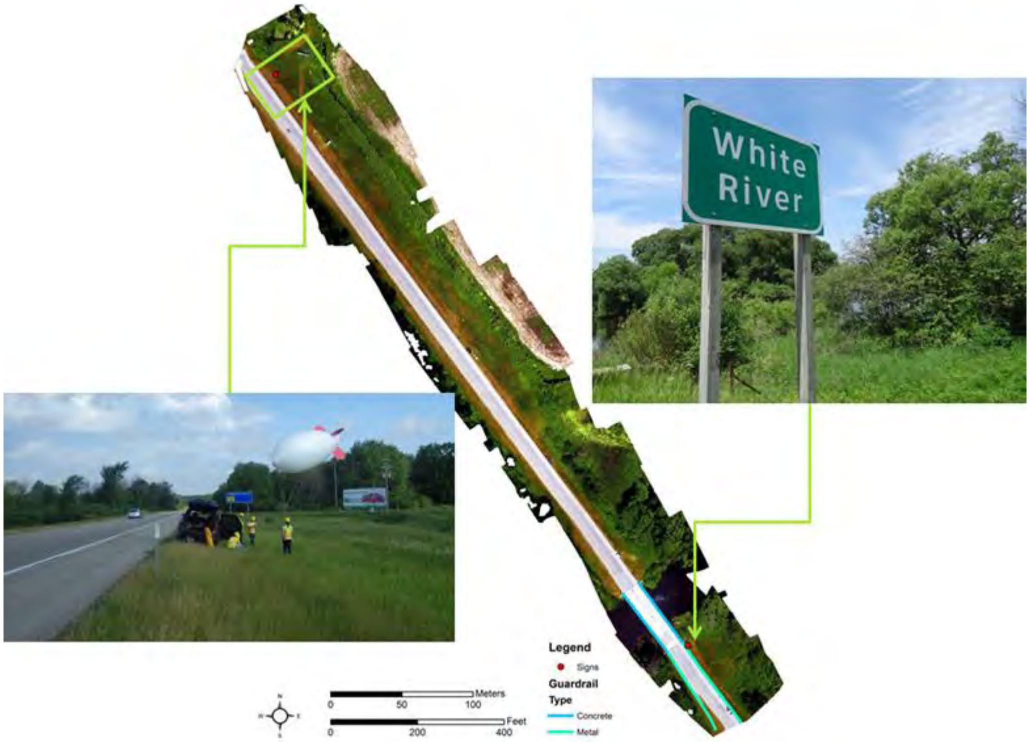


Figure 28: Detected road furniture along the US-31 corridor.

8. Merriman Road (East U-Turn) – STR 11516

Date of Collection – June 6, 2014

In Phase 1 of the project, data were collected in 2014 from a Bergen hexacopter with a Nikon D800 camera over bridges that were about to have their decks replaced as part of the "I-96 Fix" project in Livonia, MI. The Nikon D800 data were re-used from the project to provide an additional example of a bridge sensing project, this one with significant deck defects at the time of data collection. The bridge on Merriman Road over I-96 has a length of 176 feet, a width of 22 feet and a concrete deck (Figure 29). A 2014 MDOT bridge inspection report indicated the bridge surface was over 30% spalled and bit patched, for a bridge surface rating of 4. Although this data was collected in June 2014, the imagery was still utilized for this project. For this site, imagery using the Nikon D800 and FLIR Tau2 sensors were collected, which were available to the Phase 1 project at the time. GIS-based data layers include an optical and thermal orthophoto, DEM, and hillshade model of the site.

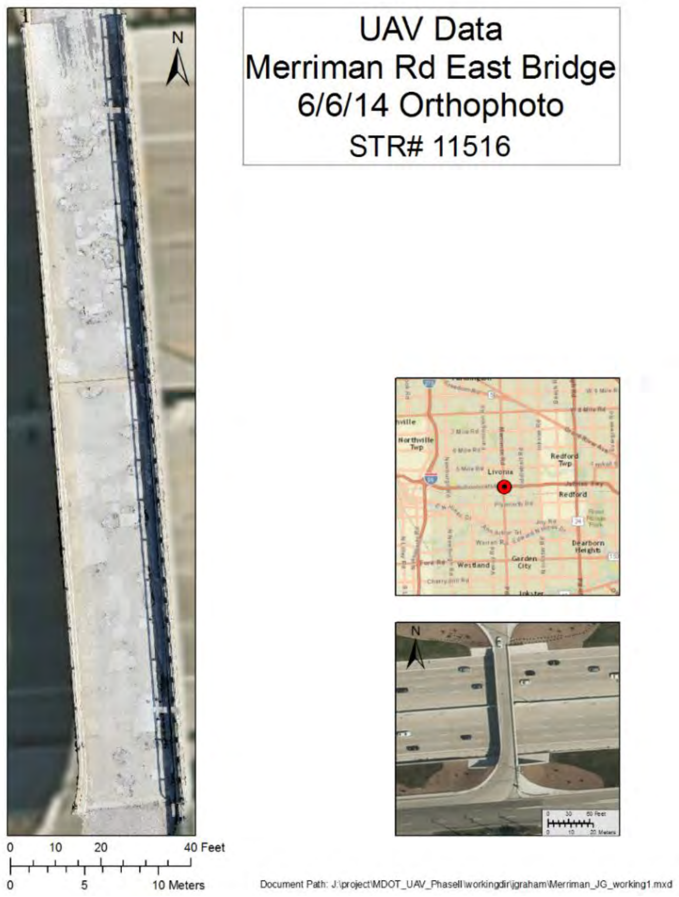


Figure 29: Merriman Road Bridge map of the 2014 Phase I study location.

9. US-23 MTRI Highway Corridor
Dates of Collection – October 14, 2015, November 11, 2015, September 6, 2016, November 4, 2016, August 11, 2017, and November 10, 2017
This site was used to collect traffic videos of north and southbound traffic along US-23 (Figures 30 and 31), adjacent to the MTRI facility in Ann Arbor. UAV-based video was simple to collect here as the highway was easily visible while flying from the MTRI property. The videos were used as training data for the traffic monitoring algorithm. The Mavic Pro was primarily used for these collections since it records videos with 4k resolution, however, video was also collected using the GoPro Hero3 camera in 1080p resolution.

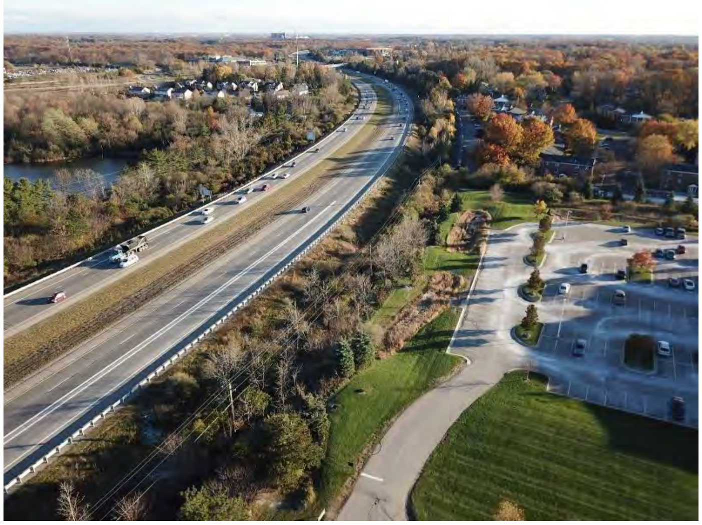


Figure 30: US-23 Road Corridor (looking south) as visible from a MTRI UAV.

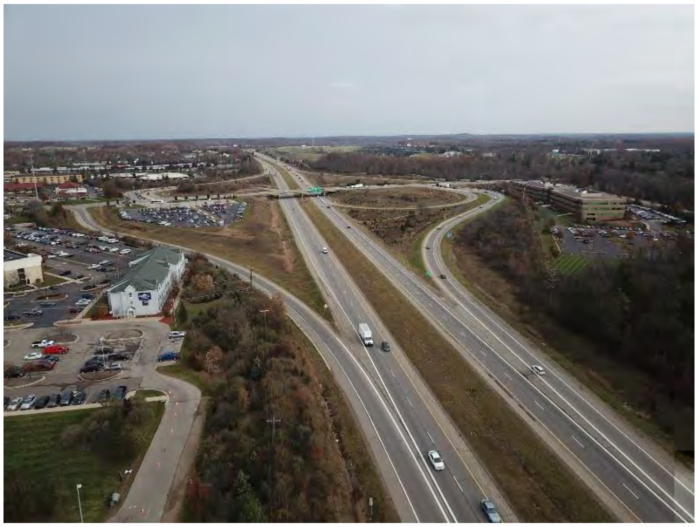


Figure 31: US-23 Road Corridor (looking north) as visible from a MTRI UAV.

10. Lidar Test at MTRI

Date of Collection – November 17, 2017

The Bergen Hexacopter with an onboard Velodyne LP-16 lidar unit was flown near US-23 at the MTRI facility in Ann Arbor along a roadway to demonstrate the potential for this method of 3D data collection after equipment issues prevented a successful data collection at the June 2017 field work near Muskegon. Figure 32 shows the setup used to fly the Velodyne LP-16 unit on a November 17, 2017 data collection. This setup used a new test method of connecting an onboard GPS with inertial measurement unit (IMU) with the lidar unit vs. the June setup.

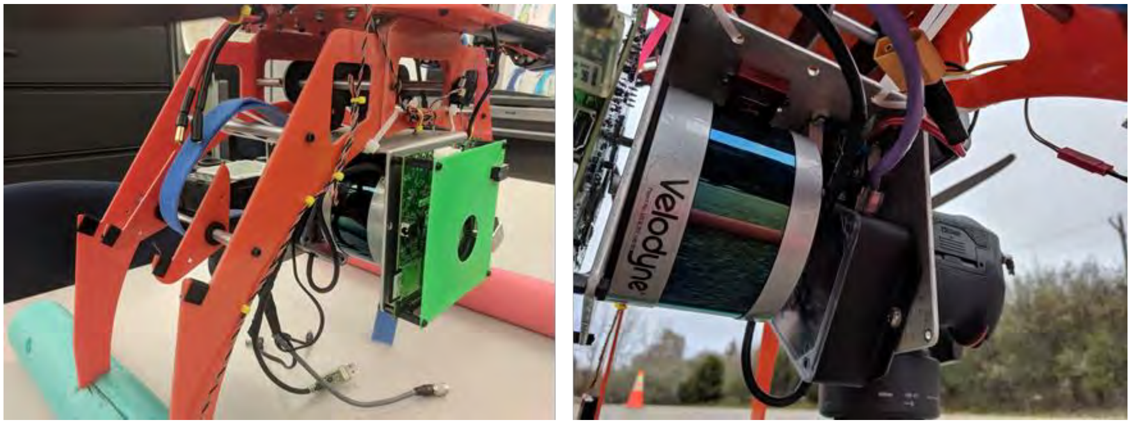


Figure 32: Velodyne LP-16 lidar unit mounted in the Bergen Quad-8 UAV about to collect data for roadway behind the MTRI facility.

An example of the point cloud created with the current lidar setup is shown in Figure 33, using the Quick Terrain Modeler viewer program. The edge of the MTRI building can be seen on the left, the roadway in the middle, and a line of trees and bushes on the right before the edge of US-23 is seen. However, the scan lines are clearly visible, and not well aligned, particularly at the edges of the data collection. The project team's analysis of the results is that this is due to insufficient accuracy in the quality of the combined GPS plus IMU data that is necessary to align individual lidar scan lines as the unit's sensor sweeps over the landscape underneath the UAV.

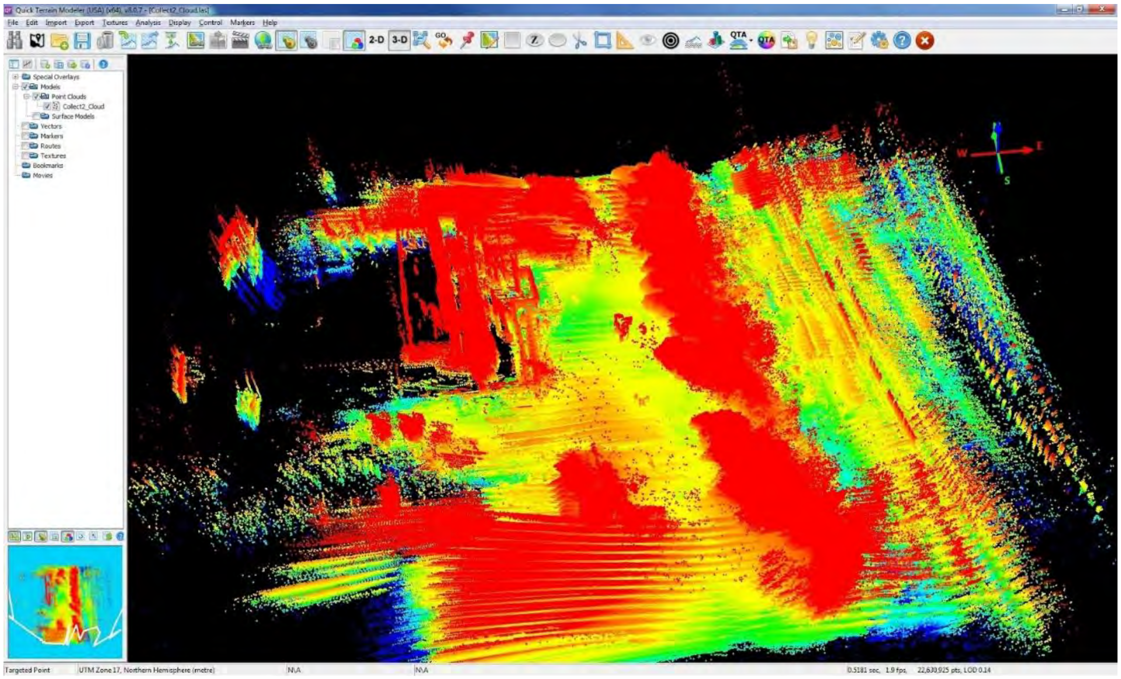


Figure 33: A view of the lidar point cloud collected with the November, 2017 version of the lidar setup mounted on the Bergen Quad-8 UAV.

Figure 34 compares the results between a hillshade version of a DEM generated by the Nikon D810 using photogrammetry vs. a hllshade version of a DEM created with the lidar point cloud using the VLP-16 lidar unit. While the DEM from photogrammetry using the Nikon D810 imagery is more detailed and accurate, the current lidar setup does produce a recognizable point cloud that can be used to create a 3D surface. The project team anticipates that with improved positioning from an updated GPS with IMU, the point cloud positioning will improve to about 1cm positional accuracy, significantly increasing the usefulness of the lidar data as a practical tool for 3D data creation. Further post-processing via higher-level point cloud registration tools, such as iterative closest point and RANSAC, could further improve the fidelity of the lidar-based DEM.

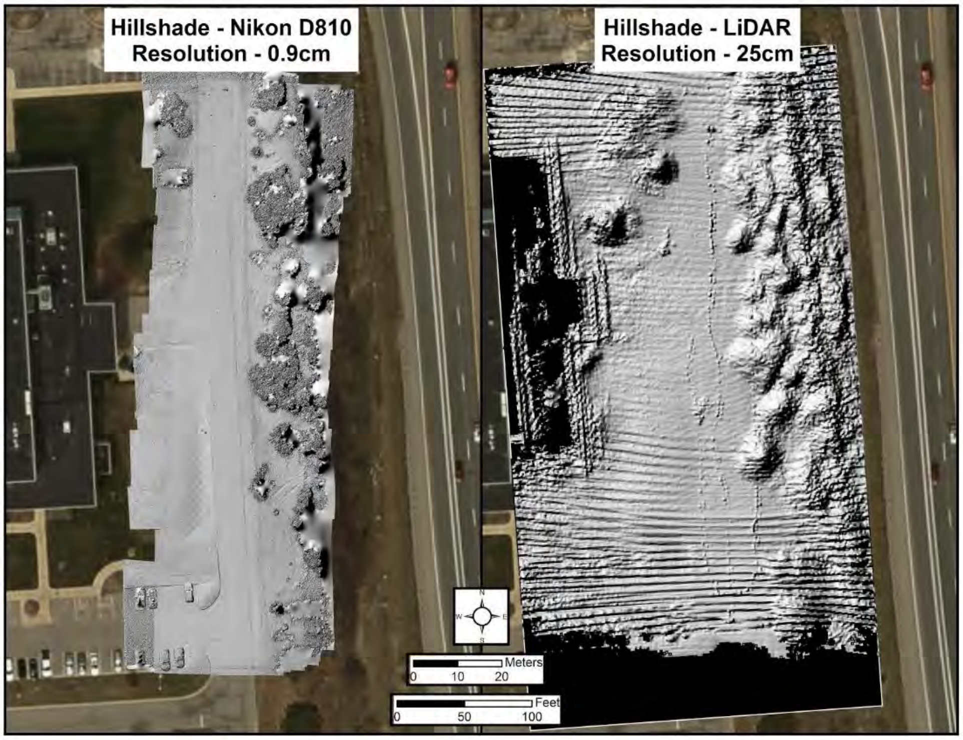


Figure 34: Comparison of a hillshade product for the same area using photogrammetric data collected using a Nikon D810 camera vs. the November, 2017 version of the lidar setup flown aboard the same Bergen Quad-8 UAV.

3.2 UAV Platforms and Sensors

In order to collect data at the sites listed above, multiple UAV platforms and sensors were used in the field. The following section provides a brief overview of each platform and sensor used during Phase 2 of the project. More information on the operation of the main UAV platforms and sensors is included as part of Appendix section 9.2 ("UAVs and Sensors Operations Manual"), featuring details on the Bergen Hexacopter, Bergen Quad-8, and DJI Mavic UAVs along with the Nikon D810, FLIR Vue Pro / Pro R, and Velodyne LP-16 LiDAR sensors.

Platforms

Bergen Hexacopter

The Bergen Hexacopter (Figure 35) is manufactured by Bergen RC located in Vandalia, Michigan. It has a typical data collection flight time of 16 minutes which leaves a 25% battery reserve. A two-axis gimbal allows for multiple sensor types to be mounted and are stabilized for roll and pitch movements. Payloads of up to 5 lbs can be flown for data collections. Under this project, a Bergen hexacopter was purchased for use by the project and MDOT, but two other Bergen hexacopters were also available originally purchased for a previous USDOT unpaved roads project (Dobson et al. 2014, Brooks et al. 2016; more information is at http://www.mtri.org/unpaved).

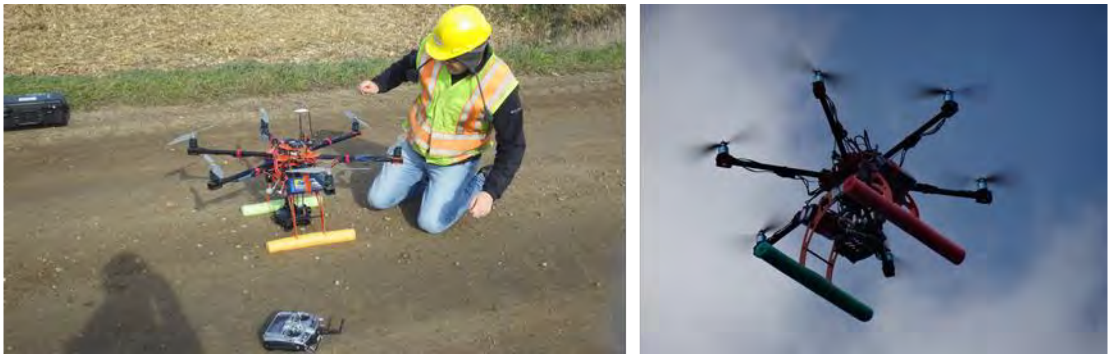


Figure 35: Example of a Bergen Hexacopter being prepared to collect data (left) and collecting data (right) over project sites.

Bergen Quad-8

The Bergen Quad-8 octocopter is also manufactured by Bergen RC (Figure 36). It has a typical data collection flight time of up to 20 minutes which leaves a 25% battery reserve. A two-axis gimbal allows for multiple sensor types to be mounted and are stabilized for roll and pitch movements. Payloads of up to 10 lbs can be flown including multiple sensors (i.e. the Nikon D810 and FLIR VUE Pro) at one time for data collections.

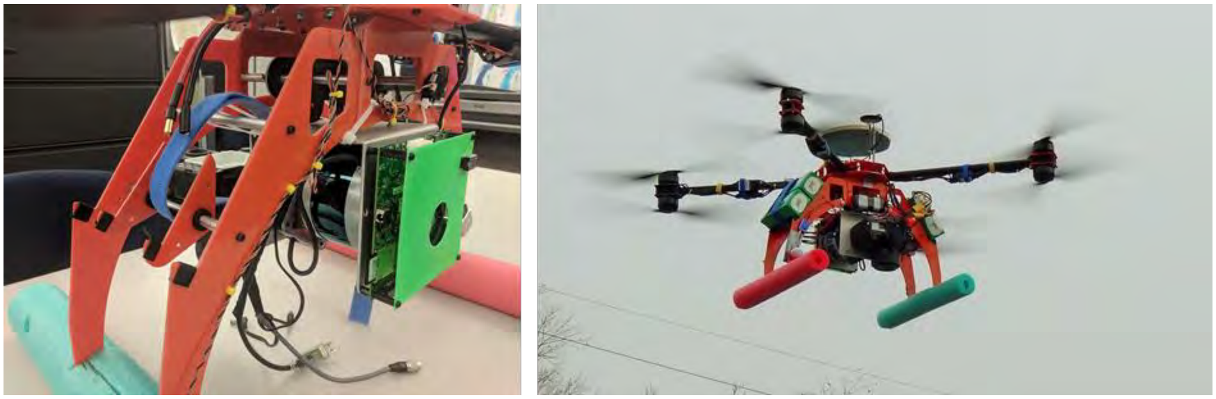


Figure 36: Quad-8 UAV with mounted lidar system (left) and deployed over the MTRI roadway site, collecting lidar data (right).

Phantom 3A

The DJI Phantom 3 Advanced is a quadcopter equipped with a 12.4MP camera capable of 2.7k video (Figure 37). It has a flight time of up to 23 minutes and a maximum speed of 35mph. It provides high resolution FPV video to the user and contains a three-axis gimbal for camera stabilization.

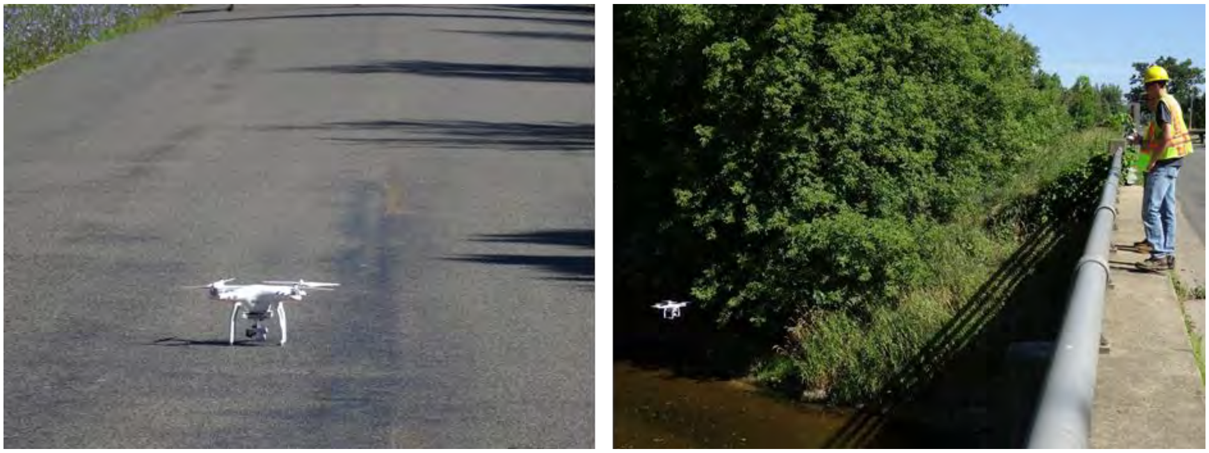


Figure 37: The DJI Phantom 3A on a bridge deck (left) and inspecting the side of a bridge (right).

Mavic Pro

The DJI Mavic Pro is a lightweight quadcopter equipped with a 12.35MP camera capable of 4k video (Figure 38). It has a flight time of 27 minutes and a maximum speed of 40mph. In addition, it is equipped with a forward and downward collision avoidance system. It provides high resolution FPV video to the user and contains a 3-axis gimbal for camera stabilization.

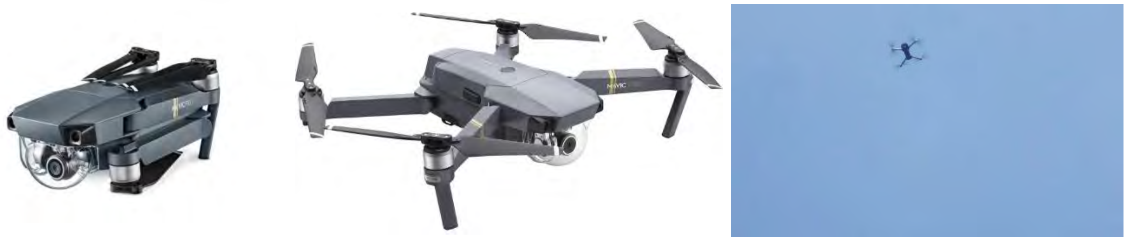


Figure 38: The DJI Mavic Pro, in folded mode for easy transport (left), in ready-to-fly mode (center), and flying over the US-31 corridor site (right).

Mariner 2 Splash

The Mariner 2 Splash is a lightweight waterproof UAV capable of landing and taking off in water environments and flying in light precipitation (Figure 39). However, once it lands in the water, the vehicle is not navigable and is susceptible to currents and flow. The platform does not come with an onboard sensor, but contains an onboard waterproof gimbal capable of holding a GoPro camera. Overall flight time ranges between 12 and 19 minutes and it also has GPS and return to launch functions.

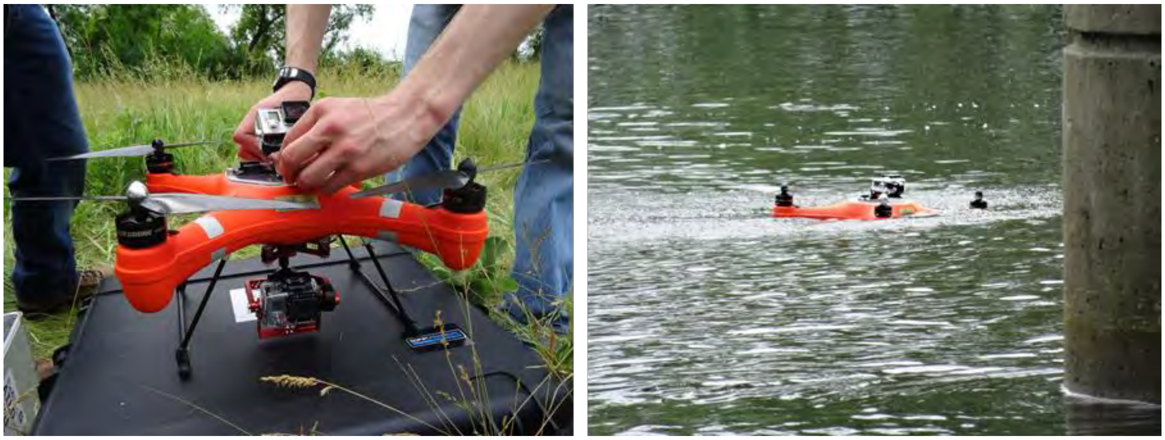


Figure 39: Waterproof Mariner 2 Splash being prepared for flight (left) and while collecting data after having landed on the water to image the underneath of the US-31 / White River bridge using GoPro cameras (right).

Sensors

Nikon D810

The Nikon D810 is a commercially available Digital Single-lens Reflex (DSLR) camera with a 36.3MP sensor (Figure 40). This camera has the capability to continuously capture images at a rate of up to two frames per second. An Aokatec AK-G1 GPS unit is attached which geotags images as they are taken. Typically, a 50mm prime lens is attached for a standard pavement bridge collects. The resulting imagery has a ground sample distance of 3mm from a flying altitude of 100ft.

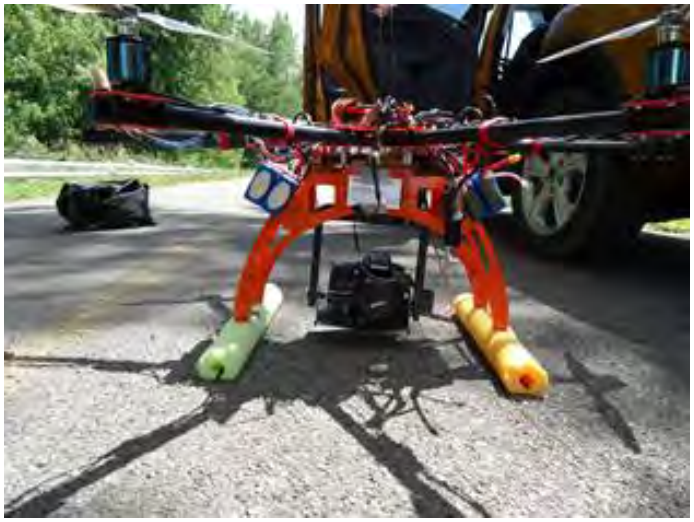


Figure 40: The Nikon D810 attached to the Bergen Hexacopter.

Phantom 3A Onboard Camera

DJI's Phantom 3A integrated camera has a 94° field of view and collects 12.4MP image frames (Figure 41 and 42). At 50 feet above ground level (AGL), it achieves about a ⅖ inch GSD and captures approximately 115 square feet per frame. At the maximum UAS flying height permitted by the FAA, 400 AGL, the DJI Phantom 3A achieves about a 2.6 inch GSD and captures approximately 860 square feet per frame. It also takes videos with 2.7k resolution at 30 frames per second.

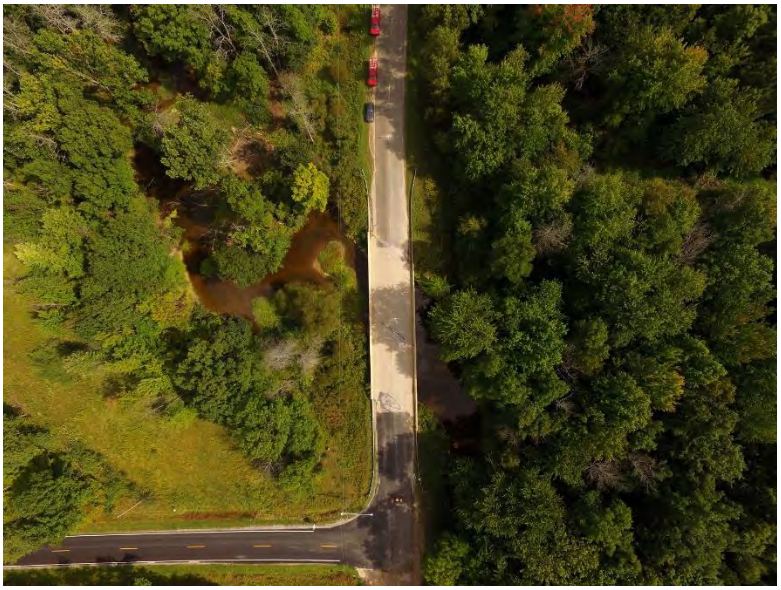


Figure 41: Example image from the Phantom 3A of the Gordonville Road bridge.

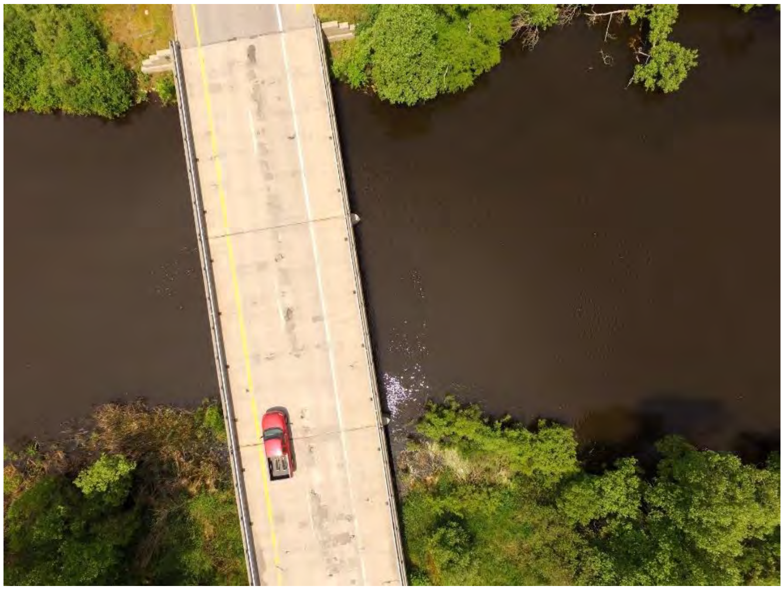


Figure 42: Example image from the Phantom 3A of the US-31 / White River bridge, taken while flying over water near the bridge.

Mavic Pro Onboard Camera

DJI's Mavic Pro integrated camera has a 78.8° field of view and collects 12.35MP image frames (Figure 43 and 44). At 50 feet AGL, it achieves a ¼ inch ground sampling distance (GSD) and captures 80 square feet per frame. At the maximum UAS flying height permitted by the FAA, 400 AGL, the DJI Mavic achieves a 2 inch GSD and captures 650 square feet per frame. It also takes videos with 4k resolution at 30 frames per second.

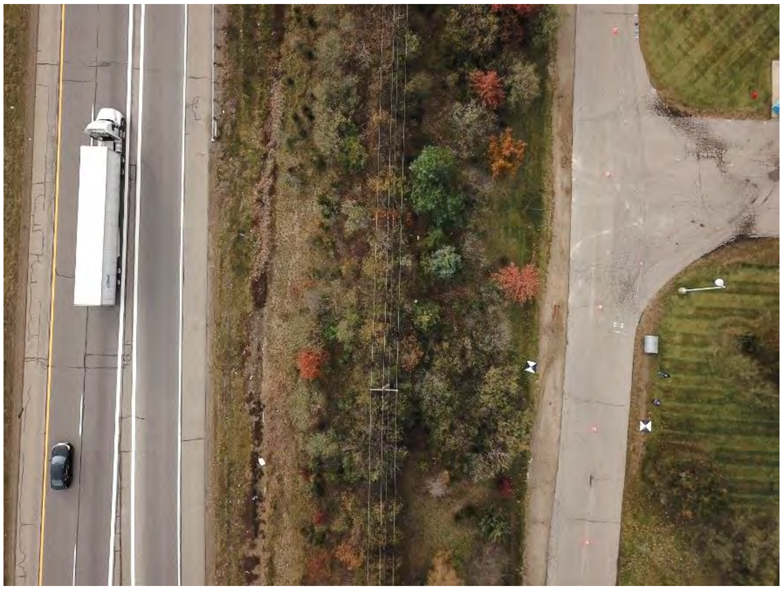


Figure 43: Example image from the Mavic Pro of US23 (left) and the MTRI roadway (right) used for lidar data collection in November, 2017.

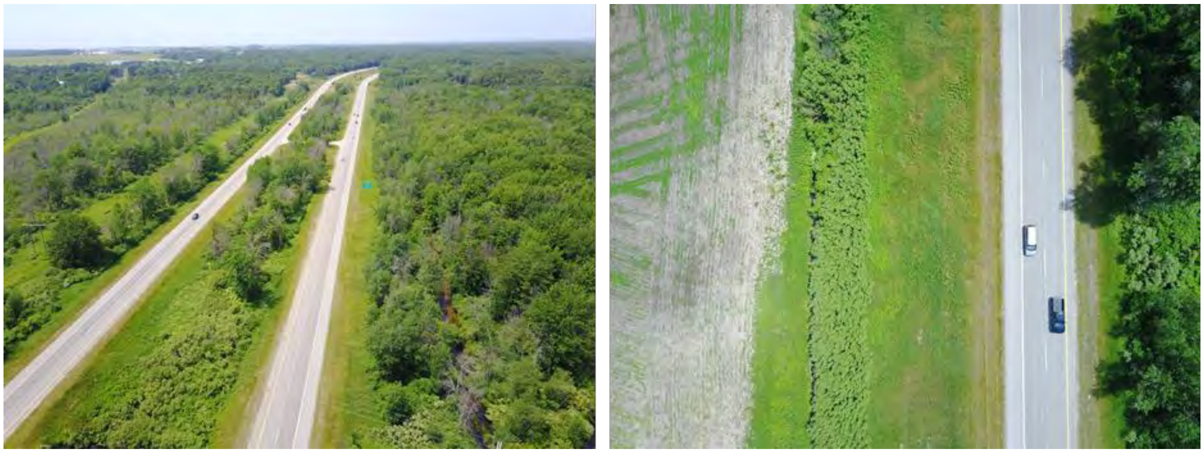


Figure 44: Two views of the US-31 / White River project location collected with the Mavic Pro UAV.

FLIR Vue Pro

The FLIR Vue Pro is a lightweight, low power consumption thermal camera designed for use onboard UAVs, with the one available for this project having a 13 mm lens and a 45º x 37º field of view and a 640 x 480 pixel sensor array (Figure 45). The output includes video (up to 30 frames per second) and still images of non-calibrated digital numbers that scale linearly with radiance (see next section on FLIR Vue Pro R for the calibrated version). An output example of a raw image is shown in Figure 46. The small size (2.26 X 1.75 inches) allows for easy mounting in UAV platforms. As part of Task 1A "Collect data from the UAV platform using sensing technology in near-time (as real-time as can be achieved) demonstrating, developing, and implementing storage capabilities of large amounts, usage, and application development of data," the sensor was connected to on-board UAV video transmission radios for live viewing of the thermal data in real time, as seen in Figure 47. Despite the lack of radiance or temperature calibration, the camera is particularly useful in tasks where relative differences in temperature are relevant, e. g. for detection of delaminations in concrete structures, as in such cases only the difference in temperatures is relevant for the detection process (see example in Figure 48).

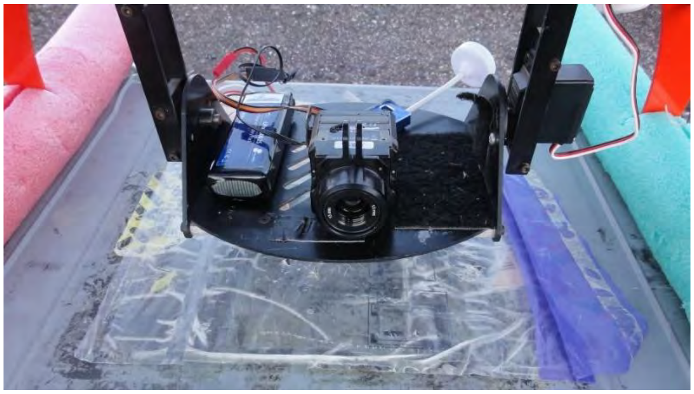


Figure 45: The FLIR Vue Pro onboard of the Bergen Hexacopter.

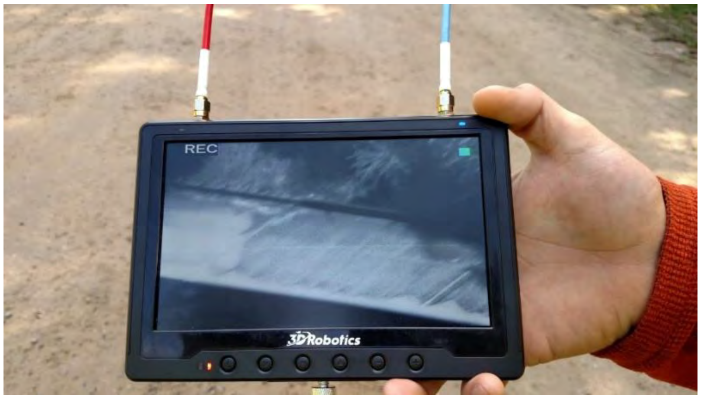


Figure 46: Thermal imagery of a bridge deck being shown live via FPV to the operator and nearby project staff.

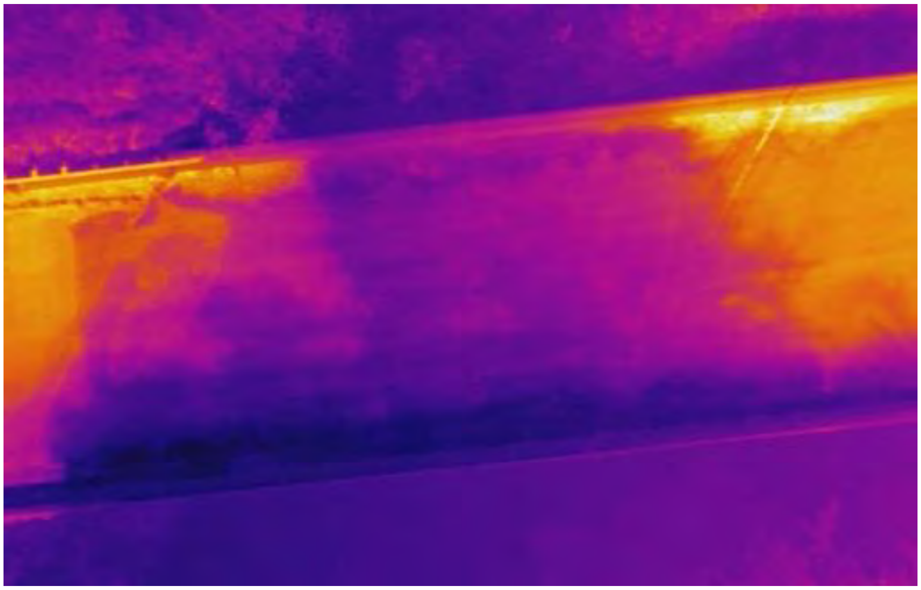


Figure 47: Example image of part of the Gordonville Road bridge deck. Note the extensive shadowing (dark center of the image, with cooler purple colors) that make analyzing the image for delaminations challenging.

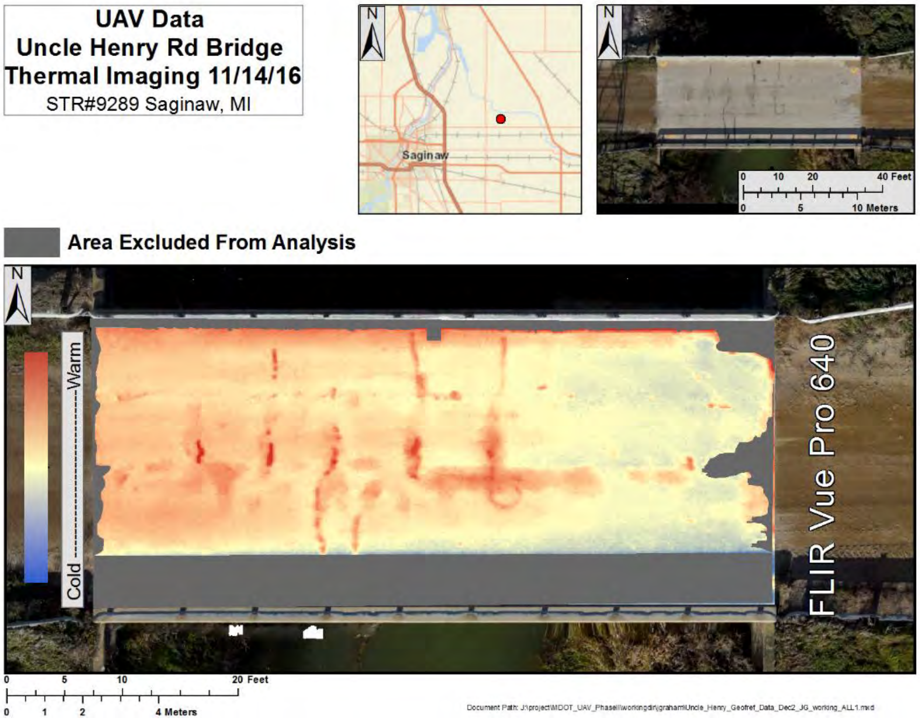


Figure 48: Example of merged and georeferenced thermal images captured with the FLIR Vue Pro aboard a Bergen Hexacopter used to create a map of relative temperatures for the Uncle Henry Road bridge deck. Red values are warmer, blue values are cooler.

FLIR Vue Pro R

Virtually equal to the FLIR Vue Pro, but with the additional capability of providing calibrated temperature outputs. This camera is particularly useful for tasks where the actual temperature values are desired. Although this has not been a requirement for the project team's delamination detection method, the FLIR Vue Pro R was tested in the field, to observe what temperature values could be collected, and how this per-pixel temperature value capability could potentially help with delamination detection (Figure 49).

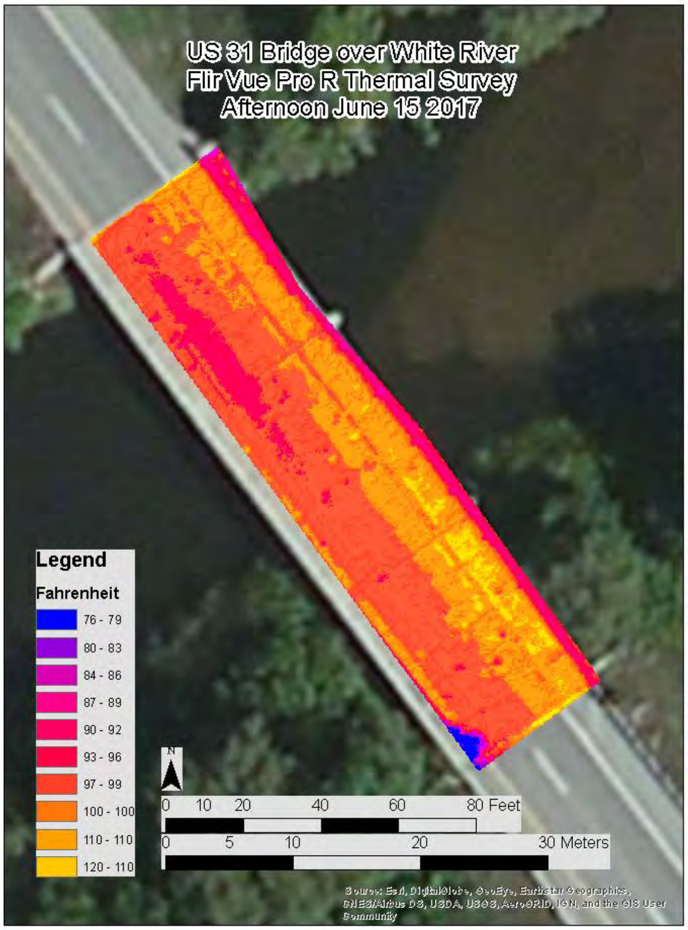

US 31 Bridge over White River
Flir Vue Pro R Thermal Survey
Afternoon June 15 2017
Legend
Fahrenheit
76 - 79
80 - 83
84 - 86
87 - 89
90 - 92
93 - 96
97 - 99
100 - 100
110 - 110
120 - 110
N
0 10 20 40 60 80 Feet
0 5 10 20 30 Meters
Source: Esri, DigitalGlobe, GeoEye, Earthstar Geographics, CNES/Airbus DS, USDA, USGS, AeroGRID, IGN, and the GIS User Community

Figure 49: Example output from having merged several FLIR Vue Pro R images together, converted them to per-pixel temperature values, and georeferenced the result to 3D photogrammetric data also collected at the US-31 / White River site. Note the temperature values are in degrees Fahrenheit, rather than just relative values, due to using a radiometric thermal camera.

FLIR Duo R

The FLIR Duo R is an even more compact and portable (1.2 X 1.6 X 2.3 inches) combined, thermal and optical camera, with a wider lens (57º X 44º field of view), but at a lower resolution (120 X 160 sensor array) for the thermal sensor than the tested FLIR Vue Pro units (note that a higher resolution version has since come out) (Figure 50). Outputs include temperature calibrated video (up to 7.5 frames per second) and still images (1920x1080 resolution = 2 mp). Due to its small size, the FLIR Duo R is useful for deploying for small UAVs (e. g. DJI Phantom) or larger UAVs when a lighter payload is desired. However, the low resolution of the FLIR Duo R vs. the available FLIR Vue Pro units meant that the project team determined it was not useful for bridge deck assessment, at least at an oblique angle (Figure 51).

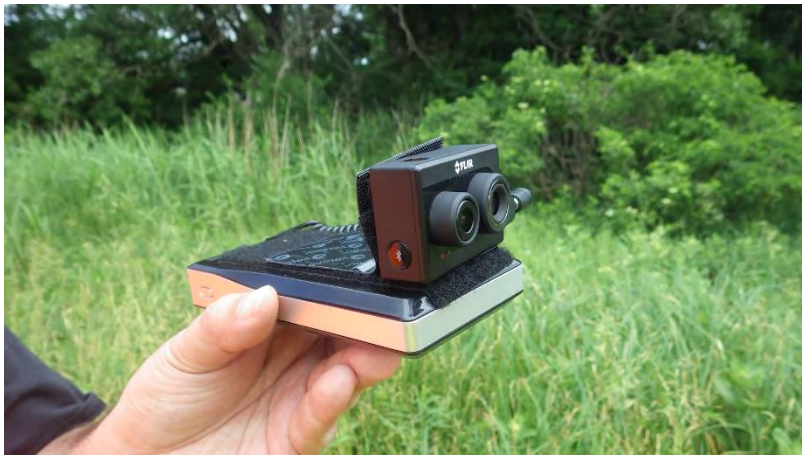


Figure 50: The FLIR Duo Pro.

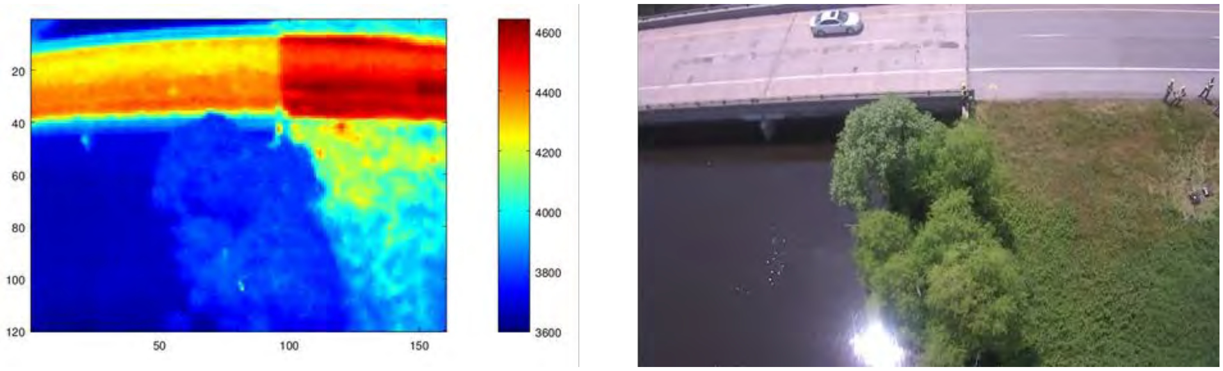


Figure 51: FLIR Duo Pro output with relative thermal values (left) and near simultaneous digital photo (right).

GoPro Hero 3 Black

The Hero3 Black camera can take 12Mp stills and record video at up to 4k with 15 fps. The camera lens has a focal length of 17.2mm and a horizontal field of view of 118.2°. This camera is lightweight which makes it ideal for mounting to UAVs. This sensor was utilized for inspecting the beams and soffits under the White River Bridge while deployed from a waterproof UAV (a Mariner 2 Splash) (Figures 52, 53, and 54).

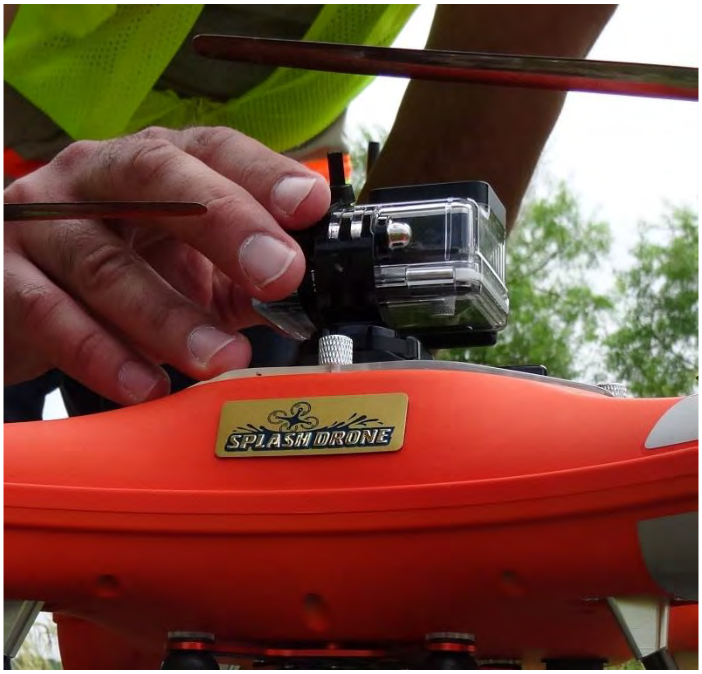


Figure 52: GoPro camera mounted onboard the Mariner 2 Splash UAV, about to collect imagery of the underside of the northbound US-31/ White River bridge.

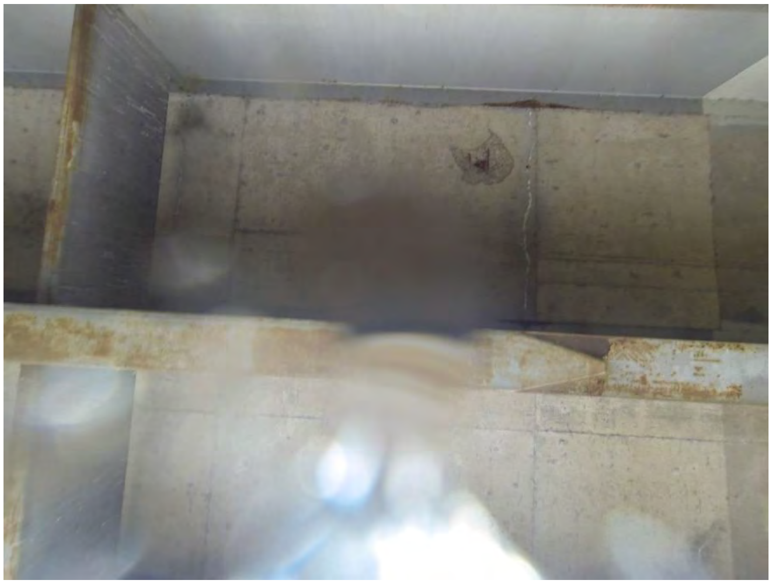


Figure 53: Example photo taken with the top-mounted, upwards facing GoPro while the Splash UAV performed a powered float underneath the test bridge, providing imagery of a spall despite the lens cover being wet.

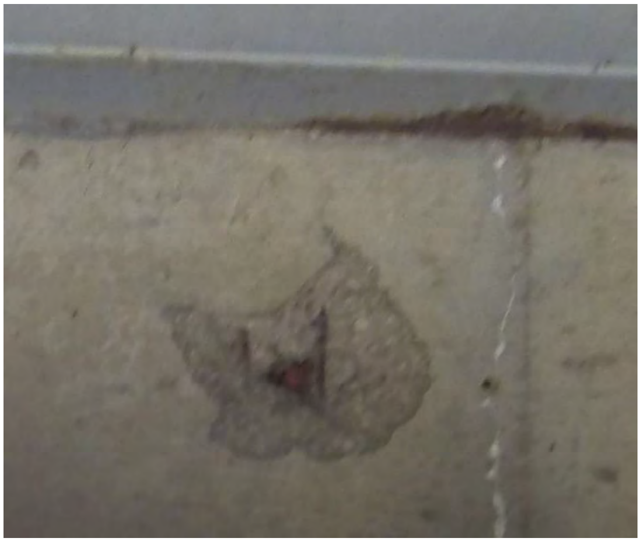


Figure 54: Zoomed-in view of part of the underside GoPro image featuring a spall that was visible using the waterproof UAV, without having to send a person under the bridge.

Velodyne LP-16

The Velodyne LP-16 lidar has 16 lasers spanning field of view of 30 degrees vertically and 360 degrees around the puck (Figures 55 and 56). These lasers spin between 5 Hz and 20 Hz collecting points every 0.4 to 0.1 degrees respectively. The lidar is interfaced via ethernet and configured through a web server hosted on the lidar control unit. The control unit is a Jetson TK1 by NVIDIA, a computer platform with a 192-core NVIDIA GPU and a 4-core 2.3 GHz ARM CPU, which makes it particularly useful for image processing tasks and other GPU workloads. The platform offers several common interfaces such as: a single ethernet interface that we use to collect lidar data and communicate with the device over SSH, a single USB 3.0 that we use to connect the VN200 GPS/IMU and other miscellaneous peripherals, and a single SD card slot that serves as a data repository. The software used to collect the lidar data is the Robotic Operating System (ROS). ROS is a linux-based open source platform for robotics. ROS enables the declarative representation of robots and sensor systems while facilitating communication between system processes. Additionally, ROS has a rich open source community and many community-developed packages. ROS utilizes a publisher / subscriber model of inter-process communication. Nodes subscribe to topics and publish messages to topics.

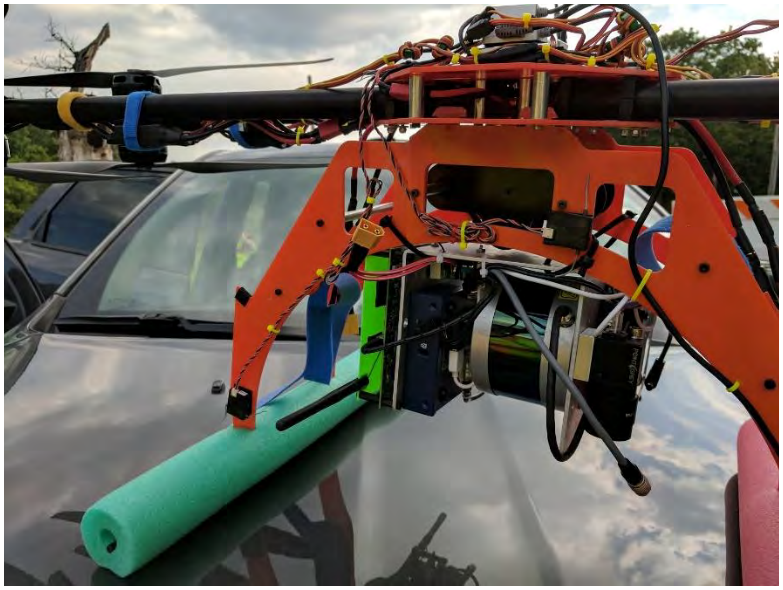


Figure 55: The Velodyne lidar puck on the Bergen Quad-8 with the original June, 2017 setup.

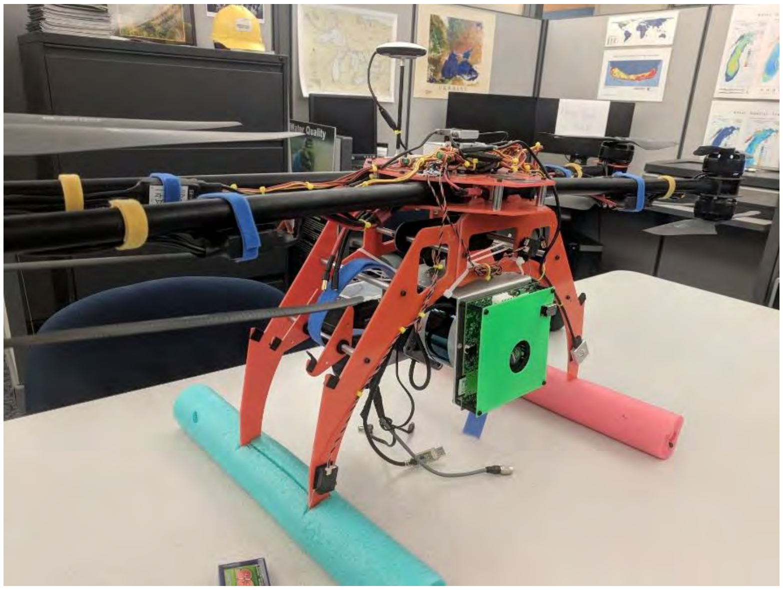


Figure 56: The Velodyne lidar puck on the Bergen Quad-8 in the revised November, 2017 setup. The Jetson TK1 control board is underneath the green protective panel.

GPS Units

The following section overviews the types of GPS used to collect ground control point location data at each site to ensure proper georeferencing of UAV data during post-data collection image processing. Two types of GPS data were collected, one having decimeter (4 in) accuracy (Trimble GeoExplorer 6000) and the other having centimeter accuracy (Leica survey grade GPS).

Trimble GeoExplorer 6000 Decimeter Edition

The Trimble GeoExlorer 6000 used for this project has decimeter (4 in) real time accuracy while collecting carrier phase data (Figure 57). Carrier phase positions with decimeter accuracy are collected over a period of 20 minutes as the Trimble GPS is collecting position data at a rate of once per second. While the survey-grade Leica GPS unit made available through the project's inclusion of SSI has higher accuracy and more rapid data collection, the GeoExplorer served as a useful backup unit in case the SSI unit was unavailable.

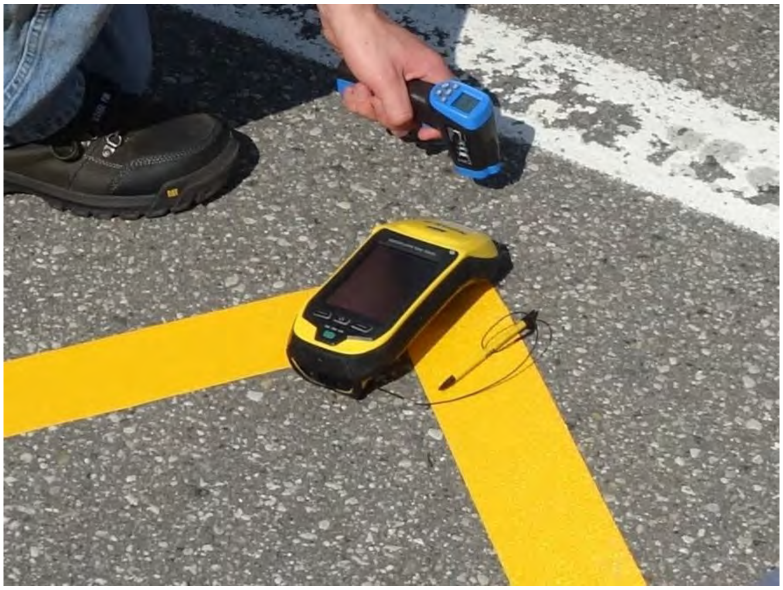


Figure 57: The Trimble GeExplorer GPS unit being used to collect the location of a ground control point. In the background, a hand-held digital thermometer is being used to collect ground-reference data to compare to thermal imaging results.

SSI Leica Survey-grade GPS

SSI provided positioning of the points used to calibrate the UAV imagery (Figure 58). The number of points varied per bridge. Real Time Kinematic (RTK) positioning techniques were used. All RTK observations were tied to the MDOT Continuously Operating Reference Station (CORS) network. All of the positions provided were within the specifications for the RTK CORS positioning which was +/- 0.1ft (3 cm) horizontally and +/- 0.2 ft (6 cm) vertically.

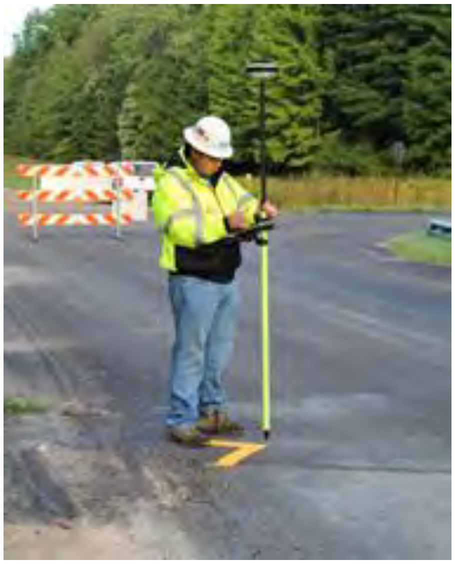


Figure 58: SSI team member A.Semenchuk collecting high accuracy GPS points.

The equipment used consisted of a Leica GS15 receiver coupled with a Lecia GS20 controller and an AT&T SIM card to allow connectivity to the MDOT CORS network. The GS15 receiver is a survey grade Global Navigation Satellite System (GNSS) receiver. A fixed height rod was used to ensure consistency in the measurements. Each position was measured with 4 observations and each observation consisted on 10 epochs. After each observation, the GNSS initialization was reset and a new measurement was taken.

The number of targets varied with each bridge depending on its size with a minimum of four targets for a bridge 200 ft in length or less. The number of targets set was ultimately dictated by the flying height, the coverage of the camera, and the final accuracy required by the project. Mathematically, three targets can be used to properly geo-reference imagery to control. However, based on practical experience, a minimum of 4 targets are required to properly georeference and adjust the imagery to control. With 3 targets there is no redundancy to check the georeferenced solution.

The targets that were set consisted of a chevron 4 in (10 cm) wide with 12 in (30 cm) legs. A special road marking tape for the chevron with the tip of the chevron as area that was positioned was used (Figure 59).

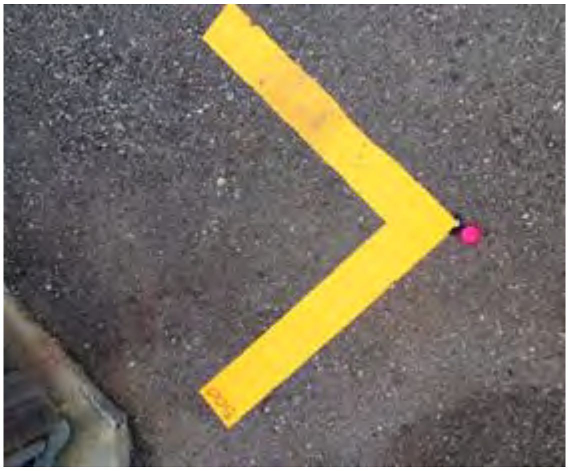


Figure 59: A chevron used to mark where GPS measurements were collected.

The chevrons were configured so that the area of interest for project is fully contained ("boxed in") within the control that was set. For example, of there was a dirt road leading up to the bridge, the target was set on the bridge approach that is supported by the abutment or on the wing walls. The target locations were free from movement as much as possible to ensure proper and accurate positioning.

The ground was swept to remove dirt and debris to ensure proper adhesion of the targets. A blower may also be used to clean the pavement and to dry the pavement if it is damp. It is very important to ensure that the pavement is clean and dry otherwise the target material will not stick and may move causing an erroneous interpretation the tip of the chevron.

All of the collected RTK data was reduced to the Michigan State Plane System and the latest Geoid model applied to calculate the orthometric heights of the point. Using a well-defined coordinate system such as the Michigan State Plane coordinates allows for the data to be plotted easily onto a GIS or other geospatial software for visualization purposes.

4. Products and Analysis Results

The following section overviews the data collected, analyses, and products, with each related directly to their respective tasks. Additionally, descriptions of the algorithms developed under Phase 2 are also provided.

4.1 Field Demonstration Results

As part of the Phase 1 of the project, the team developed six standard GIS output products; the orthoimage, DEM, hillshade, thermal image, detected spalls, and detected delaminations (Brooks et al. 2015, Brooks et al. 2017). As part of Task 5B "Compare data collected from UAV sensors to current data collected and systems used at MDOT for highway assets/operations," the goal was to continue developing the same six products as well as developing additional standard datasets such as point clouds and GPS data, all of which were shared on the project's data portal and MTRI's FTP site. The following sections highlight the data products produced for each site:

Digital Elevation Model

The DEM for each site was created through the processing of the Nikon D810 optical imagery through AgiSoft PhotoScan Pro, a commercially available software that uses structure from motion (SfM) to create 3D point clouds from stereo imagery pairs. To create the DEM, the optical photos are imported into the program and aligned, creating a sparse point cloud. The sparse point cloud is then densified based on the scene structure from the photos and then processed into a mesh for the 3D model, generating a surface that can be exported as a DEM. The DEM was used for the spall detection algorithm (Section 3.6), which compared the change in elevation between pixels to determine if a potential spall exists in a specific spot. Additional processing of the DEM can help with identifying cracking. Table 1 provides the resolution for each DEM created at each bridge/roadway location. Figures 60, 61, 62, 63, 64, and 65 provide examples of the DEM output from lower-resolution DJI quadcopters and the higher-resolution Bergen / D810 combination.

Table 1: DEM resolution for each site.

Location	DEM Resolution
South Warren Road	0.129 ft (39 mm)*
Gordonville Road	0.026 ft (8 mm)
Uncle Henry Road	0.017 ft (5 mm)
Beyer Road	0.023 ft (7 mm)
M-120 / Holton Road	0.03 ft (9 mm)
US-31 / White River Corridor	0.03 ft (9 mm)

* = created with 12 mp DJI Phantom 3A imagery; the rest were created with Nikon D810 36 mp imagery

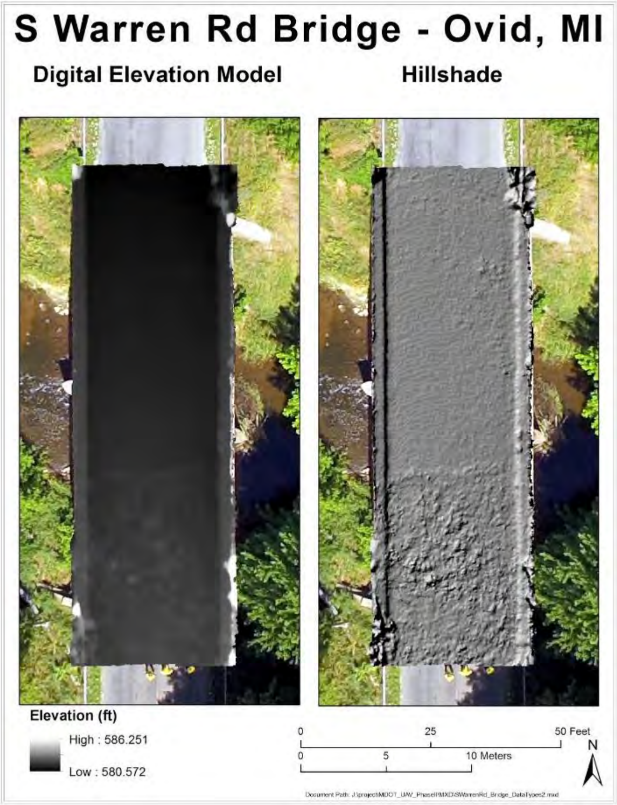

S Warren Rd Bridge - Ovid, MI
Digital Elevation Model
Hillshade
Elevation (ft)
High : 586.251
Low : 580.572
0
25
50 Feet
0
5
10 Meters
N
Document Path: J:\project\MDOT_UAV_PhaseII\MXD\SWarrenRd_Bridge_DataTypes2.mxd

Figure 60: South Warren DEM on left (with hillshade on right).

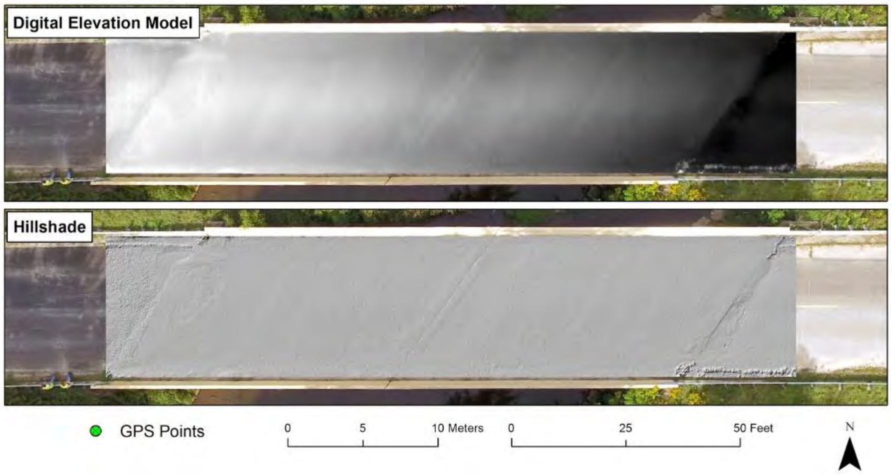


Figure 61: Gordonville Road DEM (with hillshade at bottom).

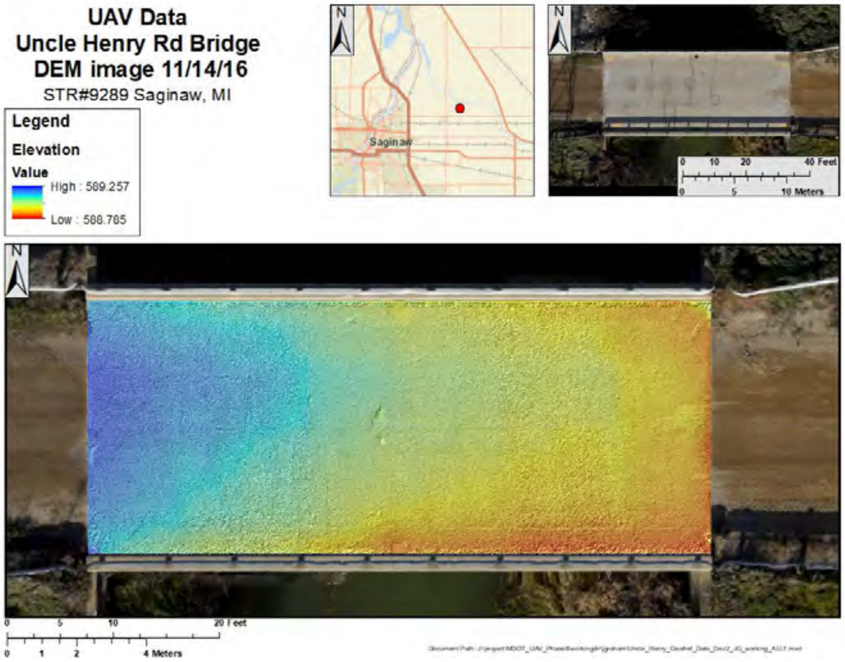


Figure 62: Uncle Henry DEM (with hillshade effect integrated with DEM).

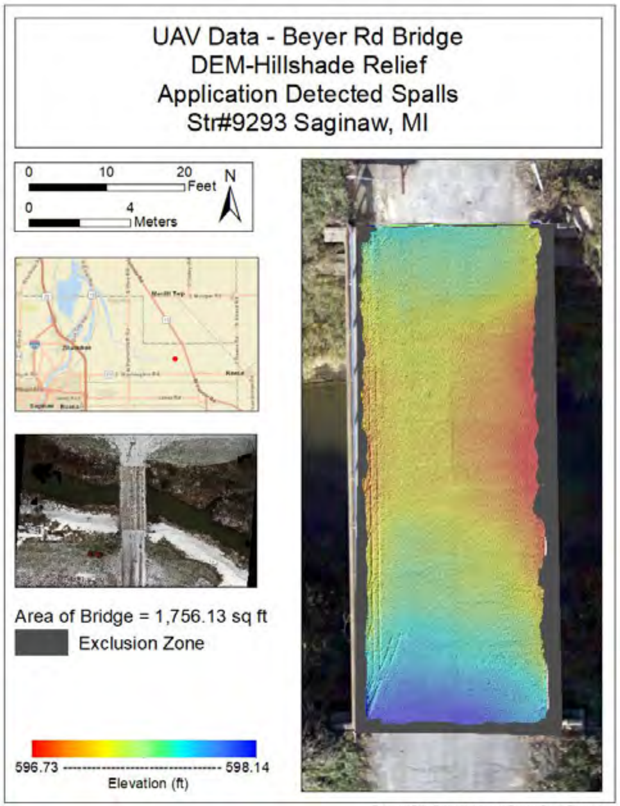

UAV Data - Beyer Rd Bridge
DEM-Hillshade Relief
Application Detected Spalls
Str#9293 Saginaw, MI
0 10 20 Feet
0 4 Meters
N
Area of Bridge = 1,756.13 sq ft
Exclusion Zone
596.73 598.14
Elevation (ft)
Document Path: D:\Users\dmbanach\Desktop\Beyer_Figs.mxd

Figure 63: Beyer Road DEM (with hillshade effect).

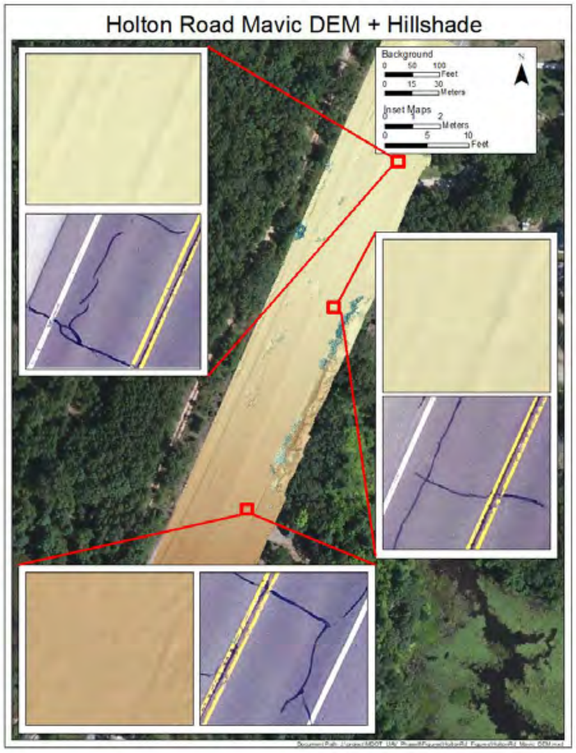


Figure 64: Holton Road DEM (with hillshade).

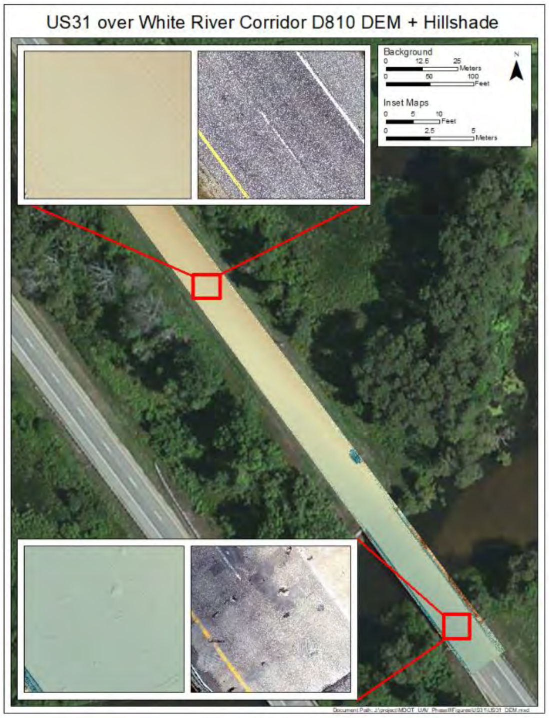


Figure 65: US-31 DEM (with hillshade).

Orthophoto

The orthophoto for each site was created by processing the high-resolution Nikon D810 optical imagery through AgiSoft PhotoScan Pro. Orthophotos could also be created with imagery from the Phantom 3A or Mavic Pro when flown, but with lower resolution. To create the orthophoto, the mesh is used to orthorectify a mosaic of the collected imagery. The orthophoto was also used for the development of the delamination detection algorithm, which compares potential delaminations to any visible features on the bridge deck. If there is a visible feature on the bridge deck in the orthophoto, it was determined that the thermal anomaly was not a delamination. Table 2 provides the resolution for each orthophoto created at each bridge/roadway location. Figures 66, 67, 68, 69, 70, and 71 provide examples of the orthophoto output from the higher-resolution Bergen / D810 combination.

Table 2: Orthophoto resolution for each site.

Location	DEM Resolution
South Warren Road	0.016 ft (4 mm)
Gordonville Road	0.026 ft (8 mm)
Uncle Henry Road	0.008 ft (2 mm)
Beyer Road	0.011 ft (3 mm)
M-120 / Holton Road	0.007 ft (2 mm)
US-31 / White River Corridor	0.009 ft (2 mm)

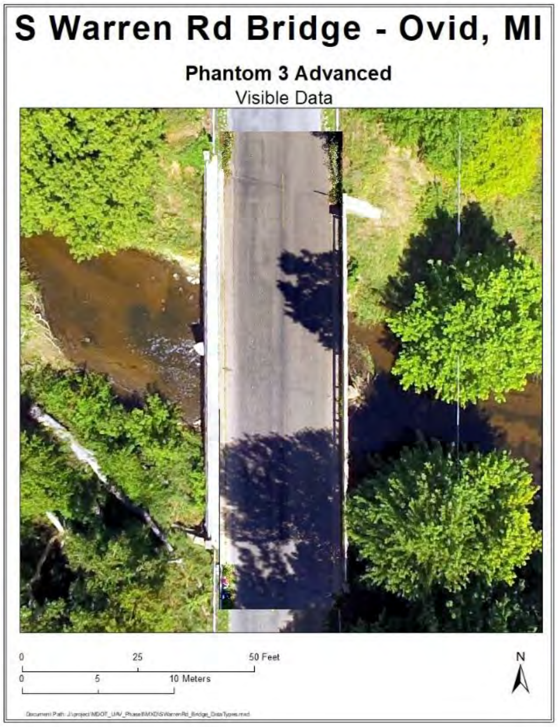

S Warren Rd Bridge - Ovid, MI
Phantom 3 Advanced
Visible Data
0
25
50 Feet
0
5
10 Meters
N
Document Path: J:\project\MDOT_UAV_PhaseII\MXD\SWarrenRd_Bridge_DataTypes.mxd

Figure 66: South Warren Road orthophoto.

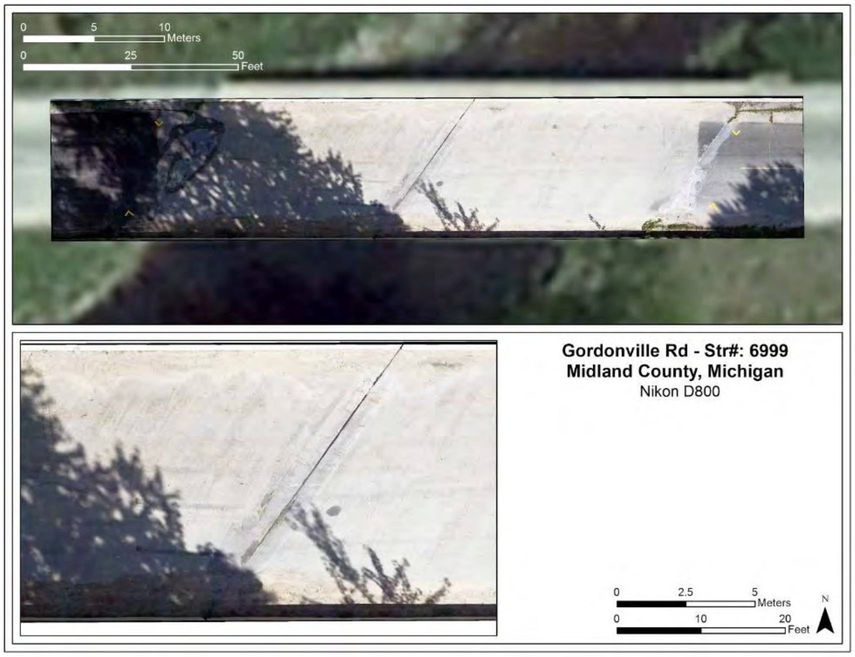


Figure 67: Gordonville Road orthophoto.

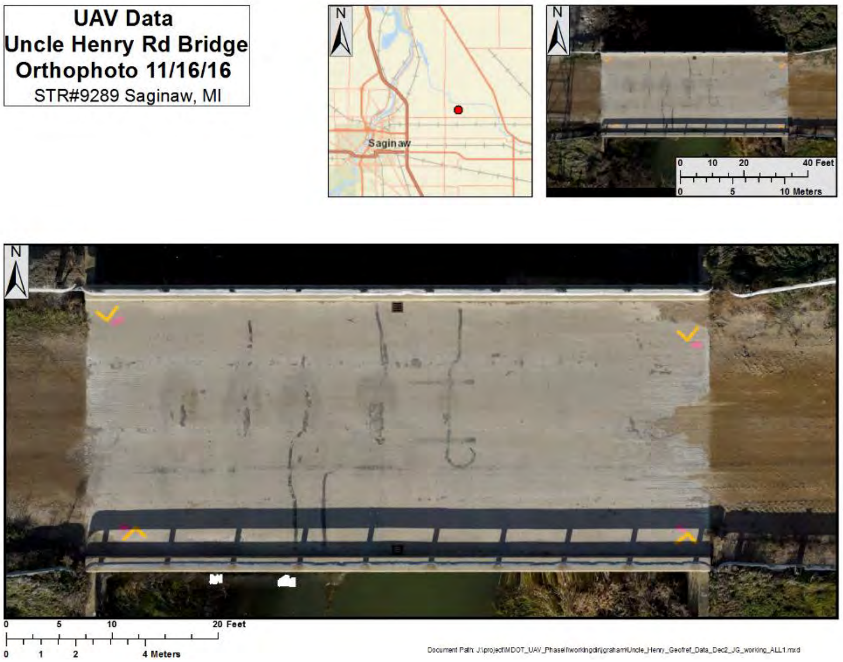


Figure 68: Uncle Henry orthophoto.

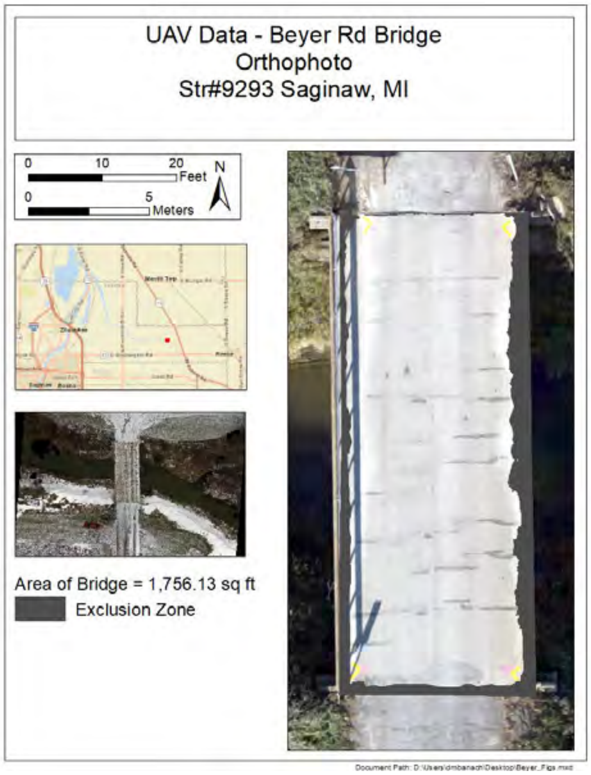

UAV Data - Beyer Rd Bridge
Orthophoto
Str#9293 Saginaw, MI
0
10
20
Feet
0
5
Meters
N
Area of Bridge = 1,756.13 sq ft
Exclusion Zone
Document Path: D:\Users\dmbanach\Desktop\Beyer_Figs.mxd

Figure 69: Beyer Road orthophoto.

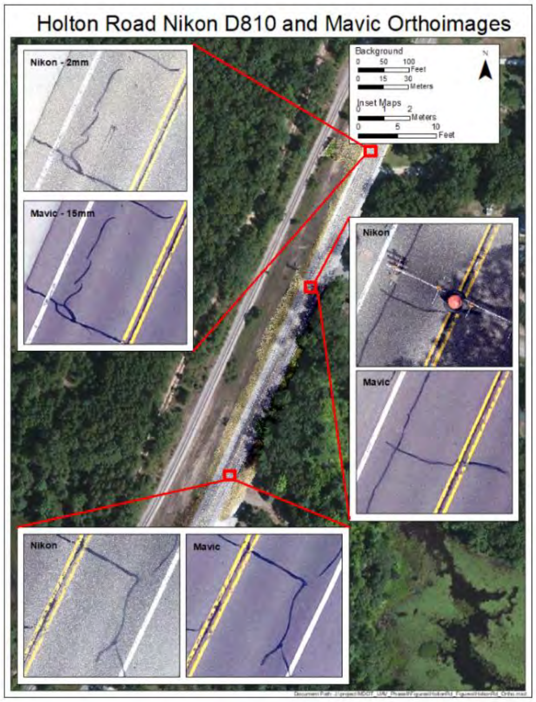

Holton Road Nikon D810 and Mavic Orthoimages
Nikon - 2mm
Mavic - 15mm
Background
0
50
100
Feet
0
15
30
Meters
N
Inset Maps
0
1
2
Meters
0
5
10
Feet
Nikon
Mavic
Nikon
Mavic
Document Path: J:\project\MDOT_UAV_PhaseII\Figures\HoltonRd_Figures\HoltonRd_Ortho.mxd

Figure 70: Holton Road orthophoto.

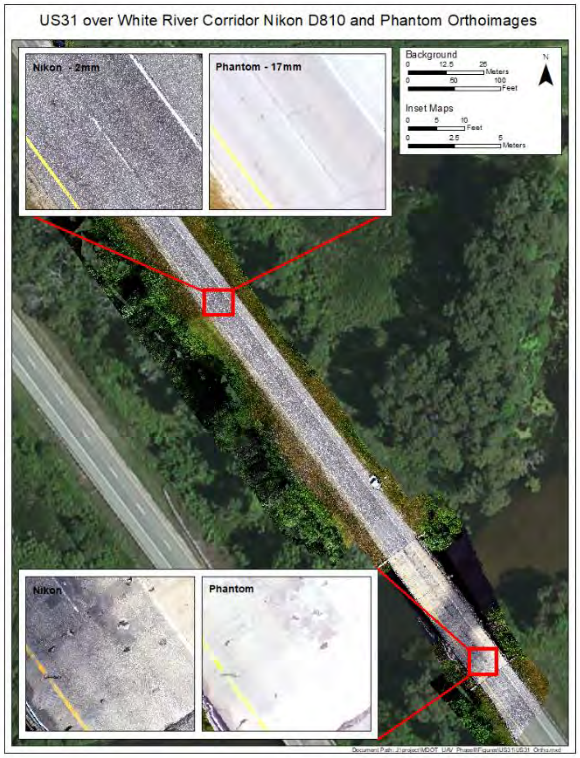


Figure 71: US-31 corridor orthophoto.

Hillshade

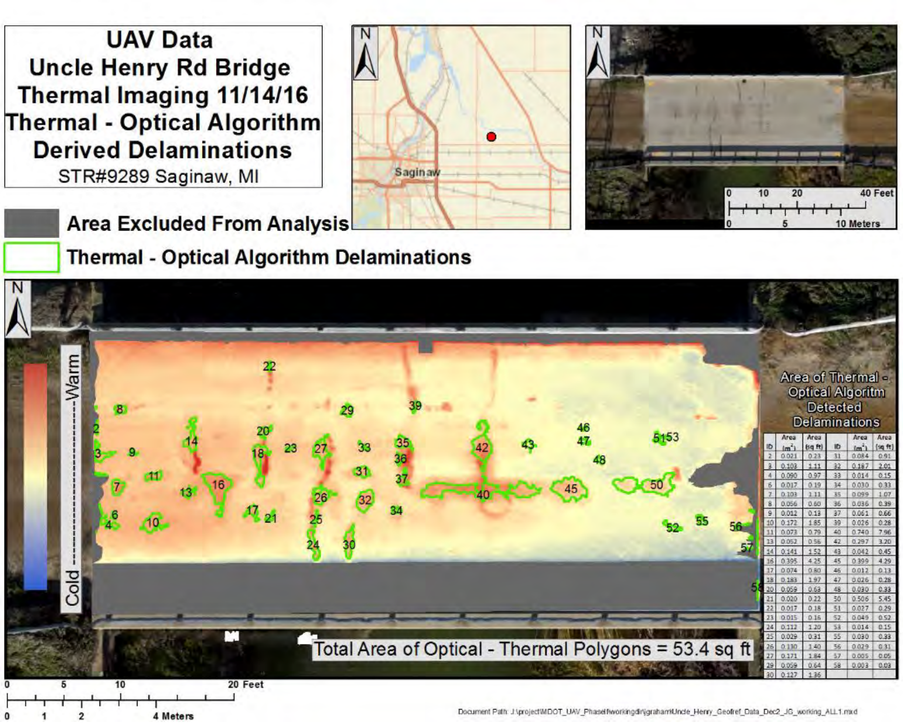
Each location's hillshade dataset is derived directly from the DEM. While the DEM contains the elevation values for each cell, it is not easy for it to show the presence of distress features. However, by processing the DEM into a hillshade within ESRI's ArcGIS Desktop software, a 3D visual representation of the road or bridge deck surface is able to be viewed showing distress features such as spalls and cracks. ESRI defines a hillshade as "a grayscale 3D representation of the surface, with the sun's relative position taken into account for shading the image". Elevation data is unable to be drawn from the hillshade and therefore the datasets should only be used for visualization purposes. Since the hillshade datasets are derived directly from the DEM, each has the same resolution as the respective DEM. Figures 60 through 65 above all have the hillshade effect that emphasizes distresses visible with 3D features (such as spalls).

Thermal Imagery

As part of Task 5A "demonstrate, develop, and implement enhanced testing of UAV-based thermal imaging for bridge deck structural integrity," thermal images collected with the FLIR sensors were georeferenced using as a georeferenced base, the high resolution orthophotos, and a set of tie points visible in both the thermal and orthophoto images. Ground resolution values (pixel size at the ground after georeferencing) varied depending on the sensor and the elevation at which it was flows, from 0.083 ft (25 mm) for low elevation UAV flights with the high resolution sensor, to 0.33 ft (101 mm) for higher elevation UAV flights with the lower resolution sensor as shown in Table 3. Figures 72, 73, and 74 provide examples of the thermal image output from the FLIR Vue Pro. The thermal imagery in Figure 74 was collected from an angle (off-nadir) and therefore is not of sufficient resolution to use automated methods to detected delaminations; differences in temperature can be seen but not detected.

Table 3: Thermal image resolution for each site.

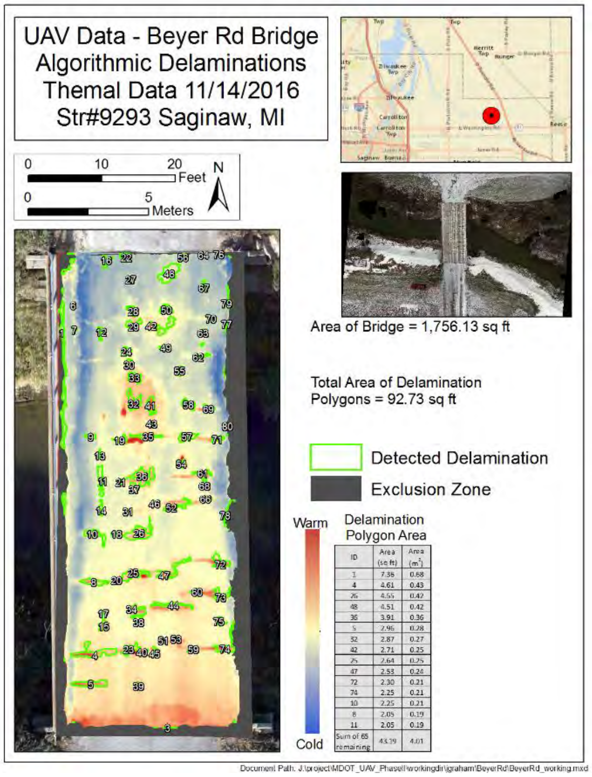
Location	DEM Resolution
Gordonville Road	0.056 ft (17 mm)
Uncle Henry Road	0.035 ft (11 mm)
Beyer Road	0.11 ft (34 mm)
M-120 / Holton Road	0.17 ft (52 mm)
US-31 / White River Corridor	0.11 ft (34 mm)
Merriman Road	0.043 ft (13 mm)



ID	Area (m²)	Area (sq ft)	ID	Area (m²)	Area (sq ft)
2	0.021	0.23	31	0.084	0.91
3	0.103	1.11	32	0.187	2.01
4	0.090	0.97	33	0.014	0.15
6	0.017	0.19	34	0.030	0.33
7	0.103	1.11	35	0.099	1.07
8	0.056	0.60	36	0.036	0.39
9	0.012	0.13	37	0.061	0.66
10	0.172	1.85	39	0.026	0.28
11	0.073	0.79	40	0.740	7.96
13	0.052	0.56	42	0.297	3.20
14	0.141	1.52	43	0.042	0.45
16	0.395	4.25	45	0.399	4.29
17	0.074	0.80	46	0.012	0.13
18	0.183	1.97	47	0.026	0.28
20	0.059	0.63	48	0.030	0.33
21	0.020	0.22	50	0.506	5.45
22	0.017	0.18	51	0.027	0.29
23	0.015	0.16	52	0.049	0.52
24	0.112	1.20	53	0.014	0.15
25	0.029	0.31	55	0.030	0.33
26	0.130	1.40	56	0.029	0.31
27	0.171	1.84	57	0.005	0.05
29	0.059	0.64	58	0.003	0.03
30	0.127	1.36			

Document Path: J:\project\MDOT_UAV_PhaseII\workingdir\jgraham\Uncle_Henry_Geofref_Data_Dec2_JG_working_ALL1.mxd

Figure 72: Uncle Henry thermal image with automatically detected delaminations.


UAV Data - Beyer Rd Bridge
Algorithmic Delaminations
Themal Data 11/14/2016
Str#9293 Saginaw, MI
0 10 20 Feet
0 5 Meters
N
Area of Bridge = 1,756.13 sq ft
Total Area of Delamination Polygons = 92.73 sq ft
Detected Delamination
Exclusion Zone
Warm
Cold
Delamination Polygon Area
Document Path: J:\project\MDOT_UAV_PhaseII\workingdir\jgraham\BeyerRd\BeyerRd_working.mxd

ID	Area (sq ft)	Area (m²)
1	7.36	0.68
4	4.61	0.43
26	4.55	0.42
48	4.51	0.42
36	3.91	0.36
5	2.96	0.28
32	2.87	0.27
42	2.71	0.25
25	2.64	0.25
47	2.53	0.24
72	2.30	0.21
74	2.25	0.21
10	2.25	0.21
8	2.05	0.19
11	2.05	0.19
Sum of 65 remaining	43.19	4.01

Figure 73: Beyer Road thermal image with automatically detected delaminations.

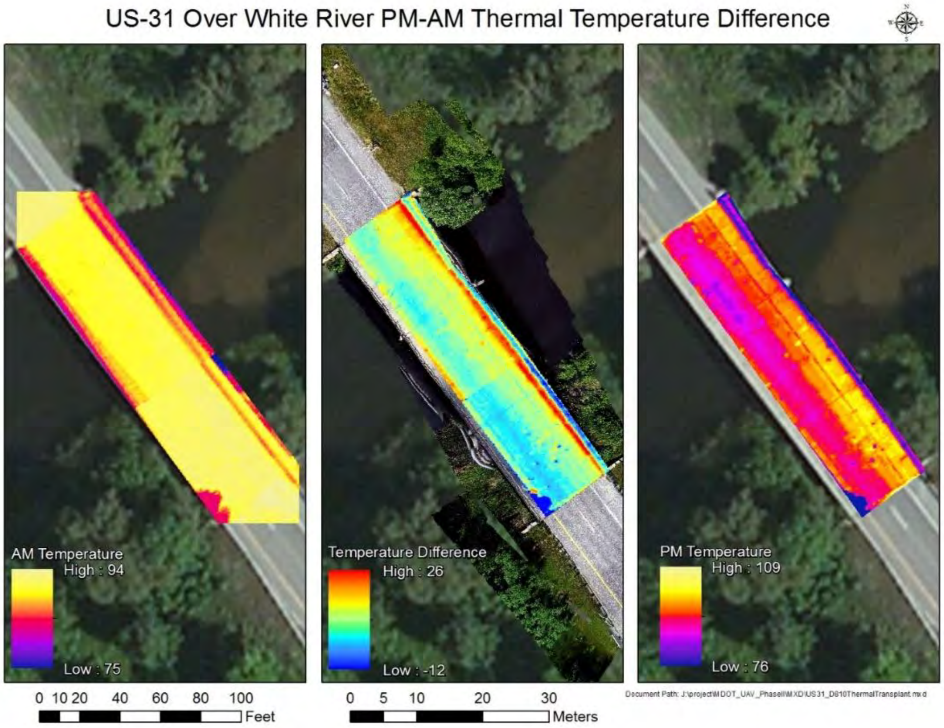

US-31 Over White River PM-AM Thermal Temperature Difference
AM Temperature
High : 94
Low : 75
Temperature Difference
High : 26
Low : -12
PM Temperature
High : 109
Low : 76
0 10 20 40 60 80 100 Feet
0 5 10 20 30 Meters
Document Path: J:\project\MDOT_UAV_PhaseII\MXD\US31_D810ThermalTransplant.mxd

Figure 74: Combined US-31 thermal images for a morning (left) and afternoon (right) data collection, with the difference between the two in the middle.

Thermal contrast for the images was usually good, but in some cases shadows from nearby trees and railing or dirt on the bridge produced uneven solar heating and artifacts in the resulting thermal images, Figures 75 shows such an example for the Gordonville Rd test site and Figure 76 for the Uncle Henry Road site. Due to the presence of trees the road surface was unevenly heated at the Gordonville Rd test site, and the resulting thermal image showed strong thermal contrasts between the sunlit and shade areas of the bridge, masking potential thermal signals from the delaminations.

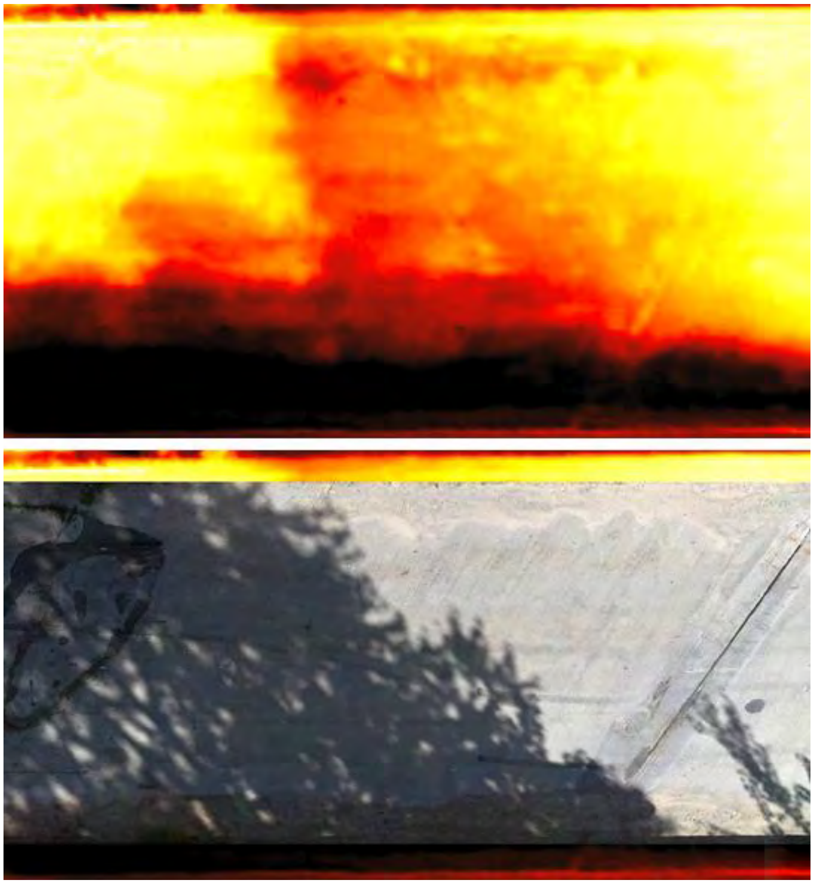


Figure 75: Comparison between the thermal (upper panel) and visible (lower panels) images at the Gordonville road test site. The presence of tree shadows, easily recognizable in the visible imagery (lower panel) mask the presence of potential thermal signatures related to delaminations (in the upper panel).

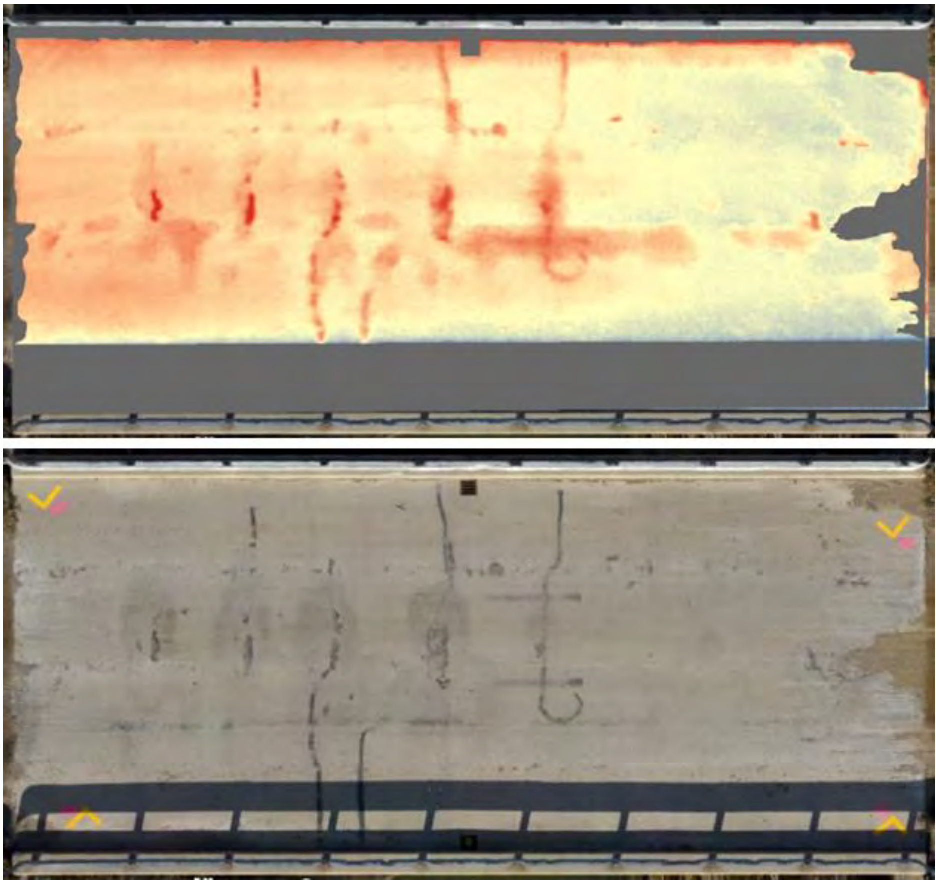


Figure 76: Comparison between the thermal (upper panel) and visible (lower panel) images at Uncle Henry Road. The presence of the rail shadow and dirt were masked out as both also show up as thermal anomalies.

Future research to improve the delamination detection tools, even in shade areas, could be pursued by more refined detection algorithm, and higher resolution (different optical / lens configuration and sensor) thermal cameras. Improved algorithms could rely on more sophisticated classification methods (e. g. supervised and unsupervised classification) rather then the relative thresholds method used in the current algorithm. Testing the detection capacity of the cameras would involve simultaneous acquisition of high resolution (e. g. FLIR Vue Pro (or Vue Pro R) 13 mm lens and 640 x 480 pixel sensor array at low flight height) and low resolution (e. g. FLIR Duo R 8 mm lens and 120 X 160 sensor array at higher flight height) of a test area affected by partial shade cover.

Spalls

The spall layer for each of the study areas was created by use of most current version of the MTRI developed spall detection algorithm (named the "Spallgorithm") python script tool (currently version 4.2), running in ESRI's Desktop ArcGIS software. The spall detection algorithm was first developed for a USDOT bridge condition remote sensing project (Vaghefi et al. 2011) and then updated for the Phase I MDOT UAV project, with additional development through the Phase II Project. The tool operates by analyzing the DEM for deviations from a smooth surface, where rapid elevation changes are likely to be a spall. As such, accuracy of the tool's output is dependent upon the quality of the DEM reconstruction, as well as significant amounts of debris, which may create noise within the DEM dataset.

The underlying processing of the tool occurs in multiple stages. The first stage of the processing is creating an "idealized" road surface, achieved by applying numerous smoothing filters across the DEM using expanding moving windows. This creates a surface layer where much of the variation as a result of spalling or other anomalies has been removed. From this smoothed layer the DEM is subtracted creating a layer where values are representative of the difference between the actual DEM and a smooth road surface, where negative values are divots and positives are bumps. The negative values are isolated in accordance with the user specified depth of which spall features should be detected. The next stage uses several functions designed to remove false positives. One function confirms that the pixels are in an area where there is a spike in the standard deviation of the DEM, another scans the roughness of the hillshade layer to exclude areas of significant roughness, and one more compares the DEM values within the spall to those directly around the spall to ensure that the difference corroborates the results of the DEM/smoothed surface subtraction. The last stage of processing occurs on the features still remaining in the analysis, and consists of appending attribute data to the detected spalls. This includes calculating geometries such as positional coordinates and area, detected the max depth within the spall, calculating volumes using triangular irregular networks from the DEM within the spall extents, and lastly converting values to meaningful units and appending attribute data associated with the spall features (Figure 77).

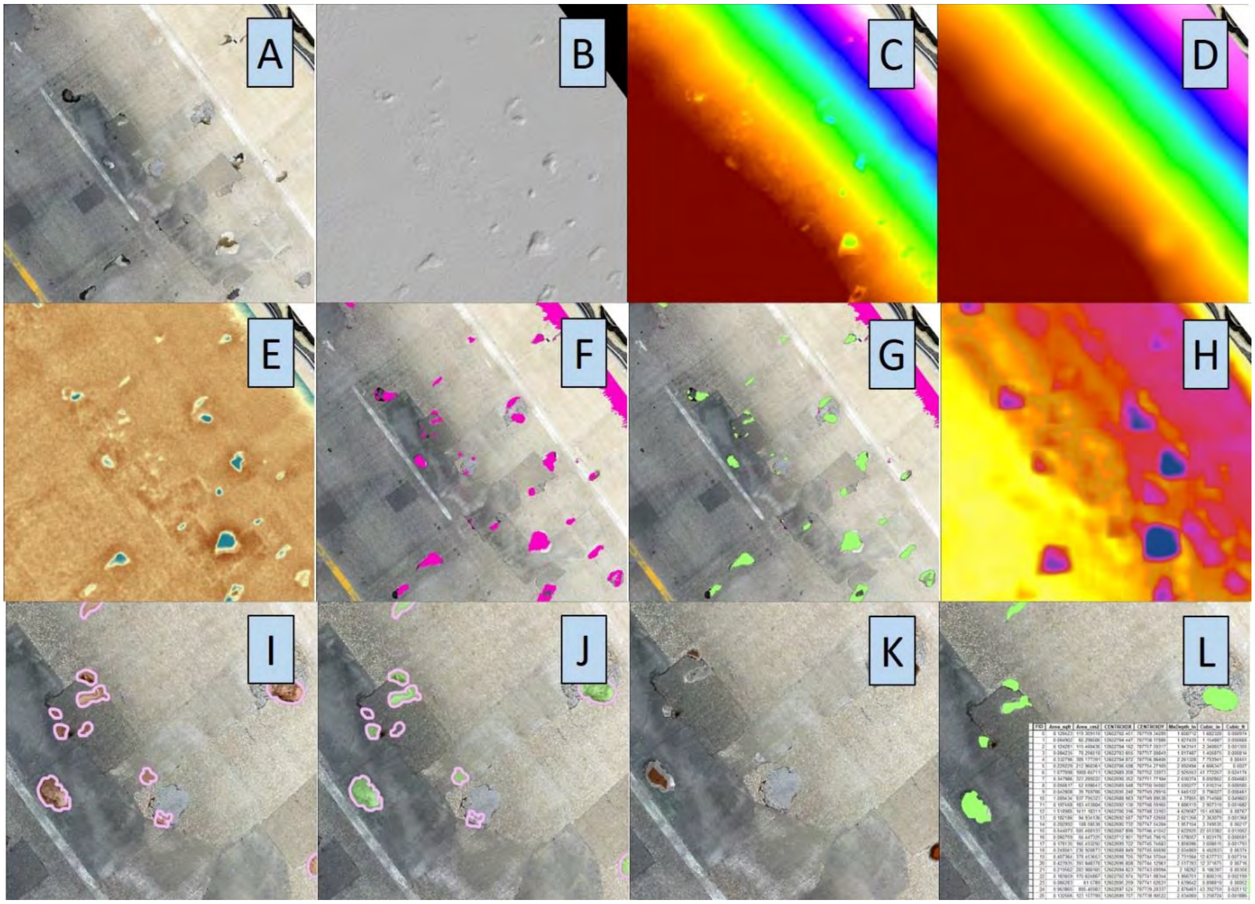


Figure 77: Example processing steps of the Spallgorithm tool. ***Included are: A) original orthophoto, B) Hillshade representation of the DEM, C) colored stretch representation of the DEM, D) the smoothed DEM surface using low pass filtering, E) the resulting layer from subtracting the DEM from the smoothed surface, F) flagged pixels meeting depth criteria, G) remaining features after filtering for size, H) inclusive standard deviation area (transparent green) overlayed on the DEM standard deviation raster, I) comparing DEM values inside of spall area to surrounding buffer, J) remaining spall features shown in green, K) triangular irregular network models used to calculate 3D geometry for volume estimation, and L) identified spalls with attribute data.***

Each spall layer contains attribute data for every detected spall feature, including: location, volume, area, and depth. The data can be indexed and sorted by any of these variables for easy identification of the most severe defects (Figures 78 & 79). Of the studied bridges the US-31 bridge over White River has the most severely spalled deck (Table 4). Spall layers are easily interpreted with the aid of the hillshade layer which aids in the visual representation of the features (Figures 80 and 81). More information regarding the usage of the Spallgorithm is located in Section 4.6 of this report, as well as in the user manual.

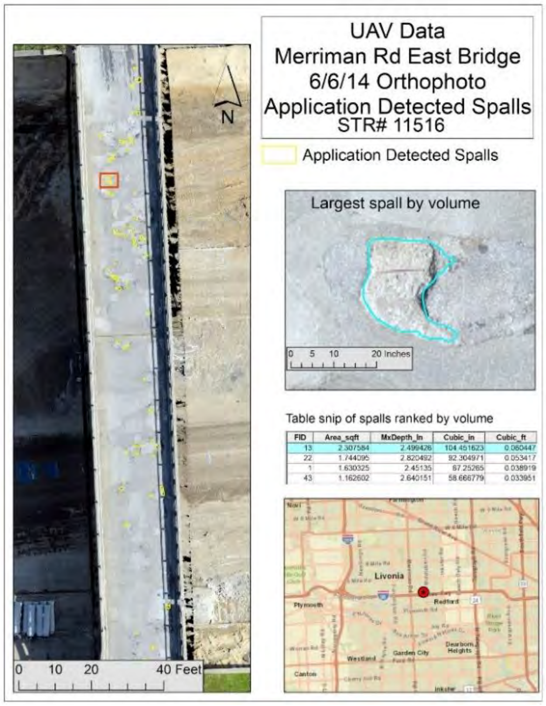
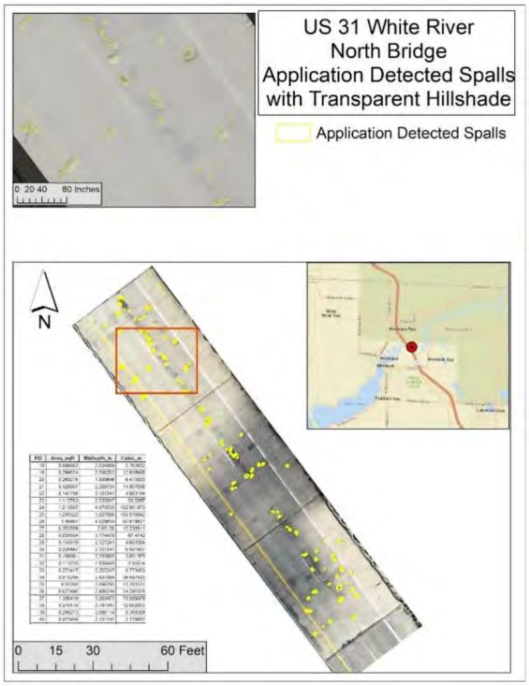


Table snip of spalls ranked by volume

FID	Area_sqft	MxDepth_in	Cubic_in	Cubic_ft
13	2.307584	2.499426	104.451623	0.060447
22	1.744095	2.820492	92.304971	0.053417
1	1.630325	2.45135	67.25265	0.038919
43	1.162602	2.640151	58.666779	0.033951

Figure 78: Spall layer of the Merriman bridge with close-up of the largest spall by volume selected from the attribute data.


US 31 White River
North Bridge
Application Detected Spalls
with Transparent Hillshade
Application Detected Spalls
0 20 40 80 Inches
N
0 15 30 60 Feet

FID	Area_sqft	MxDepth_in	Cubic_in
18	0.098663	2.034989	2.763832
19	[illegible]	[illegible]	[illegible]
20	0.260275	1.559848	[illegible]
21	[illegible]	[illegible]	[illegible]
22	0.147756	2.127241	[illegible]
23	[illegible]	[illegible]	[illegible]
24	1.213807	4.079725	102.951873
25	1.235522	[illegible]	[illegible]
26	1.98467	[illegible]	[illegible]
27	[illegible]	[illegible]	[illegible]
28	[illegible]	3.774478	67.4742
29	[illegible]	2.127241	[illegible]
30	0.228467	2.127241	[illegible]
31	[illegible]	[illegible]	[illegible]
32	[illegible]	1.535445	[illegible]
33	0.273417	3.257247	[illegible]
34	[illegible]	[illegible]	[illegible]
35	[illegible]	[illegible]	[illegible]
36	[illegible]	[illegible]	[illegible]
37	1.386419	[illegible]	[illegible]
38	[illegible]	[illegible]	[illegible]
39	0.296273	[illegible]	[illegible]
40	[illegible]	[illegible]	[illegible]

Figure 79: Spall layer of the US31 bridge with close-up of semi-transparent hillshade layer and example of attribute data.

Table 4: Summary of spalling from the analyzed bridge decks.

Bridge	Number of Spalls	Spall Area (sq ft)	Spall Volume (cu in)	Deepest Spall (in)	Largest Area (sq ft)	Largest Volume (cu in)
US-31	108	54.39	3,150.95	5.3	2.24	219.68
Uncle Henry	3	0.84	31.52	2.4	0.50	16.98
Merriman	60	28.04	953.98	2.8	2.30	104.45
Beyer	1	1.21	48.38	-	-	-
Gordonville	0	No spalling detected				

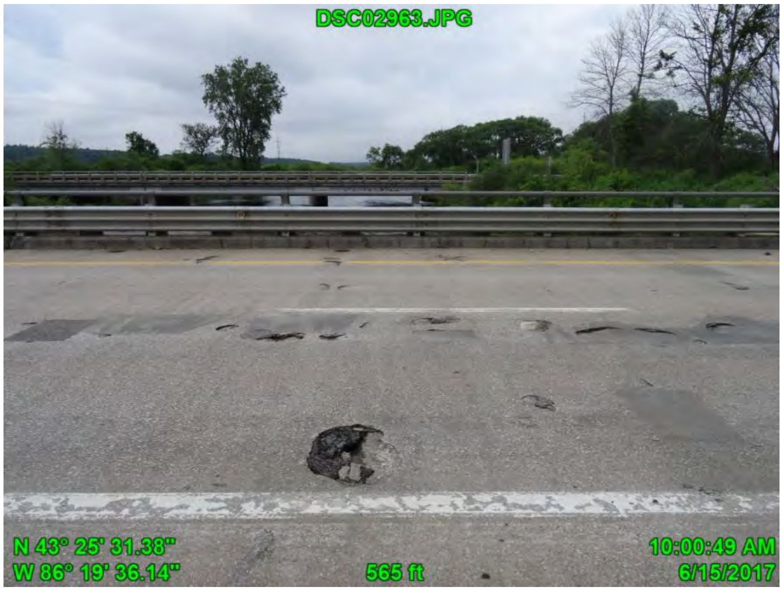
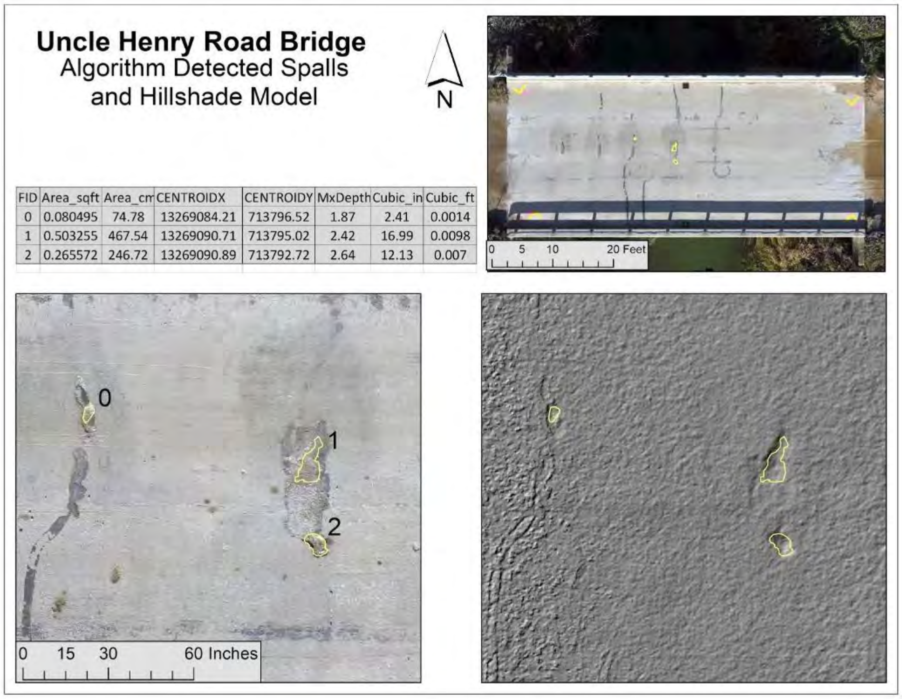


Figure 80: Field photograph of the zoomed in area in Figure 79.



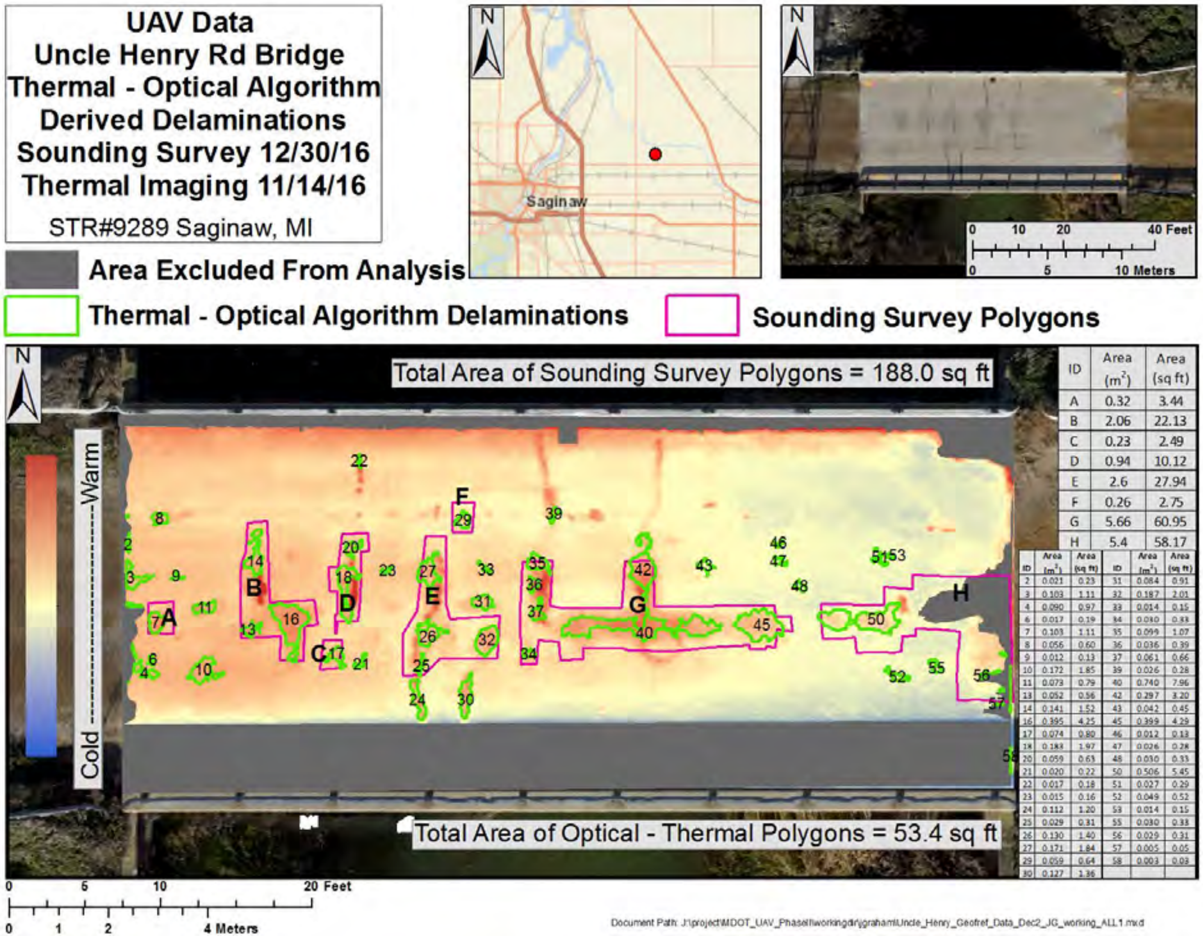
FID	Area_sqft	Area_cm	CENTROIDX	CENTROIDY	MxDepth	Cubic_in	Cubic_ft
0	0.080495	74.78	13269084.21	713796.52	1.87	2.41	0.0014
1	0.503255	467.54	13269090.71	713795.02	2.42	16.99	0.0098
2	0.265572	246.72	13269090.89	713792.72	2.64	12.13	0.007

Figure 81: Uncle Henry detected spalls and associated attribute data shown with orthophoto and hillshade layers.

The updated spall detection algorithm provides an example of how UAV-collected data can be efficiently and accurately analyzed to produce quantitative results that can be integrated into MDOT workflows. The Spallgorithm software tool for Desktop ArcGIS can be used by MDOT in its current form for providing rapid and repeatable measures of spalling for bridge decks. In a potential future phase, it can be made available to bridge inspections and managers for day-to-day use with appropriate training, as needed.

Delamination Detection from Thermal Imagery

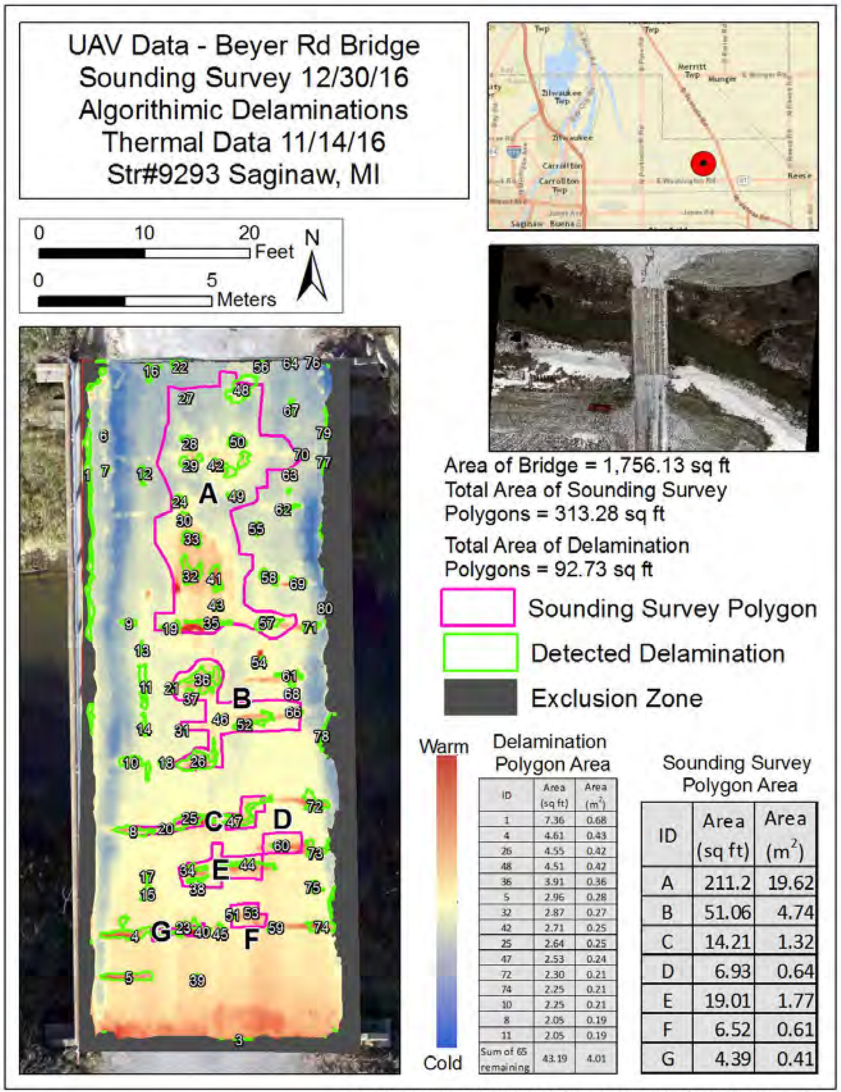
Delamination detection from thermal imagery was successfully implemented, using the UAV based thermal imagery. An algorithm, initially implemented in MATLAB, and then translated to ArcPy (for use as a Python toolbox for Desktop ArcGIS), was developed to detect, isolate and map the thermal anomalies that could be caused by concrete delaminations. The algorithm has been named the "Thermal Delamination Detection Algorithm" or TDDA, and an user manual is available as Appendix, Section 9.4. The algorithm takes both the thermal and visible images (georeferenced and orthorectified) and flags those areas (pixels in the digital image) that show anomalous high temperatures but no visible changes compared to the surroundings (from the visible imagery), which could be related to changes in the surface emissivity rather temperature differences, as would be expected from real delaminations. Figures 82 and 83 shows examples of the successful application of the method to mapping delaminations at both the Uncle Henry and Beyer road test sites, respectively. The thermal datasets were collected in November of 2016 with a rise of approximately 10 F in temperature between the coldest point in the early morning and data collection near mid-day. Visible and thermal images show the areas of higher temperature (potential delaminations) and surface emissivity variations (e. g. concrete patching, spalling, stains, etc.). Areas with likely unusable thermal data, such as areas shadowed by the bridge railing at the edge and from debris also near the bridge are marked as analysis exclusion areas. The algorithm is able to isolate the thermal anomalies that are not associated to variations in surface emissivity (as determined from the visible imagery). These areas compare well with areas mapped as delaminations in an MDOT bridge survey from December 2016.


UAV Data
Uncle Henry Rd Bridge
Thermal - Optical Algorithm
Derived Delaminations
Sounding Survey 12/30/16
Thermal Imaging 11/14/16
STR#9289 Saginaw, MI
Area Excluded From Analysis
Thermal - Optical Algorithm Delaminations
Sounding Survey Polygons
N
Saginaw
0 10 20 40 Feet
0 5 10 Meters
Total Area of Sounding Survey Polygons = 188.0 sq ft
Total Area of Optical - Thermal Polygons = 53.4 sq ft
Warm
Cold
A B C D E F G H
0 5 10 20 Feet
0 1 2 4 Meters
Document Path: J:\project\MDOT_UAV_PhaseII\working\dir\jgraham\Uncle_Henry_Geotref_Data_Dec2_JG_working_ALL1.mxd

ID	Area (m^2)	Area (sq ft)
A	0.32	3.44
B	2.06	22.13
C	0.23	2.49
D	0.94	10.12
E	2.6	27.94
F	0.26	2.75
G	5.66	60.95
H	5.4	58.17

ID	Area (m^2)	Area (sq ft)	ID	Area (m^2)	Area (sq ft)
2	0.021	0.23	31	0.084	0.91
3	0.103	1.11	32	0.187	2.01
4	0.090	0.97	33	0.014	0.15
6	0.017	0.19	34	0.030	0.33
7	0.103	1.11	35	0.099	1.07
8	0.056	0.60	36	0.036	0.39
9	0.012	0.13	37	0.061	0.66
10	0.172	1.85	39	0.026	0.28
11	0.073	0.79	40	0.740	7.96
13	0.052	0.56	42	0.297	3.20
14	0.141	1.52	43	0.042	0.45
16	0.395	4.25	45	0.399	4.29
17	0.074	0.80	46	0.012	0.13
18	0.183	1.97	47	0.026	0.28
20	0.059	0.63	48	0.030	0.33
21	0.020	0.22	50	0.506	5.45
22	0.017	0.18	51	0.027	0.29
23	0.015	0.16	52	0.049	0.52
24	0.112	1.20	53	0.014	0.15
25	0.029	0.31	55	0.030	0.33
26	0.130	1.40	56	0.029	0.31
27	0.171	1.84	57	0.005	0.05
29	0.059	0.64	58	0.003	0.03
30	0.127	1.36			

Figure 82: Example of the successful application of the TDDA thermal delamination detection algorithm. The TDDA-detected likely delaminations are outlined by green polygons (numbered 1-58) based on the November 2016 UAV thermal data and the larger manual sounding delamination areas are in pink (lettered A-H) as marked during a bridge survey by MDOT in December of 2016.



UAV Data - Beyer Rd Bridge
Sounding Survey 12/30/16
Algorithimic Delaminations
Thermal Data 11/14/16
Str#9293 Saginaw, MI

Area of Bridge = 1,756.13 sq ft
Total Area of Sounding Survey
Polygons = 313.28 sq ft
Total Area of Delamination
Polygons = 92.73 sq ft

🟪 Sounding Survey Polygon
🟩 Detected Delamination
⬛ Exclusion Zone

Delamination
Polygon Area

ID	Area (sq ft)	Area (m^2)
1	7.36	0.68
4	4.61	0.43
26	4.55	0.42
48	4.51	0.42
36	3.91	0.36
5	2.96	0.28
32	2.87	0.27
42	2.71	0.25
25	2.64	0.25
47	2.53	0.24
72	2.30	0.21
74	2.25	0.21
10	2.25	0.21
8	2.05	0.19
11	2.05	0.19
Sum of 65 remaining	43.19	4.01

Sounding Survey
Polygon Area

ID	Area (sq ft)	Area (m^2)
A	211.2	19.62
B	51.06	4.74
C	14.21	1.32
D	6.93	0.64
E	19.01	1.77
F	6.52	0.61
G	4.39	0.41

Document Path: J:\project\MDOT_UAV_PhaseII\workingdir\jgraham\BeyerRd\BeyerRd_working.mxd

Figure 83: Example of the successful application of the TDDA tool. The green polygons show likely delamination areas as detected by the TDDA, and the pink polygons are delamination marked by a MDOT bridge survey a month after the UAV thermal data collection.

The delamination detection algorithm could be further improved by using more sophisticated methods to classify pixels in the thermal and visible images as potential delaminations. Supervised classification algorithms (e. g. Nearest neighbor, support vector machines, etc.) could be tested with the thermal and three visible bands, based on a training dataset, e. g. from a bridge that has been surveyed and for which delamination data area available. Clustering and unsupervised classification algorithms could also be tested, as an alternative to supervised classification methods, which require training data. Isodata and k-mean clustering using the visible and thermal bands could be tested on different datasets, and for these case, the bridge inspection data on delaminations could be used as a validation data set. The main idea of improving the detection algorithm is for those cases where our current algorithm performs poorly, e. g. when the surface is unevenly heated due to shadowing from trees or other objects.

Point Clouds (Structure from Motion and Lidar)

The point clouds derived from SSI's mobile lidar system of 2015 (since upgraded) and SfM from the UAV-collected D810 imagery of the US-31 bridge over White River were compared (Figure 84). The point cloud densities were evaluated for density ratings using the National Cooperative Highway Research Program (NCHRP) Mobile LiDAR density rating (see NCHRP Report 748 - Olsen 2013). The SfM derived point cloud was significantly denser than the mobile lidar data set, demonstrating an advantage to SfM-derived 3D data. However, SSI's mobile lidar system is easier to deploy for longer-distance data collection since it is attached to a vehicle and does not have to deal with airspace issues or UAV battery life limits. Point cloud densities for the mobile lidar was 27.7 pts/ft^2 (298.2 pts/m^2), which is a NCHRP Report 748 density rating of "fine", while the SfM point cloud was 606.2 pts/ft^2 (6,527.1 pts/m^2) which is a also a density rating of "fine". Figure 85 shows a zoomed comparison of the two point clouds, showing the higher density achieved with UAV-collected SfM photogrammetry 3D process. SSI's newer mobile lidar system is capable of 2.1 million points per second at 250 lines per second while the older mobile lidar system shown here collected data at 300 points per second at 100 lines per second.

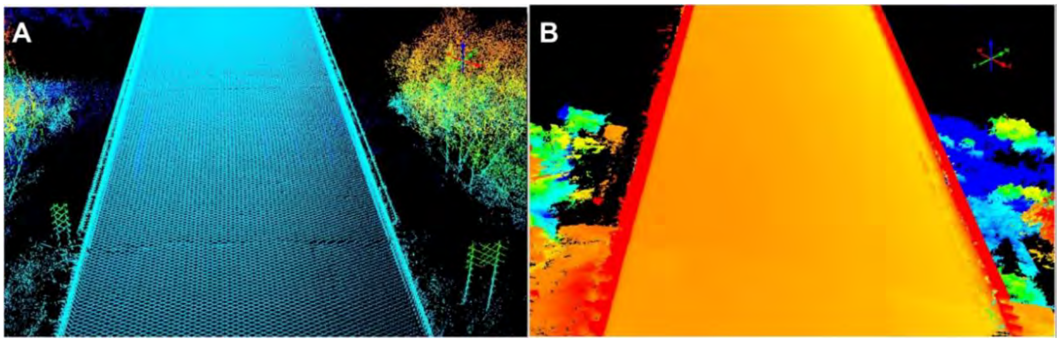


Figure 84: Lidar (left) and SfM (right) point clouds for the US-31 bridge.

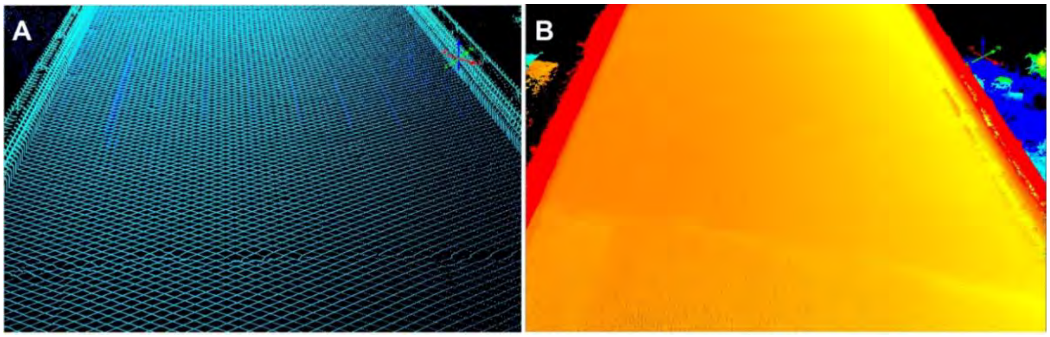


Figure 85: Lidar (left) and SfM (right) point clouds for the US-31 bridge with zoomed-in versions at bottom.

4.2 UAV Traffic Monitoring

As part of Task 4 “demonstrate, develop, and implement uses of data collection from UAV(s) and sensors for operations, maintenance, design, and asset management” and Task 1B “provide data collection from UAVs to the MDOT’s DUAP project that meets the quality, low latency delivery and data format requirements,” a follow-up to the Phase I traffic monitoring demonstrating was completed to move the technology towards implementation in traffic operations. The content of this section is as follows: Section 4.2.1 is an overview of a semi-automated UAV traffic monitoring tool” was developed, also building from a concurrent National Science Foundation project led by K.Zhang and C.Brooks focused on UAV-collected traffic video analysis. Section 4.2.2 introduces the steps to use the tool. Section 4.2.3 introduces methods and algorithms. Section 4.2.4 gives a brief expectation of the future work. A more detailed version of the traffic monitoring work is included in Appendix 9.5 and also described in a recent TRB conference by Zhao et al. (2018).

4.2.1 Overview of the Semi-automated UAV Traffic Monitoring Tool

The semi-automated UAV traffic monitoring tool was developed using C/C++ and wxWidget libraries for a Graphical User Interface (GUI), and Open Computer Vision (OpenCV) libraries for non-parametric algorithms to detect vehicles and roads in a UAV video with minimal manual annotations. By using this tool, users can add mark information (for geo-registration due to remote sensing), label an object (e.g. individual vehicle driving states and road characteristics) of interest, and conduct traffic data analysis useful for traffic operations management (e.g. traffic studies such as volume and speed studies etc.) on a semi-automated basis. This tool also outputs frame-by-frame estimated annotations of individual vehicle driving states. The automatic frame-by-frame estimated annotations provides a labeled dataset for training efficient machine learning algorithms to automatic vehicle detection and tracking.

The GUI includes two parts, one is an Annotation and Control Panel for presenting the annotation information and controlling the video; the other one is a Video Display Window for annotation operations in a video (e.g. selecting the tracked object, adding projection points, labeling the objects of interest) and animation traffic flow dynamics (e.g. volume, speed, and individual vehicle trajectory). Figure 86 shows the GUI of the semi-automated traffic monitoring tool.

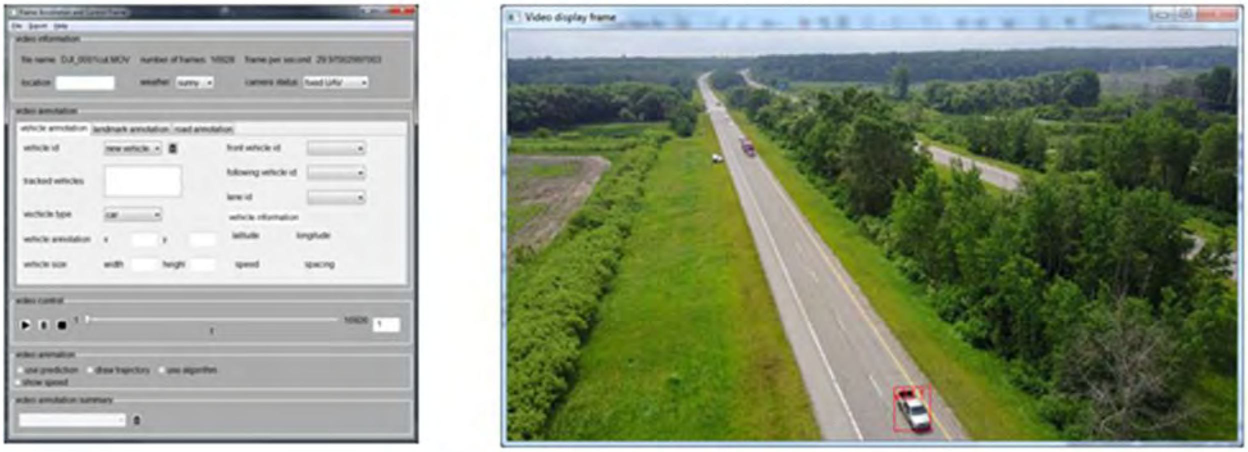


Figure 86: GUI of the semi-automated UAV traffic monitoring tool.

4.2.2 Procedure of Using the Semi-automated UAV Traffic Monitoring Tool

There are six steps to use the semi-automated UAV traffic monitoring tool for traffic data analysis: (1) landmark annotation, (2) road annotation, (3) vehicle annotation, (4) frame-by-frame annotation estimation, (5) estimated annotations output, and (6) traffic analysis and animation.

1. Landmark annotation aims to get the world coordinate of a point on a frame, which is used to calculate actual speed and distance in the real world from the remote sensing data - UAV video frames.
2. Road annotation is the second procedure for collecting traffic data. Users only need to annotate the road on the first frame, the software will generate the road annotations on the following frames automatically. The tool also provides a manual way to annotate a road or revise a road annotation.
3. Vehicle annotation aims to provide a semi-automatic way for users to annotate vehicles.
4. Frame-by-frame annotation estimation (i.e. landmark, road, and vehicle) uses manual annotations of landmark, road, and vehicles from a sample of frames in a video to automatically estimate all the annotations of landmark, road, and vehicles frame-by-frame in the video.
5. Estimated annotation outputs include three files: (1) frame-by-frame estimated annotations of landmark, road, and vehicles, (2) vehicle trajectories, and (3) traffic variables. These files can be used for different purposes such as visualization (including animation on the screen and as time-space diagram) and analysis of traffic flow statistics.
6. In traffic analysis and animation step, the data we collect in the previous steps will be presented on the screen directly. Users can use the animation to watch the results such as vehicle driving trajectory and speed, traffic volumes, number of vehicles on the road.

4.2.3 Methods and Algorithms of the Automatic Frame-By-Frame Annotation Estimation and Traffic Data Analysis

This section presents methods and algorithms used in the automatic frame-by-frame annotation estimation step in the previous Section 4.2.2. The landmark, road and vehicle manual annotations are used for landmark annotation estimation frame-by-frame annotation estimation. A homogeneous perspective projection method is used in for landmark annotation. For road annotation estimation, the k-means algorithm (Hartigan and Wong, 1979) and Hough transform algorithm (Duda and Hart, 1972) are combined to find a road segment. For vehicle annotation estimation, both linear interpolation and a mean shift algorithm (Comaniciu and Meer, 2002) based on frame-by-frame difference image are implemented. For traffic analysis, we present the time-space diagram, calculate the density, flow and estimate the speed using the coordinate information obtained in landmark annotation step.

4.2.3.1 Homogeneous Perspective Projection Method Using Landmark Annotation

The homogeneous perspective projection method is used to build the relationship between the coordinates of control points on a frame in the video and the real world coordinates (e.g. latitude and longitude). The input of homogeneous perspective projection method is the pairs of control points and its corresponding real world coordinates. The algorithm will output a transformation matrix indicating the relationship between coordinates on image and real world coordinates.

4.2.3.2 Road Identification Method Using Road Annotation

A clustering algorithm is used for road identification and the Hough transform transformation algorithm is used to detect the edges of a road. The input of the algorithm is the annotation of road in the previous estimated annotated frame and the output is the estimated annotation on current frame. There are three steps in the process: (1) choice of a color space, (2) clustering pixels using the k-means algorithm, (3) detecting edges of road using the Hough transform algorithm.

By applying the k-means algorithm, the pixels are clustered into two clusters according to the pixel values: one contains road surface pixels and the other contains non-road surface pixels. Then the Hough transform algorithm for detecting lines is applied to find the edges of a road after some morphological processing of the clustering result.

4.2.3.3 Vehicle Detection Method

Vehicle detection is based on frame-by-frame difference and the mean shift algorithm. The input of vehicle detection is the manual annotations, and output is the annotations on frames between two annotated frames. There are two steps in the vehicle detection method: (1) frame-by-frame difference image processing, and (2) the mean shift algorithm for vehicle detection.

4.2.3.4 Traffic Data Analysis

The software itself also collects data and displays the results on the screen. The software can analyze traffic variables useful for traffic operations such as the vehicle speed, cumulative number of vehicles entering the road, cumulative number of vehicles exiting the road, traffic in-flow rate, traffic out-flow rate and traffic density.

4.2.3.4.1 Traffic Volume and Flow Rate

Traffic volume is the number of vehicles passing by a specific point during a specific time period. Flow rate is the number of vehicles passing by a specific point during an hour. Typically, the time period used to calculate the flow rate is 1 minute, 5 minutes, 10 minutes, and 15 minutes. Since our video are almost less than 10 minute, the flow rate is only calculated at the end of the video.

4.2.3.4.2 Density

Density is defined as the number of vehicles per mile. Other densities of interest to MDOT could also be calculated.

4.2.3.4.3 Space Mean Speed

The definition of space-mean speed is the ratio between total distance of vehicle and total time of vehicles passing the road. There are two ways to calculate the space mean speed in our software.

1. The space mean speed can be calculated by total distance dividing total travel time.
2. The space mean speed is often calculated by arithmetic instantaneous speed of all vehicles over the road on each frame.

In animation step, the space mean speed is calculated by average instantaneous speed of on-road vehicles. Space mean speed can also be calculated according to the data in traffic variable file.

4.2.3.4.4 Time-Space Trajectory

Time-space trajectory shows the vehicle trajectory through time as it moves on a road. Some traffic data can be derived from time-space trajectory such as vehicle speed, average density between two points and average flow between two time points. It is also a direct way to reflect some special traffic phenomenon, for example, shock waves. Figure 87 is an example of time-space trajectory.

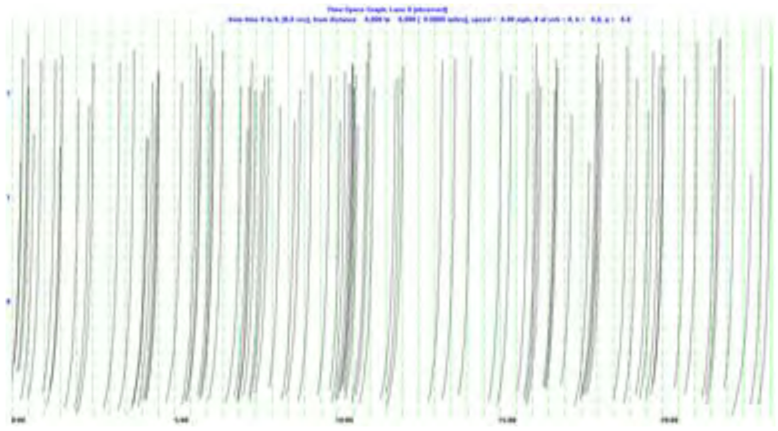


Figure 87: An Example of Time-space Trajectory.

4.2.4 Potential Future Work

Potential future work would include building an automatic UAV video-based traffic monitoring tool that can be used by MDOT traffic engineers to collect traffic data using UAV for daily operations including traffic studies (e.g. volume studies, speed and travel time studies etc.), capacity analysis, traffic simulation calibration and validation, traffic safety analysis, and Intelligent Transportation Systems operations, etc. An automatic UAV video-based Traffic monitoring tools requires a large number of annotation training samples from real world data for offline deep learning model training.

To create an automatic UAV video traffic data analysis tool, we would need to do the following tasks:

1. Collect more UAV videos for reoccurring and non-reoccurrent traffic conditions at different scenarios (such as traffic accidents and work zone etc.) and different types of roads, as well as weather conditions.
2. Use the semi-automated traffic monitoring tool in this project to annotate more data samples, especially for some vehicle overlapping cases under stop-and-go traffic conditions. This will create a database for UAV vehicle and road annotation data samples.
3. Develop a fully automatic vehicle detection and tracking algorithm using deep learning and conduct model training.
4. Develop an advanced projection (or geo-registration) algorithm (due to the remote sensing) to provide precise vehicle positions to estimate vehicle speed, size, and distance gap etc., and road information.
5. Validate and calibrate the fully automatic vehicle detection and tracking algorithm for traffic analysis.
6. Consideration of developing an on-line real-time fully automatic UAV video traffic monitoring algorithm that could be developed.
7. Demonstrate deployment of the automatic UAV traffic data analysis tool at a MDOT transportation operations center.

4.3 FAA Permission

As part of Task 8 “Secure a Federal Aviation Administration Certificate of Authorization,” MTRI staff had applied for and received FAA Section 333 Exemption approval before the start of the project that allowed the project team to fly over 1,100 different UAV platforms on March 25, 2016. To achieve this, MTRI worked with Gowdy Brothers Aerospace (located in Shakopee, Minnesota), who assisted with filing the legal paperwork. As part of the approval, certain conditions and limitations were required to be followed such as:

- All UAVs must weigh less than 55 lbs.
- All flight require a pilot in command (PIC) and visual observer (VO).
- Flights can fly up to a maximum of 400 feet AGL and must remain at least 500 feet below the cloud base.
- The UAV must remain with visual line of sight.
- All flights must include a pre-flight inspection.
- Monthly reporting required overviewing any flights that took place, name of the operator, UAV model, flight locations, number of flights, damage (if any), malfunctions (if any), and the number and duration of lost link events.

In August 2016, the FAA implemented the Part 107 - Small Unmanned Aircraft System regulations (https://www.faa.gov/uas/getting_started/part_107/). Anyone can obtain a Part 107 unmanned pilot certificate by passing an aeronautical knowledge exam. Four members of the project team have obtained the certificate: Colin Brooks, Rick Dobson, Sam Aden, and Nicholas Marion, and collected UAV data for the project. Some of the regulations listed as part of Section 333 remain under the Part 107 rules. Certain conditions and limitations must be followed under Part 107, including:

- All UAVs must weigh less than 55 lbs.
- The UAV must remain with visual line of sight.
- Operations can take place during daylight or twilight (30 minutes before official sunrise and 30 minutes after official sunset).
- Flights can fly up to a maximum of 400 feet AGL.
- Flight may not occur over anyone not directly participating in the operation or from a moving vehicle.
- Operations in airspace classes B, C, D, and E need proper air traffic control permission through the FAA waiver procedure.
- Flights that do not conform to the proper regulations can still be flown as long as a waiver has been submitted and approved (www.faa.gov/uas/request_waiver/); examples including flying at night or flying closer to airports than standard regulations. For a different project, MTRI staff obtained a waiver to fly closer to the Houghton-Hancock Airport than standard FAA rules for towered airports (i.e., closer than 5 miles). This demonstrates that waivers could be obtained for future MDOT projects.

4.4 SSI Data Collection and Storage System

As part of Task 1A "collect data from the UAV platform using sensing technology in near-time (as real-time as can be achieved) demonstrating, developing, and implementing storage capabilities of large amounts, usage, and application development of data," and Task 1B "provide data collection from UAVs to the MDOT DUAP project that meets the quality, low latency delivery and data format requirements," several different types of data transfer sites/protocols were investigated and tested during the course of the project. The goal was to find a solution to quickly upload data, allow quick access to the data, and distribute the data to multiple users including MDOT's DUAP database of information related to connected vehicles, with the idea that DUAP applications could eventually be written that integrated the UAV data results. Given the size of the data, the data transfer site had to have adequate bandwidth and storage to handle large datasets being generated.

A demo license of ERDAS Apollo software was obtained to test its functionality. The software allowed the data stored in its catalog to be visualized, metadata to be viewed, and data disseminated to other users. One of the strengths of Apollo was its ability to allow visualization of the stored data along with the associated metadata for each data set. The cost for a license was not within the budget so this solution was no longer pursued after the demo license expired, but did demonstrate feasibility of this commercial software.

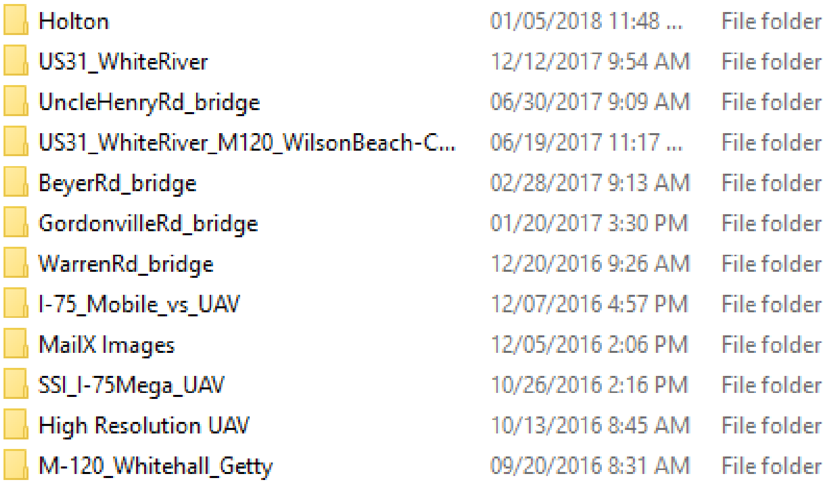
The final solution explored was SSI's cloud storage software setup. This option allowed the team members to collaborate and transfer files easily. It also had a provision that allowed DUAP to directly harvest the data from the cloud. For this project, SSI already had the data stored in their cloud based environment. Therefore, a segment of the point cloud data was re-purposed for this UAV project. The advantages of this solution include a unique credential (ID/password) for each user, a full audit trail, and data backup mechanism stored within the secure cloud. Administrators can quickly track down any changes to the files or the cloud which were performed by the users. Security can be setup as group security, individual credential-based security, folder-based security, or all the way down to file level security. The file storage system is completely scalable so that data can reside on many machines but to the user it would appear as a simple file structure.

The cloud has a "one click" option that allows the file owner to quickly create a secure hyperlink that can be shared with other users. This hyperlink can be embedded in websites, emails, reports, DUAP etc. all while maintaining the data in a central repository.

The DUAP team was setup with full credentials to the site to allow them full access to the data. Several discussions were conducted with the DUAP team on the matter. Lee Mixon of Mixon-Hill harvested the Gordonville Road data set from the SSI cloud system and was able to integrate the data into the DUAP database. A future project could use this to create custom tools helpful to MDOT operations, such as integrating 3D data into connected vehicle programs.

The site is setup so it can allow any device to directly access the site. The remote device would also need to have the capabilities to automatically upload its data to the site.

All the data for the project was hosted on the site and still currently resides on the site (Figure 88). Data owners and end users can access the site as a mapped drive, through a web portal, through a hyperlink, or through their iOS / Android devices. This allows complete flexibility for accessing the data.



Holton	01/05/2018 11:48 ...	File folder
US31_WhiteRiver	12/12/2017 9:54 AM	File folder
UncleHenryRd_bridge	06/30/2017 9:09 AM	File folder
US31_WhiteRiver_M120_WilsonBeach-C...	06/19/2017 11:17 ...	File folder
BeyerRd_bridge	02/28/2017 9:13 AM	File folder
GordonvilleRd_bridge	01/20/2017 3:30 PM	File folder
WarrenRd_bridge	12/20/2016 9:26 AM	File folder
I-75_Mobile_vs_UAV	12/07/2016 4:57 PM	File folder
MailX Images	12/05/2016 2:06 PM	File folder
SSI_I-75Mega_UAV	10/26/2016 2:16 PM	File folder
High Resolution UAV	10/13/2016 8:45 AM	File folder
M-120_Whitehall_Getty	09/20/2016 8:31 AM	File folder

Figure 88: Data hosted on the SSI cloud from the various project data collection efforts.

4.5 Cost-Benefit Analysis / Return on Investment

As the infrastructure industry has adopted UAVs (Tatum and Liu 2017), an important application has been to combine them with thermal sensors, optical sensors, and lidar to determine surface and sub-surface defects in concrete bridge decks. The first phase of this application focuses on the inspection stage, with long-term goals of extending the application to other project life-cycle stages as well. This research looks at the application of UAVs to locate spalls and delamination in concrete bridge decks efficiently with an eye to ensuring that the costs do not outweigh the benefits. Hence, the objective of Task 7 of this research was to develop a framework for conducting a cost-benefit analysis to compare the overall economies of two comparable Non-Destructive Testing (NDT) techniques used to evaluate bridge deck condition. The conclusions presented through this analysis are based on the data collected from two bridges namely, Beyer Road Bridge and Uncle Henry Road Bridge, both located in Saginaw County. Figure 89 indicates the scope of the project.

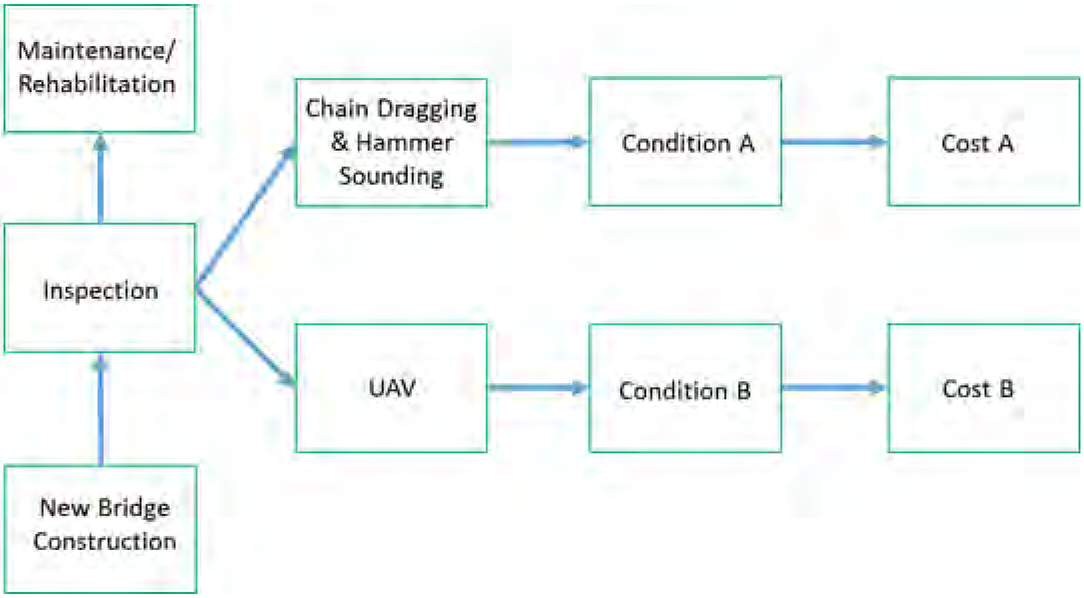


Figure 89: Scope of Cost-Benefit Analysis.

The comparable NDT methods used in this study are chain dragging (CD) and hammer sounding (HS). UAV based methods have significant advantages over techniques such as ground penetrating radar, CD and HS, impact echo method, infrared thermography and several others are widely used to assess the condition of a bridge deck (Yehia, Abudayyeh, Nabulsi and Abdelqader, 2007). While CD and HS methods are simple to use, economical, and hence a widely adopted NDT technique for inspection of bridge decks, distress data from this method is highly subjective and vary based on the proficiency of inspection personnel. In addition, accuracy and precision of this method are low as compared to other NDT methods (Gucunski et al., 2012). Additionally, CD and HS require complete stoppage of traffic during the inspection. This limitation is shared by other NDT techniques as well.

To understand the benefits and costs associated with the two NDT techniques under consideration, a literature review was conducted about NDTs in general. As far as the application of UAV technique is concerned a limited literature is available as it is fairly new, while CD and HS has a fair amount of literature. Previous research on NDT techniques for bridge inspection, emphasizes the use of qualitative assessment and, predicted rankings [(Yehia, Abudayyeh, Nabulsi and Abdelqader, 2007), (Omar, Nehdi and Zayed, 2017), (Huang, Adams and Pincheira, 2004)]. These rankings were based on the accuracy, precision, speed, ease of use, and other factors for different NDT methods. However, this study aims to quantitatively compare the distresses from UAV technique with CD and HS. The study by Gucunski et al., (2012) is most relevant to this research as it compares the costs associated with different NDT techniques. Gucunski et al., (2012) considered the following costs:

1. Cost of data collection
2. Cost of data analysis and interpretation
3. Cost of equipment, supplies and equipment maintenance and,
4. Cost of traffic control

All these costs could be treated as fixed costs for a bridge data collection. However, a drawback of previous studies is that they did not use objective measures to assess the quality of data from different NDT techniques. This may be attributed to the absence of industry standards for quantitatively comparing the accuracy and precision of data. Hence, this study indirectly accounts for data quality using maintenance costs as an analog. The argument is that if the condition rating returned by an NDT method over or underestimates the extent of deterioration of a bridge deck, then the associated maintenance costs will be inconsistent as well. For instance, if the actual condition is fair, but the data indicates a poor condition, then the actual cost of maintenance will likely be higher than what it should be. This discrepancy is used as an indirect indicator of data quality. To indicate the quality of data from CD and HS, and UAV methods, maintenance actions mentioned in Huang, Adams, and Pincheira (2004) and the MDOT Bridge Management and Scoping (2018) document were considered.

The common maintenance actions for concrete bridge decks include deck patching, concrete overlay, asphaltic concrete overlay without and with membrane, and deck replacement (Huang, Adams and Pincheira 2004). There may be some additional maintenance actions under consideration dependent on the scenario. These actions are usually used in schedules established by agencies using specific bridge deterioration models. The prevailing deterioration model for MDOT is indicated by Figure 90, based on the inspections conducted in the period 2006-2008. The model uses a stochastic method that determines the median deterioration year for a condition using transition probability from one condition to another condition of a bridge. As per this model, it takes 18 years for a bridge to deteriorate to “fair” condition and 52 years to reach “poor” condition. These conditions correspond to ratings of “6” and “4” respectively of the National Bridge Inventory (NBI) ratings. Additionally, to enhance the efficiency of a Bridge Management System (BMS), project-specific bridge deterioration models could be formulated using the interface developed by Winn and Burgueño (2013) for MDOT. However, due to unavailability of maintenance schedules, the cost-benefit analysis was conducted by considering these maintenance actions individually. The cost of a given schedule could be easily derived by adding the individual costs of the maintenance actions.

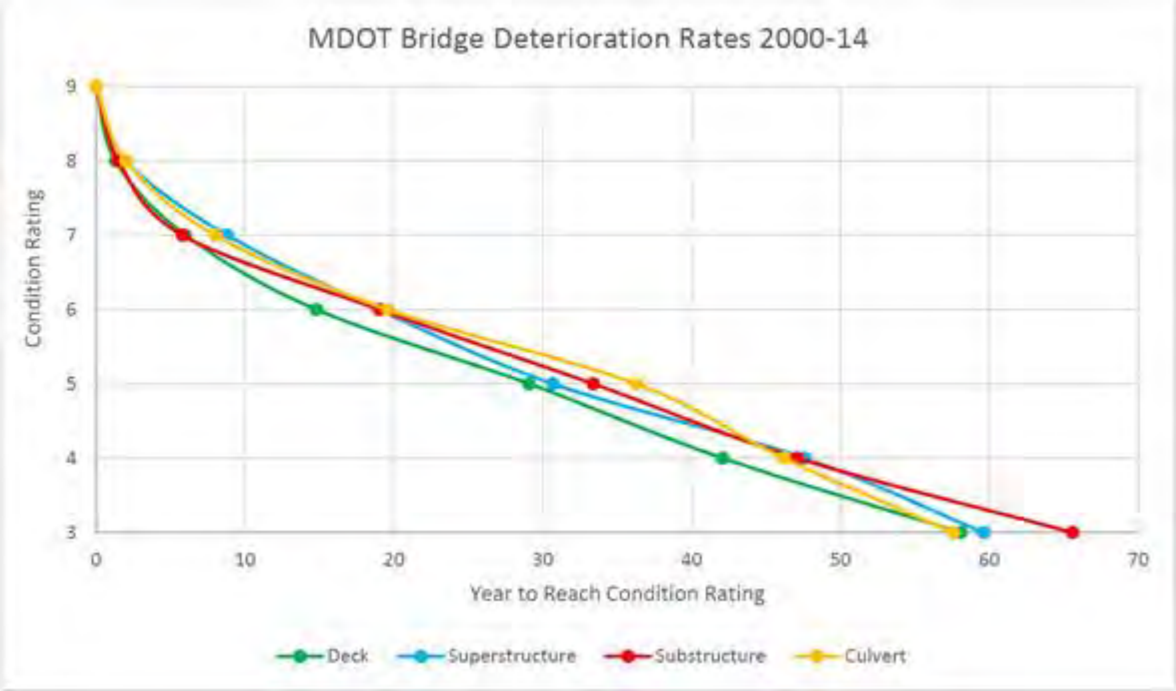


Figure 90: MDOT Bridge Deterioration Rates 2000-14 (Kelley, 2016).

As mentioned in the previous section, a major factor determining the efficiency of an NDT technique is the reliability of its data. Unfortunately, at present, there are no standardized guidelines to quantify accuracy and precision of data. Hence, we consider the costs of future maintenance actions to indirectly account for the quality of data from the two NDT techniques under consideration. Table 5 indicates the unit costs of maintenance actions under consideration. The costs mentioned in Table 5 are based on MDOT's price specifications as per 2017.

Table 5: Common Maintenance Actions for Concrete Decks ***[(Huang, Adams and Pincheira, 2004), (MDOT - Bridge Management and Scoping, 2018)].***

Method	Cost/sq ft ($)	Estimated Service Life (years)
Patching	35	3
Concrete Overlay	37	15
Asphaltic concrete overlay without membrane	1.5	3
Asphaltic concrete overlay with membrane	5.5	6
Deck replacement (new deck with epoxy-coated bars)	70	30

Several methods are available to conduct a cost-benefit analysis (Remer and Nieto 1995). In this study, cost-benefit analysis is conducted with the aid of Net Present Value (NPV). NPV indicates the present worth of a maintenance action for an assumed service period and discount rate. The NPV is computed for individual maintenance actions by assuming a service life of 30 years and a discount rate of 3.5% and with the aid of expenditure stream diagrams, or cash-flow diagrams presented through Figure 91. These are based on the estimated service life and cost for each maintenance action mentioned in Table 5. Figure 91 presents the cash-flow diagrams through which the NPVs are calculated for the study.

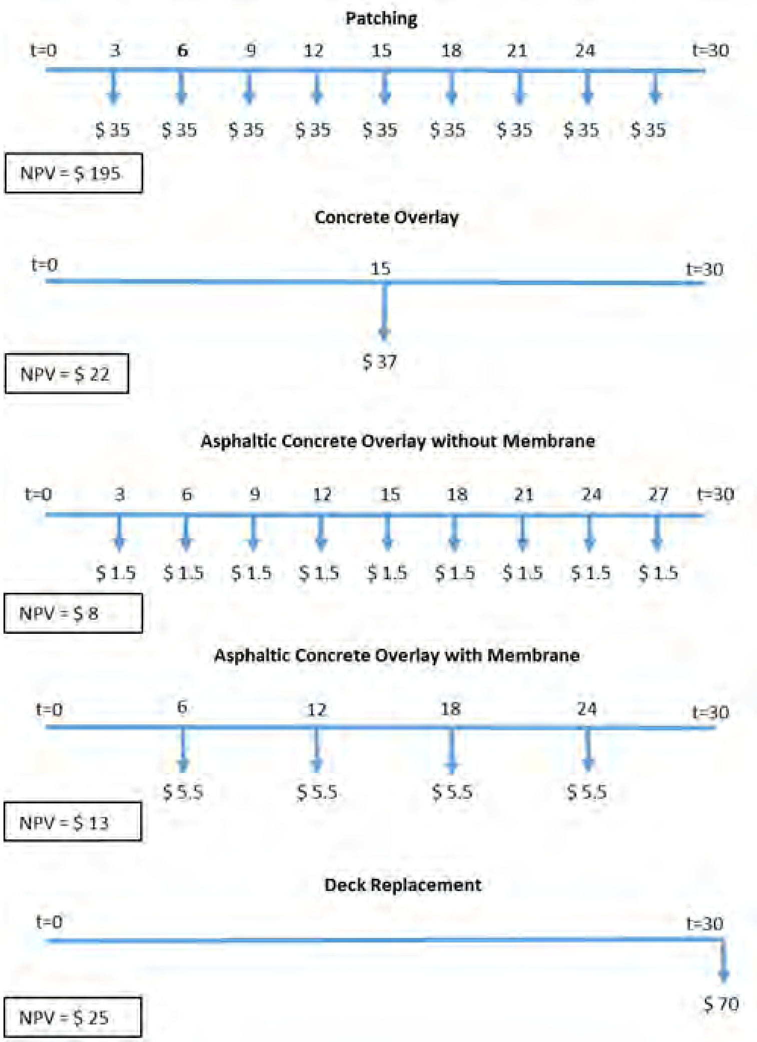


Figure 91: Cash-Flow Diagram.

Figure 92 presents the graphical representation of NPV from different maintenance actions mentioned in Table 5. Patching has significantly higher NPV than other maintenance actions.

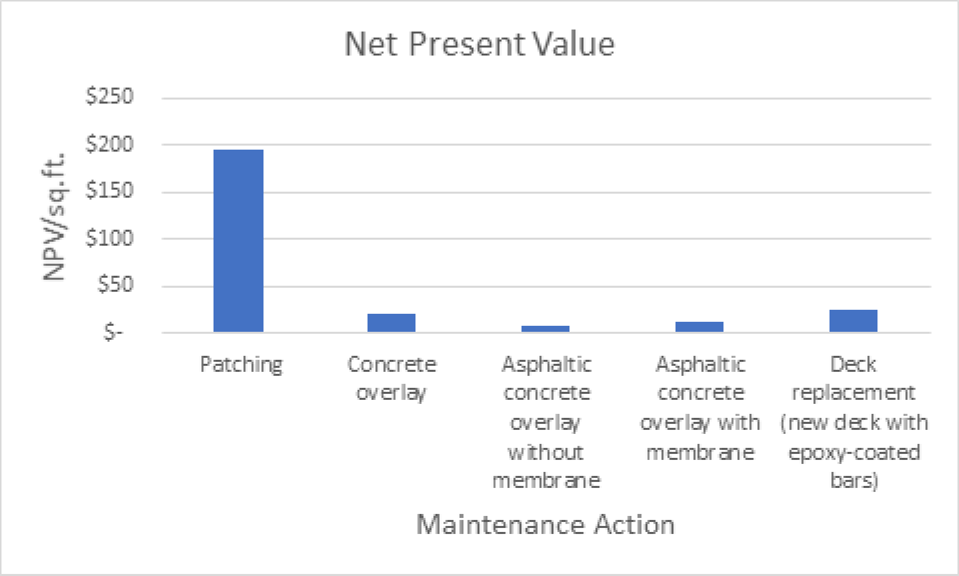


Figure 92: Net Present Value.

The cash-flow diagrams could potentially calculate the treatment costs for a bridge with certain extent of deterioration and maintenance scenarios. For the prevailing study, the cash-flow diagrams are applied to calculate the NPV of the above-mentioned maintenance actions for two bridges: Uncle Henry Road and Beyer Road in Saginaw County. However, in practice, the maintenance actions mentioned in Table 5 are applied in various combinations and not individually.

Table 6 presents the deterioration data for these two bridges obtained through both UAV and CD and HS NDT techniques.

Table 6: Distress Data

Distress: Delamination (sq ft)	**NDT Technique**	
Bridge	**UAV**	**CD and HS**
Uncle Henry Road	53.59	188.0
Beyer Road	92.73	313.28

For the amount of distress mentioned in Table 6, Table 7 presents the NPV for both Uncle Henry Road Bridge and Beyer Road Bridge.

Table 7: Net Present Value Table

	Net Present Value			
Bridge	**Uncle Henry Road**		**Beyer Road**	
NDT Technique	**UAV**	**CD and HS**	**UAV**	**CD and HS**
Condition	Fair	Poor	Fair	Poor
Treatments				

Patching	$10,438	$35,614	$18,061	$61,016
Concrete Overlay	$1,184	$4,152	$2,048	$6,919
Asphaltic concrete overlay without membrane	$447	$1,569	$774	$2,615
Asphaltic concrete overlay with membrane	$723	$3,364	$1,250	$4,224
Deck replacement (new deck with epoxy-coated bars)	$1,337	$4,688	$2,313	$7,813

From Table 7, it can be inferred that CD and HS techniques over-estimate the amounts of distress, at least compared to UAV methods, and hence could lead to higher maintenance costs than are needed. This difference in area estimation may be due to the UAV-based thermal imaging of bridge deck method being able to target delamination distresses more precisely in location and size vs. manual CD & HS methods (as shown earlier in Figures 82 and 83). However, in practice, the above-mentioned maintenance actions will be applied in various combination scenarios, and not individually. Hence, for practical applications, there is a need to collaborate with the MDOT bridge management and scoping team, and formulate costs for project-specific maintenance scenarios. There is a need to establish industry standards to quantify the accuracy and precision associated with the data from different NDT techniques.

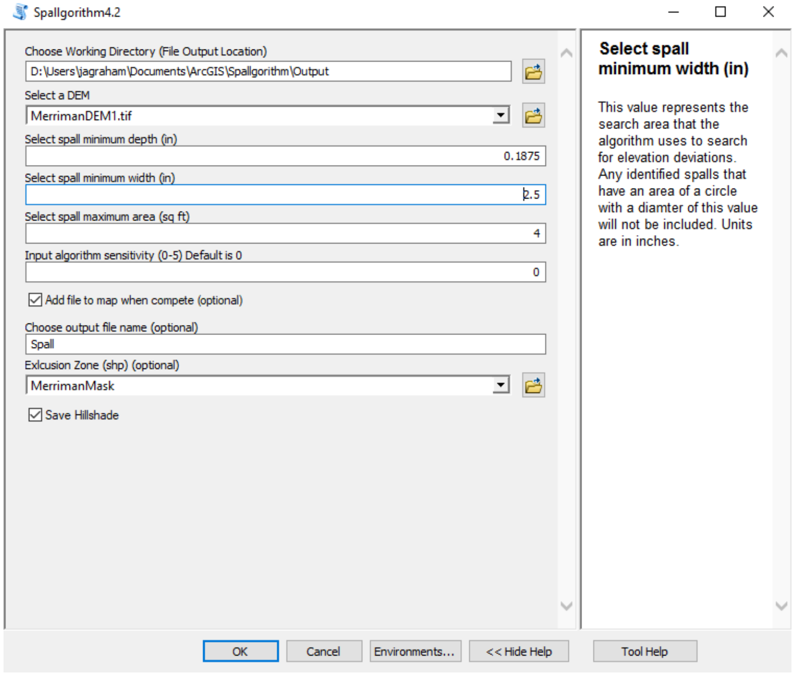
Moving forward, an important recommendation for MDOT is to invest in the development of element level data models of bridge surface damage elements using the geometry captured by UAVs. The purpose of these data models will be to attach semantics to the captured geometry using data structures that reflect the age of the bridge surface, point in time the element is captured, and its progression through the bridge life cycle. This can be very useful in further improving bridge deterioration models as a function of how individual distresses behave over time. It can provide a baseline for assessing the quality of data captured in NDT techniques, and improve the accuracy and precision of the assessed bridge condition. In turn, this can help optimize the cost of bridge maintenance and rehabilitation.

4.6 Spall Detection Algorithm – "Spallgorithm"

The spall detection algorithm or "Spallgorithm" is the newest version of the spall detection algorithm that makes use of road surface DEMs to delineate spall features based upon detection of height anomalies (existence of detectable depressions). The tool is a Python script now improved to work directly within ESRI's Desktop ArcGIS with Python 2.7 and comes packaged within an ArcGIS toolbox or can also be ran as a standalone script. It has been tested as compatible with ArcGIS versions up to 10.4.1. Instructions for installation of the tool can be found in the appended Spallgorithm user manual for potential implementation by MDOT (see Appendix Section 9.3).

The only required input for the Spallgorithm is a DEM of a bridge deck or road surface. Better results from the tool are achieved from DEMs with high accuracy and small (high) spatial resolution. Photogrammetrically derived DEMs with 2cm or better resolution and accuracy from the Nikon D810 have been tested and have demonstrated successful demarcation of spall features on several bridge decks. Similar resolution lidar elevation data should work as well. An optional input is the use of an exclusion zone which allows users to set boundaries on the area of the DEM to be excluded for analysis. This is particularly useful for avoiding bridge joints and also along the edges of the road, where DEM construction can be noisy as a result of accumulated debris or artifacts from guardrails. Output from the Spallgorithm is a single GIS shapefile which contains attribute data about each spall feature, including: coordinates, area, depth, and volume. Optionally, a hillshade layer derived from the DEM can also be created and saved by running the tool to help with visualization. Additional details about running the tool and examining outputs are included in the user manual.

The most current version has several notable upgrades from previous versions. Spallgorithm 4.2 is a more user friendly tool, coming with integrated help documentation, a user manual, and interfacing through the Desktop ArcGIS toolbox. The tool features drag and drop inputs, output directory and name specification, optional addition to map, and also optional creation of a hillshade layer (Figure 93). Also optional is the use of a user created layer which defines the area of the bridge to exclude from analysis. This can improve results by excluding bridge joints and the edges of the bridge, where DEM reconstruction contains noise and can be influenced by road artifacts such as debris and railings. The running time of the tool has been significantly reduced, with average processing times between 30 seconds and 2 minutes, depending on computer specifications. Additionally, the tool self-deletes all intermediary data that is created during a run, leaving the user with only the output spall shapefile (Figure 94), helping to save space and reduce confusion with interim datasets. The attribute table of the shapefile has been reduced to contain only relevant information: location, spall area, spall depth, and spall volume. Fields contain both metric and imperial units. The approach to the spall detection algorithm was overhauled, which has, in addition to processing time, has greatly increased accuracy. A potential next step would be to train MDOT staff in how to implement the tool, potentially starting with specific test region or two.



Spallgorithm4.2

Choose Working Directory (File Output Location)
D:\Users\jagraham\Documents\ArcGIS\Spallgorithm\Output

Select a DEM
MerrimanDEM1.tif

Select spall minimum depth (in)
0.1875

Select spall minimum width (in)
2.5

Select spall maximum area (sq ft)
4

Input algorithm sensitivity (0-5) Default is 0
0

☑ Add file to map when compete (optional)

Choose output file name (optional)
Spall

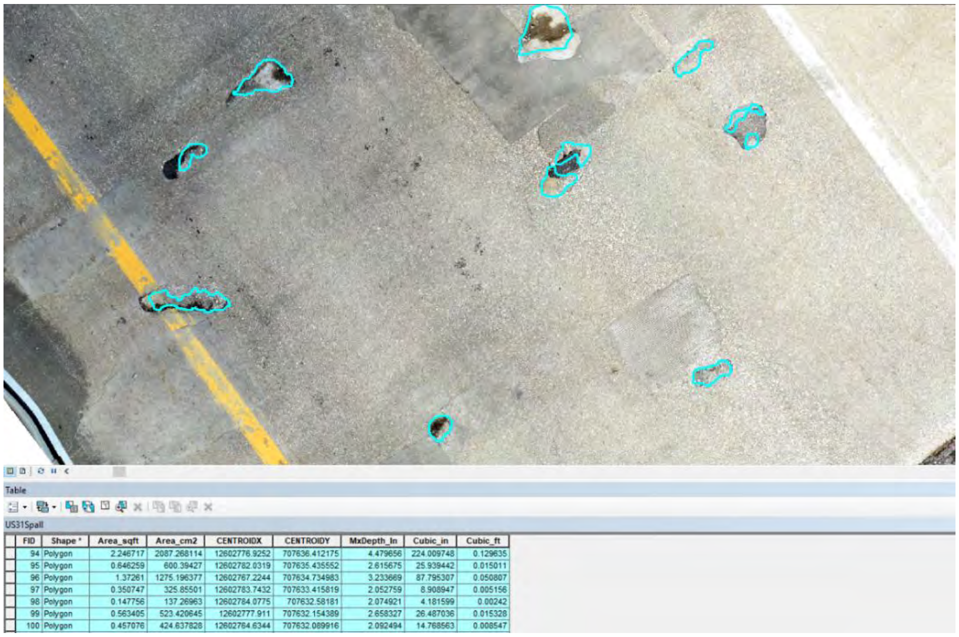
Exlcusion Zone (shp) (optional)
MerrimanMask

☑ Save Hillshade

Select spall minimum width (in)

This value represents the search area that the algorithm uses to search for elevation deviations. Any identified spalls that have an area of a circle with a diamter of this value will not be included. Units are in inches.

Figure 93: The user interface of the Spallgorithm with tool help displayed in the right window.



Table

US31Spall

FID	Shape *	Area_sqft	Area_cm2	CENTROIDX	CENTROIDY	MxDepth_In	Cubic_in	Cubic_ft
94	Polygon	2.246717	2087.268114	12602776.9252	707636.412175	4.479656	224.009748	0.129635
95	Polygon	0.646259	600.39427	12602782.0319	707635.435552	2.615675	25.939442	0.015011
96	Polygon	1.37261	1275.196377	12602767.2244	707634.734983	3.233669	87.795307	0.050807
97	Polygon	0.350747	325.85501	12602783.7432	707633.415819	2.052759	8.908947	0.005156
98	Polygon	0.147756	137.26963	12602784.0775	707632.58181	2.074921	4.181599	0.00242
99	Polygon	0.563405	523.420645	12602777.911	707632.154389	2.658327	26.487036	0.015328
100	Polygon	0.457076	424.637828	12602764.6344	707632.089916	2.092494	14.768563	0.008547

Figure 94: Example output from the Spallgorithm tool from the US31 bridge dataset showing detected spalls and associated attribute data such as area and maximum depth.

4.7 Thermal and Visible Delamination Detection Algorithm

The Thermal Delamination Detection Algorithm was developed as a tool for user-friendly processing of UAV acquired remote sensing data. Originally developed in MATLAB, for developing and testing purposes, it was translated to ArcPy due to the need to make the tool available in a more user friendly and accessible platform, i. e. through ESRI's Desktop ArcGIS software as a Python toolbox. The toolbox is imported into ArcGIS as a Python toolbox (Figure 95) and executed from there for data processing and analysis.

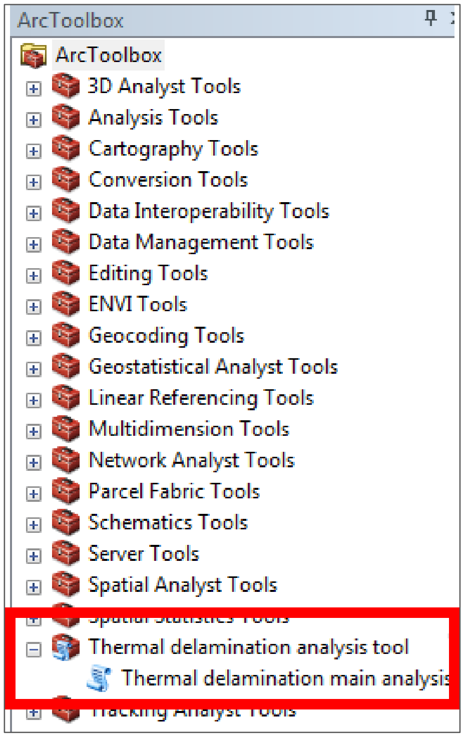
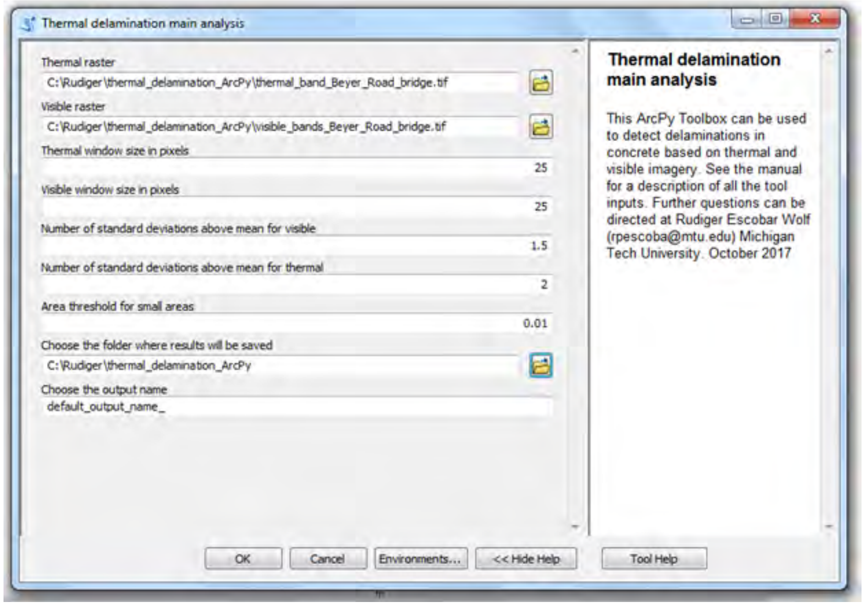


Figure 95: Thermal delamination analysis tool and thermal delamination main analysis script, displayed in the Desktop ArcGIS toolbox.

The toolbox requires two datasets, and a series of user defined parameters as inputs. Executing the tool brings up a standard ArcMap dialog box in which the user has to introduce the data files and parameter values (Figure 96). Thermal and visible image data are given in any file format compatible with ArcGIS (e. g. geotiff), importantly, both raster files have to be defined in the same coordinate system, and this requires the data having previously been georeferenced. The user-tunable parameters include the window sizes and number of standard deviations above the mean, for helping to analyze the thermal and visible images. The window sizes refer to the size of the squared neighborhood of pixels that will be analyzed for each pixel in the thermal and visible images, to extract mean and standard deviations values, that will be used to assess if a pixel is anomalous or not, i. e. if its value is far enough from the mean of surrounding pixels, as defined by the user given number of standard deviations, to be considered an anomalous pixel. A threshold area (given in the units of the projected datasets) is also implemented to avoid mapping single pixels in noisy data. Finally, the user has to define an output location and file name for the resulting polygons of mapped delaminations in a standard shapefile format compatible with all GIS software.



Thermal delamination main analysis

Thermal raster
C:\Rudiger\thermal_delamination_ArcPy\thermal_band_Beyer_Road_bridge.tif

Visible raster
C:\Rudiger\thermal_delamination_ArcPy\visible_bands_Beyer_Road_bridge.tif

Thermal window size in pixels
25

Visible window size in pixels
25

Number of standard deviations above mean for visible
1.5

Number of standard deviations above mean for thermal
2

Area threshold for small areas
0.01

Choose the folder where results will be saved
C:\Rudiger\thermal_delamination_ArcPy

Choose the output name
default_output_name_

Thermal delamination main analysis

This ArcPy Toolbox can be used to detect delaminations in concrete based on thermal and visible imagery. See the manual for a description of all the tool inputs. Further questions can be directed at Rudiger Escobar Wolf (rpescoba@mtu.edu) Michigan Tech University. October 2017

Figure 96: Thermal and visible delamination tool dialog box.

The resulting polygon can then be loaded into Desktop ArcGIS or other GIS software to be analyzed with other datasets for the target road, bridge, or transportation corridor. All parameter values are initially populated with default values, which are likely to work in most cases, but the user can introduce their own chosen values to do a more specific analysis. Sometimes the process requires some trial and error testing, varying the window sizes, but particularly changing the number of standard deviation thresholds for both the thermal and the visible datasets. With a TDDA user manual included as Appendix Section 9.4, a future project phase could include rolling out the tool to MDOT users through training and implementation sessions, perhaps at a target region or two to start with.

4.8 Crack Detection Algorithm Assessment

Cracking has come up with discussions with MDOT staff as a type of road and bridge distress that would desirable to be able to map and quantify on a reliable, automated basis. While not part of the project's original statement of work, the project team did investigate how cracks could be automatically detected with the type of high-resolution imagery being collected via UAV, with the idea that this effort could be followed up in a potential future project phase if results were promising. The team developed three initial types of crack detection algorithms that show promising results that have the opportunity to become operational algorithms with additional research effort. These were:

1. A low-pass filter crack detection method
2. A crack detection through saddle point detection method
3. A crack detection through color analysis method

With all three methods producing interesting results, they could become particularly useful for assessing longer corridors as part of MDOT's efforts to assess and manage their roadway assets.

Low-pass Filter Crack Detection Method

Image based crack detection on speckled road surfaces, such as some types of asphalt, is difficult due to high noise in the image. The complication of parsing actual cracks from the background is exacerbated from the variable pixel data of the road surface. However, a wide variety of image processing techniques are available and may be suited for different road surfaces. Some preliminary efforts demonstrate the feasibility of image based crack detection on surfaces of an irregular nature (for example, see Zou et al. 2012). An approach shown here uses a sample from a geographically referenced orthomosaic constructed from UAV flight imagery on US-31 north of the northbound White River bridge flown on an oblique angle (Figure 97). The original image (A) has pixel values inverted to (B) and mathematically transformed to stretch the high end pixel values (C). The image is smoothed using a series of low-pass filters (D) and filtered by an edge detection kernel (E) and pixels are thresholded to highlight cracks (F). The process builds hillshade like outputs which may aid in defect visualization and resulting data outputs are georeferenced, potentially allowing features like width, length and location to be known.

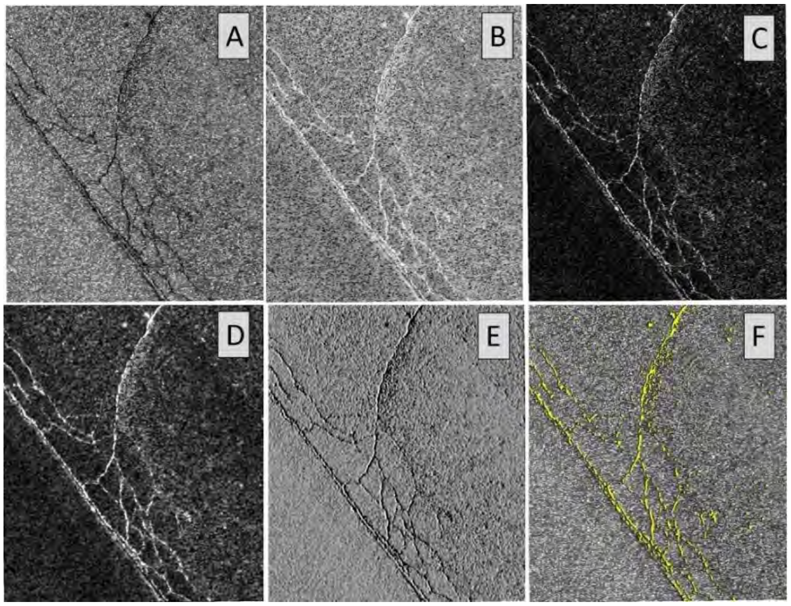


Figure 97: Example processing results and outputs from a low-pass filter method to detect cracks using Nikon D810 imagery collected from a UAV.

Crack Detection Through Saddle Point Detection

This method is run on a single image band where it exploits the tendency that cracks tend to be darker than surrounding pavement and there is a gradual change in brightness in the pixels that cross them. MATLAB is used to run the initial detection on cracks and the final product is produced in ArcGIS. The first step is to identify a threshold value for what is considered to be a crack. These pixels are then selected and then run through a "saddle point" detection method which selects the “middle” of the detected crack. Potential cracks are highlighted in a raster image (Figure 98 (B)) where crack are a value greater than 0 and the background is 0. This output is then brought into ArcGIS to merge neighboring line segments into a shapefile of potential cracks.

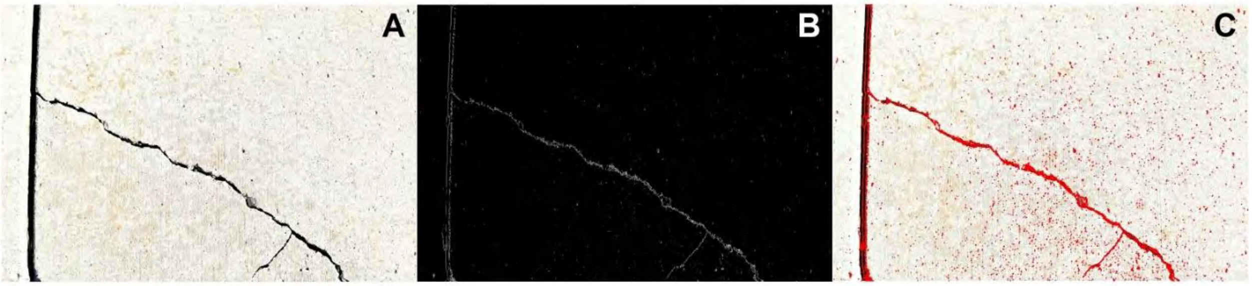


Figure 98: A. The original image of concrete containing cracks. B. Output from Matlab code where white is “detected” crack. C. Final merged product over original image.

Crack Detection Through Color Analysis

This method uses digital imagery (with regular red / green / blue color bands) to look for distinct color patterns that indicate cracks. Multispectral and hyperspectral imagery would be likely to enhance crack detection by providing additional colors to differentiate cracks from non-cracked pavement (Resende et al. 2014). For this method, an input of cracked road pavement was developed, using an input image from an area of asphalt pavement in a road that goes around the MTRI facility in Ann Arbor (Figure 99 (A)). The image was run through the ISODATA algorithm available within Desktop ArcGIS (version 10.4.1), frequently used for land cover analysis (Otukei and Blaschke 2010), to differentiate the pavement into different classes, resulting in Figure 99 (B), where dark crack areas have been differentiated into a separate class shown in the yellow areas. The areas classified as likely cracks were made into a single separate layer (Figure 99 (C)), which was then run through the Desktop ArcGIS “Thin” routine to create vectors of likely cracks (Figure 99 (D)). The yellow lines in Figure 99 (D) follow the visible cracking very closely; with improvements, this process could be automated for crack detection.

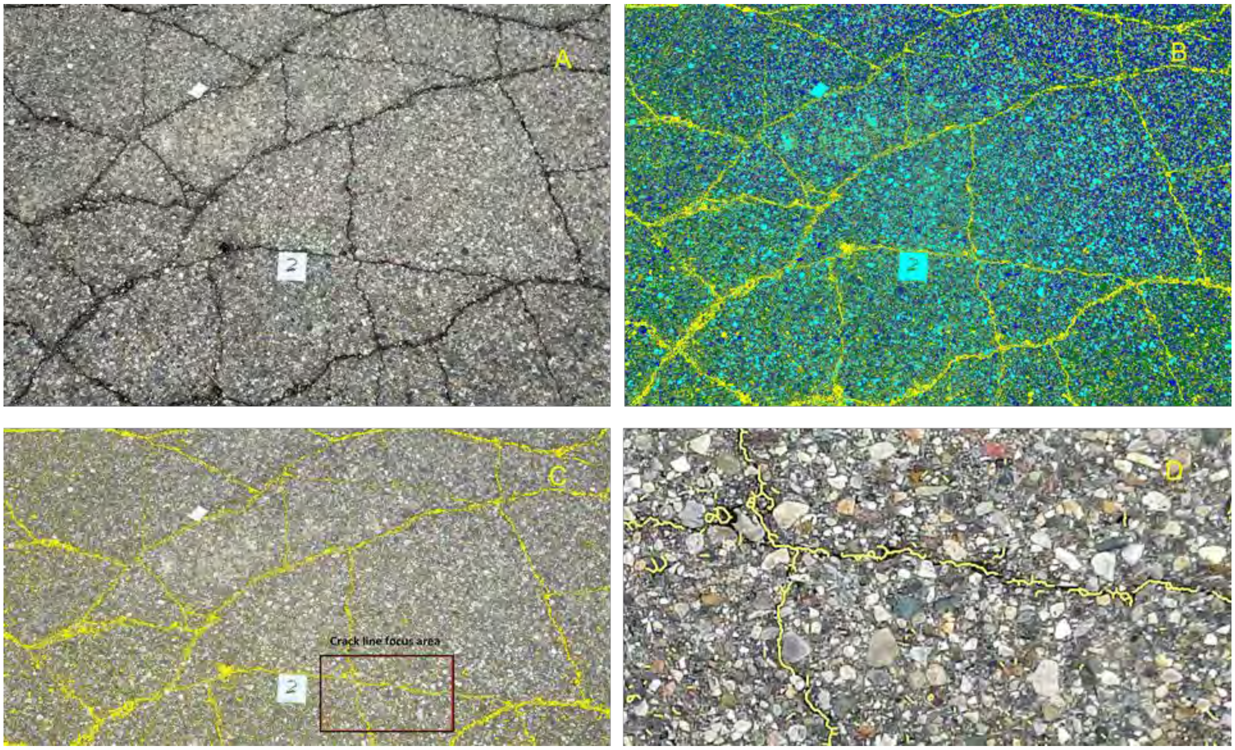
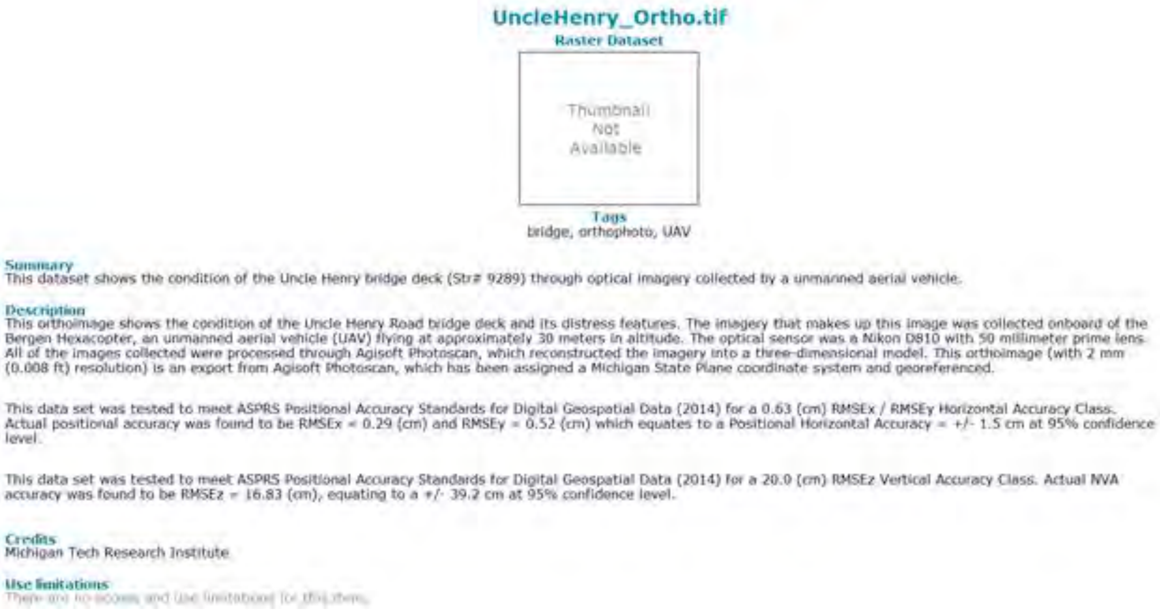


Figure 99: Example results from the color analysis method of crack detection. Part A shows the input image of cracked road pavement, Part B shows an image classification result with yellow indicating likely crack areas in yellow, part C shows just the likely crack areas (in yellow) with the inset area showing likely crack lines highlighted in part D.

With three rapid demonstrations of finding cracks in pavement examples, the project team recommends that this be further developed by MDOT in a potential future project phase to extend the value of UAV-collected imagery for corridors and bridges that are likely to become part of MDOT operations.

4.9 Metadata with Accuracy Documentation

Metadata was written for each dataset placed onto MTRI's FTP site and shared with SSI, following Federal Geographic Data Committee (FGDC) Content Standard for Digital Geospatial Metadata (CSDGM). This provides a "chain of evidence" capability for when data products are used for other purposes, such as development of applications using DUAP data. The metadata was written in ESRI's ArcCatalog and includes information such as key words, a summary, a description, credits, and contacts (Figure 100).



UncleHenry_Ortho.tif

Raster Dataset

Thumbnail Not Available

Tags

bridge, orthophoto, UAV

Summary

This dataset shows the condition of the Uncle Henry bridge deck (Str# 9289) through optical imagery collected by a unmanned aerial vehicle.

Description

This orthoimage shows the condition of the Uncle Henry Road bridge deck and its distress features. The imagery that makes up this image was collected onboard of the Bergen Hexacopter, an unmanned aerial vehicle (UAV) flying at approximately 30 meters in altitude. The optical sensor was a Nikon D810 with 50 millimeter prime lens. All of the images collected were processed through Agisoft Photoscan, which reconstructed the imagery into a three-dimensional model. This orthoimage (with 2 mm (0.008 ft) resolution) is an export from Agisoft Photoscan, which has been assigned a Michigan State Plane coordinate system and georeferenced.

This data set was tested to meet ASPRS Positional Accuracy Standards for Digital Geospatial Data (2014) for a 0.63 (cm) RMSEx / RMSEy Horizontal Accuracy Class. Actual positional accuracy was found to be RMSEx = 0.29 (cm) and RMSEy = 0.52 (cm) which equates to a Positional Horizontal Accuracy = +/- 1.5 cm at 95% confidence level.

This data set was tested to meet ASPRS Positional Accuracy Standards for Digital Geospatial Data (2014) for a 20.0 (cm) RMSEz Vertical Accuracy Class. Actual NVA accuracy was found to be RMSEz = 16.83 (cm), equating to a +/- 39.2 cm at 95% confidence level.

Credits

Michigan Tech Research Institute

Use limitations

There are no access and use limitations for this item.

Figure 100: Screenshot of metadata for Uncle Henry Rd's orthoimage geospatial output layer.

Additionally, for applicable locations and datasets (such as the orthophoto), an accuracy assessment of geolocation quality was conducted using the high-accuracy GPS points collected by SSI as ground truth data, with the results being stated in the metadata. Following the American Society for Photogrammetry and Remote Sensing (ASPRS)'s Positional Accuracy Standards for Digital Geospatial Data, the Root Mean Square Error (RMSE) for horizontal (x, y) and vertical (z) accuracy was calculated at the 95% confidence level (more information is https://www.asprs.org/pad-division/asprs-positional-accuracy-standards-for-digital-geospatial-data.html and the Smith and Heidemann 2015 article). For example, results indicated that for the Uncle Henry orthophoto:

"This data set was tested to meet ASPRS Positional Accuracy Standards for Digital Geospatial Data (2014) for a 0.63 (cm) RMSEx / RMSEy Horizontal Accuracy Class. Actual positional accuracy was found to be RMSEx = 0.29 (cm) and RMSEy = 0.52 (cm) which equates to a Positional Horizontal Accuracy = +/- 1.5 cm at 95% confidence level."

"This data set was tested to meet ASPRS Positional Accuracy Standards for Digital Geospatial Data (2014) for a 20.0 (cm) RMSEz Vertical Accuracy Class. Actual NVA accuracy was found to be RMSEz = 16.83 (cm), equating to a +/- 39.2 cm at 95% confidence level."

Furthermore, the point cloud created from the processing of the high-resolution optical imagery was assessed for accuracy and resolution using the NCHRP Report 748, at the recommendation of the project's SSI partners, based on it being useful for documenting the quality of mobile lidar data. For example, the Uncle Henry point cloud has a point density of

3,906 points per square meter, which categorized the point cloud as fine density. Additionally, since the Positional Horizontal Accuracy was calculated to be +/- 1.5 cm at a 95% confidence level, NCHRP categorizes the point cloud as high accuracy. Therefore, the point cloud is designated as Level 1-A High Accuracy and Fine Density.

4.10 List of Geospatial Products for Each Location

For each project site, certain GIS-compatible output products were created, and made available for sharing through the SSI cloud portal for potential integration into the DUAP database. These were:

South Warren Road
(ftp://ftp.mtri.org/pub/MDOT_UAV_Phase2/SouthWarrenRd/)

1. Orthoimage (Nikon D810)
2. Hillshade (Nikon D810)
3. Thermal Image (FLIR Vue Pro)

Gordonville Road
(ftp://ftp.mtri.org/pub/MDOT_UAV_Phase2/Gordonville/)

1. Point cloud (Nikon D810)
2. Orthoimage (Nikon D810)
3. DEM (Nikon D810)
4. Hillshade (Nikon D810)
5. Overview (DJI Phantom 3A)
6. Exclusion Zones
7. Spalls (Nikon D810)
8. Thermal Image (FLIR Vue Pro)
9. Manually Detected Delaminations
10. High-accuracy GPS Points (from SSI)

Uncle Henry Road
(ftp://ftp.mtri.org/pub/MDOT_UAV_Phase2/UncleHenry/)

1. Point Cloud (Nikon D810)
2. Orthoimage (Nikon D810)
3. DEM (Nikon D810)
4. Hillshade (Nikon D810)
5. Overview (DJI Phantom 3A)
6. Exclusion Zones
7. UAV-Detected Delaminations
8. Spalls (Nikon D810)
9. Thermal Image (FLIR Vue Pro)

10. Manually Detected Delaminations
11. Overview - November 2016 (DJI Phantom 3A)
12. Overview - December 2016 (DJI Phantom 3A)

Beyer Road
(ftp://ftp.mtri.org/pub/MDOT_UAV_Phase2/Beyer/)

1. Point Cloud (Nikon D810)
2. Orthoimage (Nikon D810)
3. DEM (Nikon D810)
4. Hillshade (Nikon D810)
5. Overview (Nikon D810)
6. Overview (DJI Phantom 3A)
7. Exclusion Zones
8. Spalls (Nikon D810)
9. Thermal Image (FLIR Vue Pro)
10. UAV-Detected Delaminations
11. Manually Detected Delaminations

M-120/Holton Road
(ftp://ftp.mtri.org/pub/MDOT_UAV_Phase2/Holton/)

1. Orthoimage (Nikon D810)
2. DEM (Nikon D810)
3. Hillshade (Nikon D810)
4. Orthoimage (DJI Mavic Pro)
5. DEM (DJI Mavic Pro)
6. Hillshade (DJI Mavic Pro)
7. Thermal Image (FLIR Vue Pro)
8. Thermal Image (FLIR Vue Pro R)
9. Radiometric Thermal Image - Fahrenheit (FLIR Vue Pro R)
10. Radiometric Thermal Image - Fahrenheit clipped to roadway surface only (FLIR Vue Pro R)
11. High-accuracy GPS Points (from SSI)

US-31 White River Corridor
(ftp://ftp.mtri.org/pub/MDOT_UAV_Phase2/US31_WhiteRiver/)

1. Orthoimage - Corridor (Nikon D810)
2. Orthoimage - Bridge Only (Nikon D810)
3. DEM - Corridor (Nikon D810)
4. Hillshade - Corridor (Nikon D810)
5. High-accuracy GPS Points (from SSI)

Note that thermal data were also collected for the US-31 corridor, but the project team did not consider them of sufficient quality to share due to the longer-distance, off-nadir angle that they were collected at so that traffic would not have to be closed.

Merriman Road (from Phase I)
(ftp://ftp.mtri.org/pub/MDOT_UAV_Phase2/Merriman_PhaseI/)

1. Orthoimage (Nikon D800)
2. DEM (Nikon D800)
3. Hillshade (Nikon D800)
4. Spalls (Nikon D800)
5. UAV-Detected Delaminations

5. Deliverables

5.1 Draft UAV Guidance Document Overview

Under Task 1C "Providing a report that describes and recommends optimal methods to store and distribute potentially large imaging, point cloud, and 3D surface datasets created through UAV-based data collection," a guidance document for using UAVs with MDOT Right-of-Way was written and provided to MDOT. The purpose of this document was to provide guidelines for an application process that enables safe and timely access to its Rights-of-Way (ROW) where external groups (companies, researchers, etc.) or internal staff may want to use UAVs to benefit MDOT.

The document provides sufficient detail about current UAV regulations (FAA Section 333 - Part 107) and the Part 107 waiver process for flights that operate outside of the regulations (i.e. flights at night, over people, etc.). Safety procedures and equipment were also provided and suggested that all MDOT UAV flights strictly follow the rules including pre-flight checks and developing a safety plan for each flight. Additionally, standard safety equipment and gear that are required by MDOT (i.e. hardhats, reflective vests, and eye and foot protection) were suggested as being required for all flights. A sample MDOT UAS Request Form was also included in the document, intending for it to be completed and reviewed by a MDOT standing committee that would either approve or deny the form and flights. The form included information such as the contractor/contacts information, pilot's information, UAV being used, and planned flight pattern information. Lastly, the document included an overview of a interview session between Michigan Tech and MDOT employees Steve Cook, P.E., Maintenance Engineer and Matt Brinker, Bureau of Aeronautics & Freight Services to discuss UAV operations priorities. The interview was conducted to better understand MDOT thoughts on UAV procedures and use in MDOT right-of-way. Some highlights of the interview included, 1. For each flight that occurs, MDOT must be notified and provided information concerning the data collection, 2. All flights must be approved by a MDOT UAS standing committee, and 3. All flight operations must follow both FAA and MDOT standards.

The guidance document was submitted and reviewed by MDOT and the state's Attorney General by March 2017. Although the document was found to be informative, it was eventually determined that the department would following guidelines set forth by the Attorney General office, which are currently that MDOT will follow FAA rules for operations of UAVs over its ROW, rather than setting up additional rules.

5.2 Technical Demonstration Session with Supporting Documents

As part of Task 6 "Demonstrate, develop, and implement systems management and operations uses," the project team will supply MDOT with device training and documentation. Originally, the plan was to hold two full training days with MDOT staff, overviewing how UAVs can be used in evaluating transportation infrastructure, basic control of the UAVs and sensors, and usage protocols. However, towards the end of the project, it was decided that the informing MDOT staff on the results of the project would better be served by providing a technical demonstration at the Operations and Traffic Safety Conference in Mount Pleasant between May 9th and 10th, 2018. During this event, the project team will provide a presentation overviewing the project, a UAV demonstration in a nearby parking lot / field area (weather permitting), a break-out session for a more in depth discussion, access to the UAV and sensor hardware, and a "Questions and Answers" session to discuss implementation next steps with MDOT staff.

6. Outreach and Reporting

The project team provided an extensive outreach effort, including presentations, posters, a TRB conference paper, and regular MDOT reporting. A MDOT "Spotlight" article will be forthcoming around completion of the project in coordination with MDOT's contractor for these types of articles. An Implementation Action Plan (IAP) is being produced from this report's content to provide a shorter view of next steps for implementation that can be shared with regions. Future peer-reviewed papers are likely and planned after project completion. Outreach efforts included:

1. 2018 TRB
    a. Presentations
        i. Academic Research Perspectives on Creating High-Quality Data for UAS Transportation Application. January 8, 2018
        ii. Implementing UAV Applications at the Michigan Department of Transportation. January 11, 2018.
    b. Paper
        i. A comprehensive overview to improve traffic flow observability using UAVs as mobile sensors.
    c. Poster
        i. Transportation Infrastructure Assessment Through the Use of UAVs.

2. 2017 TRB
    a. Presentations
        i. Drone Based Transportation Infrastructure Assessment. January 9, 2017.
        ii. Lessons Learned from Implementing UAS-Enables Research into the Assessment of Transportation Infrastructure. January 12, 2017.
    b. Paper
        i. Transportation Infrastructure Assessment Through the Use of Unmanned Aerial Vehicles
    c. Poster
        i. Transportation Infrastructure Assessment Through the Use of UAVs
3. 2017 Michigan Bridge Conference
    a. Presentation
        i. Using UAV Collected Bridge Condition Data for a Rating Exercise. March 21, 2017.
4. 2017 EMU Drone Day
    a. Presentation
        i. Implementing Applications of UAS-Enables Sensing through Applied Research Programs: *Transportation Infrastructure Assessment and Mapping Invasive Plants*. October 24, 2017.
5. 2016 Iowa Innovations in Transportation Conference
    a. Presentation
        i. Applying Unmanned Aerial Vehicles to Help Solve Transportation Infrastructure Condition Assessment Issues. C. Brooks. - August 16, 2016.
6. 2016 LTAP TTAP National Conference, Madison WI
    a. Presentation
        i. Evaluation and Implementation of UAVs for Transportation Applications. July 19, 2016.
7. 2016 USDOT Innovation Day, USDOT Headquarters, Washington, D.C.
    a. Presentation
        i. Research and Implementation Interests for Unmanned Aerial Vehicles: a Michigan DOT example. June 22, 2016.
    b. Poster
        i. MDOT UAV and Unpaved Roads Assessment. June 22, 2016.
8. 2017 INSPIRE University Transportation Center Webinar
    a. Presentation
        i. Drone-Enabled Remote Sensing for Transportation Infrastructure Assessment. December 13, 2017.
9. 2017 Midwest Rail Conference
    a. Presentation

            i. Application of UAVs for Inspecting Transportation Infrastructure. August 16, 2017.

10. 2017 Minnesota's Transportation Conference
    a. Presentation
        i. Evaluating the Use of Unmanned Aerial Vehicles for Transportation Purposes. March 01, 2017.
11. 2017 GeoYou Conference, St. Louis
    a. Presentation
        i. Applications of Remote Sensing for Environmental and Transportation Infrastructure Assessment to Meet Agency Decision Support Needs. September 13, 2017.
12. NCHRP Project 20-68A, Domestic Scan 17-01: Successful Approaches for the Use of Unmanned Aerial Systems by Surface Transportation Agencies, San Diego
    a. Presentation
        i. Implementing UAV Applications at the Michigan Department of Transportation: Progress through MDOT research. April 10, 2018.
13. MDOT Operations / Traffic & Safety 2018 Conference, Mt. Pleasant
    a. Presentation and training demonstration
        i. Implementation of Unmanned Aerial Vehicles for Assessment of Transportation Infrastructure – Phase II: Overview. May 9, 2018
14. Mid America Association of State Transportation Officials (MAASTO)
    a. August 2018
15. Spotlight Article (forthcoming)
16. Quarterly, Annual, and Final Reports
    a. Reports
        i. Two Annual Reports (Fiscal Years 2016 and 2017)
        ii. Eight Quarterly Reports
        iii. Implementation Action Plan, focused on implementation through MDOT regions (forthcoming)
        iv. This document serves as the Final Report, with the revised version being completed by May 31, 2018.

7. Conclusions

7.1 Conclusions from the Study

This applied research project enabled the continued testing and demonstration of five main UAV platforms, eight different sensors, four main types of infrastructure needs, extensive lab testing, and ten main sets of demonstrations. Ten different presentations were made at conferences across the country, enabling local, regional, and international exposure to the results funded by MDOT research. Useful data were collected by multiple UAVs and this information was shared through data portals set up by project partner SSI, through a data sharing platform for integration with the DUAP database, and through meetings, graphics, and other frequent communication with the project's research advisory panel (RAP) and managers.

The resulting data products showed the ability to rapidly and accurately detect bridge deck surface and subsurface condition issues using UAV-collected imagery, how a construction site could be quickly imaged to include aggregate quantities, and how small UAVs can be used to evaluate traffic conditions and provide overview basemap imagery of sites of interest. Improvements to the collection, processing, and visualization of data originally developed under the Phase 1 of this project were enabled through the project, including improved spall detection and delamination detection software tools compatible with Desktop ArcGIS software. Project work also included testing of a low-cost lidar sensor producing a point cloud and DEM of a testing site located in Ann Arbor, Michigan. A cost-benefit analysis indicated that traditional bridge deck assessment methods may be overestimating the area of defects such as delaminations; with UAV-based detection, the area of deck problems could be more accurately assessed, leading to lower maintenance costs and improved deterioration modeling.

Together, these research efforts have shown the potential benefits of implementing UAV into MDOT workflows, helping with many transportation management and operations data needs, including providing MDOT with updated methods to collect needed bridge element inspection data including identifying spalls with optical images and likely delaminations with thermal data, quantifying the volumetrics of construction aggregate mounds, providing flexible low-cost traffic monitoring, and helping to identify and inventory types of roadway assets through UAV-based image analysis.

UAV-based sensing of transportation infrastructure has become more practical. with the release of the Section 333 and Part 107 regulations on small UAVs from the FAA. Four members of the project team have successfully passed the Remote Pilot Knowledge Test, allowing Michigan Tech to safely and properly operate these platforms at each of the study sites. The project team provided MDOT with advice on UAV operations, following the guidance of the Michigan Attorney General. Similarly, an upcoming FAA sponsored program, the UAS Integration Pilot Program (IPP), will provide an opportunity for state, local, and tribal governments and private sector entities to work together to accelerate safe UAS operations. The IPP will promote open dialogue about operational concepts (i.e. night operation and flying over people and beyond line of sight).

7.2 Recommendations for Further Research

During two phases of research, the project team has demonstrated applications of UAV-enabled data collection and then developing the workflows to implement the applications into regular use to meet MDOT's needs for timely, high-resolution, safely collected, repeatable, and objective data for management, operations, and maintenance. A third phase is needed to make these data, applications, and workflows part of day-to-day usage for management and operations (planning, design, construction, operations, maintenance, asset management, etc.), especially under developing federal rules that are increasing the practical deployment of UAVs. A new phase would enable leveraging of these rapidly advancing technologies to meet operational data needs for connected vehicle corridor surveillance, construction site monitoring, asset management, traffic monitoring, geophysical evaluations, incident management, infrastructure inspection (e.g., bridges, roads, railways, etc.), confined space safety assessment, and pro-active monitoring of roadway corridors and bridges, among other applications. For corridor assessment, how UAVs can help replace or augment current systems operations sensor devices needs to be considered. In a funding-limited environment, UAVs have the potential to collect critical data more inexpensively and accurately, if data can be effectively used through timely and innovative processing, analysis, and sharing through decision support tools.

7.3 Potential Objectives for Further Research

1. Enable everyday usage of UAVs by MDOT to meet critical systems operations and maintenance data needs.
2. Work closely with MDOT staff to get UAV capabilities into their hands through efficient access to UAS-collected data, platforms, sensors, and software tools through integration with DUAP, operations, corridor asset management, bridge inspections, and other MDOT programs.
3. Have at least three MDOT Sections in one or more Regions using UAS as part of day-to-day operations by the end of a new project. This will involve implementation-focused case studies and technology transfer training sessions.
4. Investigate the application of hyperspectral remote sensing from UAS platform for transportation asset and bridge inspection and health monitoring. The hyperspectral cameras can provide an image from wavelengths as high as 2500 nm. The high spatial and spectral resolution of hyperspectral remote sensing offers the opportunity to detect the subtle changes in concrete or steel properties due to deterioration or corrosion. The hyperspectral sensing combined with the success of optical and thermal sensing (demonstrated in phase I & II) can potentially improve the early detection of structural defects to transportation structures such as bridges and retaining walls. The early detection of these defects can provide significant cost saving and improve the safety of the transportation corridor.
5. Recommend how to effectively work with the private sector to take advantage of rapidly developing sensors (lidar, thermal, optical, acoustic, chemical, etc.) and platforms (multi-rotor, fixed wing) to provide data, applications, and tools for operations and maintenance.
6. Leverage Phase I & II data collection systems, lessons learned, data storage needs, applications, etc. to provide a Phase III project focused on rapid implementation of developing UAV technologies.

This additional research would fully integrate the usage a UAS into the day-to-day operations of MDOT through the use of contractor supplied UAS based data and through MDOT staff utilizing UAS capabilities internally.

UAVs and the sensors that they can carry are continuing to develop rapidly. As compared to Phase 1 of this project, the number, types, and resolutions of platforms and sensors available for practical deployment have changed dramatically, as well has the number of people using UAVs for research and operational purposes. Similarly, the data processing and analysis software are also continuing to develop, with some companies now offering data processing in the cloud for faster data processing and easier data access. The full-impact and usefulness of these types of advancing technologies are developing rapidly, and MDOT should stay on the forefront of this trend. The project team thanks MDOT for the opportunity to continuing demonstrating UAV capabilities and looks forward to moving the results towards further implementation.

8. Bibliography

Comaniciu, D. and Meer, P., (2002). Mean shift: A robust approach toward feature space analysis. IEEE Transactions on pattern analysis and machine intelligence, 24(5), pp.603-619.

Brooks, C.N., Dobson, R.J., Banach, D.M., Dean, D.B., Oommen, T., Wolf, R.E., Havens, T.C., Ahlborn, T.M., Hart, B.E., Cook, S.J. and Clover, A., (2015). Evaluating the Use of Unmanned Aerial Vehicles for Transportation Purposes: Final Report, No. RC-1616. Michigan Technological University, 201 pgs. http://www.michigan.gov/mdot/0,4616,7-151-9622_11045_24249-353767--,00.html

Brooks, C., Dobson, R., Banach, D., Roussi, C., Colling, T., Lefler, V., Hart, B., Garbarino, J., Lawrence, J., White, B., Aden, S., Singh, C. (2016). Characterization of Unpaved Road Condition through the Use of Remote Sensing Project – Phase II Deliverable 8-D: Final Report. Michigan Tech Research Institute, 97 pgs. http://www.mtri.org/unpaved/media/doc/deliverable_Final_Report_Del8D_Unpaved_Roads_Phase_II_RevisedB.pdf

Brooks, C., Dobson, R., Banach, D. and Cook, S.J., (2017). Transportation Infrastructure Assessment through the Use of Unmanned Aerial Vehicles. TRB 2017 Annual Meeting Compendium of Papers (2017): 13pp. no. 17-05629. https://trid.trb.org/view/1439232

Dobson, R., Colling, T., Brooks, C., Roussi, C., Watkins, M. and Dean, D., (2014). Collecting decision support system data through remote sensing of unpaved roads. Transportation Research Record: Journal of the Transportation Research Board, (2433), pp.108-115.

Duda, R.O. and Hart, P.E., (1972). Use of the Hough transformation to detect lines and curves in pictures. Communications of the ACM, 15(1), pp.11-15.

Escobar-Wolf, R., Oommen, T., Brooks, C.N., Dobson, R.J. and Ahlborn, T.M., (2017). Unmanned Aerial Vehicle (UAV)-Based Assessment of Concrete Bridge Deck Delamination Using Thermal and Visible Camera Sensors: A Preliminary Analysis. Research in Nondestructive Evaluation, pp.1-16. http://www.tandfonline.com/doi/abs/10.1080/09349847.2017.1304597

Gucunski, N., Imani, A., Romero, F., Nazarian, S., Yuan, D., and Wiggenhauser, H. et al. (2012). Nondestructive Testing to Identify Concrete Bridge Deck Deterioration. http://dx.doi.org/10.17226/22771

Hartigan, J.A. and Wong, M.A., (1979). Algorithm AS 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1), pp.100-108.

Huang, Y., Adams, T., and Pincheira, J. (2004). Analysis of Life-Cycle Maintenance Strategies for Concrete Bridge Decks. Journal of Bridge Engineering, 9(3), 250-258. http://dx.doi.org/10.1061/(ASCE)1084-0702(2004)9:3(250)

Kelley, R. (2016). A process for systematic review of bridge deterioration rates. Michigan Department of Transportation. https://www.michigan.gov/documents/mdot/A_Process_for_Systematic_Review_of_Bridge_Deterioration_Rates_522422_7.pdf

MDOT - Bridge Management and Scoping web page. (2018). Michigan.gov. Retrieved 19 February 2018, from http://www.michigan.gov/mdot/0,4616,7-151-9625_70811_70814---,00.html

Omar, T., Nehdi, M., and Zayed, T. (2017). Integrated Condition Rating Model for Reinforced Concrete Bridge Decks. Journal of Performance of Constructed Facilities, 31(5), 04017090. http://dx.doi.org/10.1061/(ASCE)cf.1943-5509.0001084

Olsen, M.J. (2013). Guidelines for the use of mobile LIDAR in transportation applications (Vol. 748). Transportation Research Board, 208 pgs. http://onlinepubs.trb.org/onlinepubs/nchrp/nchrp_rpt_748.pdf

Otukei, J.R. and Blaschke, T. (2010). Land cover change assessment using decision trees, support vector machines and maximum likelihood classification algorithms. International Journal of Applied Earth Observation and Geoinformation, 12, pp.S27-S31.

Remer, D., and Nieto, A. (1995). A compendium and comparison of 25 project evaluation techniques. Part 1: Net present value and rate of return methods. International Journal of Production Economics, 42(1), 79-96. http://dx.doi.org/10.1016/0925-5273(95)00104-2

Resende, M.R., Bernucci, L.L.B. and Quintanilha, J.A. (2014). Monitoring the condition of roads pavement surfaces: proposal of methodology using hyperspectral images. Journal of Transport Literature, 8(2), pp.201-220.

Smith, D. and Heidemann, H.K. (2015). New Standard for New Era: Overview of the 2015 ASPRS Positional Accuracy Standards for Digital Geospatial Data. Photogrammetric Engineering & Remote Sensing, 81(3), pp.173-176.

Tatum, M., and Liu, J. (2017). Unmanned Aircraft System Applications in Construction. Procedia Engineering, *196*, 167-175. http://dx.doi.org/10.1016/j.proeng.2017.07.187

Winn, E., and Burgueño, R. (2013). Development and validation of deterioration models for concrete bridge decks. Michigan Department of Transportation.

Yehia, S., Abudayyeh, O., Nabulsi, S., and Abdelqader, I. (2007). Detection of Common Defects in Concrete Bridge Decks Using Nondestructive Evaluation Techniques. Journal of Bridge Engineering, 12(2), 215-225. http://dx.doi.org/10.1061/(ASCE)1084-0702(2007)12:2(215)

Zhao, S., K. Zhang, C. Brooks, D. Banach, and S. Aden. (2018). A Comprehensive Overview of Improving Traffic Flow Observability Using UAVs as Mobile Sensors. Transportation Research Board 2018 Annual Meeting Online 18, no. 4949: 16 pgs.

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

9. Appendices

9.1 List of Acronyms, Abbreviations and Symbols

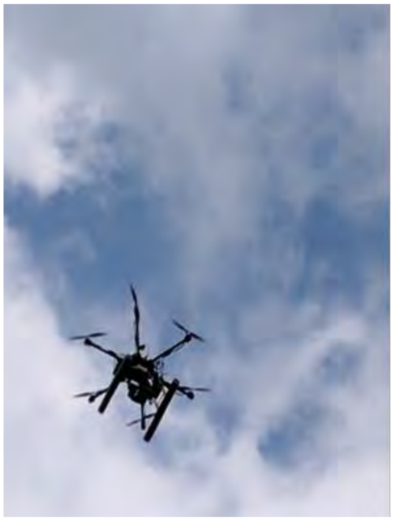
Above Ground Level (AGL)
American Society for Photogrammetry and Remote Sensing (ASPRS)
Bridge Management System (BMS)
Certificate of Authorization (COA)
Chain Dragging (CD)
Content Standard for Digital Geospatial Metadata (CSDGM)
Continuously Operating Reference Station (CORS)
Data Use, Analysis, and Process (DUAP)
Digital Elevation Model (DEM)
Digital Single-lens Reflex (DSLR)
Federal Aviation Administration (FAA)
Federal Geographic Data Committee (FGDC)
Global Navigation Satellite System (GNSS)
Global Positioning System (GPS)
Graphical User Interface (GUI)
Ground Sampling Distance (GSD)
Hammer Sounding (HS)
Hot Mix Asphalt (HMA)
Implementation Action Plan (IAP)
Inertial Measurement Unit (IMU)
Integration Pilot Program (IPP)
Intelligent Transportation System (ITS)
Light Detection and Ranging (LiDAR or lidar)
Michigan Department of Transportation (MDOT)
Michigan Tech Research Institute (MTRI)
National Bridge Inventory (NBI)
National Cooperative Highway Research Program (NCHRP)
Net Present Value (NPV)
Non-Destructive Testing (NDE)
Open Computer Vision (OCV)
Pilot in Command (PIC)
Real Time Kinematic (RTK)
Research Advisory Panel (RAP)
Rights-of-Way (ROW)
Root Mean Square Error (RMSE)
Structure from Motion (SfM)
Surveying Solutions Inc. (SSI)
Three Dimensional (3D)
Transportation Research Board (TRB)
Unmanned Aerial System (UAS)
Unmanned Aerial Vehicles (UAVs)
Visual Observer (VO)

9.2 UAVs and Sensors Operation Manual for MDOT Use

Bergen Hexacopter, Quad-8 and DJI Mavic User's Manual

Nikon D810, FLIR Vue Pro / Pro R, Velodyne LP-16 LiDAR

Operations Guide



Michigan Tech Research Institute

Michigan Technological University

9.2.1 Safety

9.2.1.1 Flight Safety Precautions

It is the responsibility of the user of any UAV (unmanned aerial vehicle) to follow all FAA operational guidelines and ensure safe operating conditions at all times. This includes the safety of the operator in addition to support personnel and bystanders. Some safety concerns to be mindful of include:

- Proximity to people and vehicles
- Proximity to powerlines, trees and other objects that may interfere with the UAV's flight path
- Preventing operations near airports and controlled airspace
- Ensuring the UAV remains within line of sight at all times
- Keeping the UAV below 400ft AGL (above ground level)
- Maintaining control of the aircraft at all times
- Avoiding flights in high winds, rain, near thunderstorms or in other adverse conditions
- Keeping all fingers, hands and limbs away from spinning rotors

9.2.1.2 LiPo Battery Safety

The Hexacopter uses four LiPo batteries which comprise a "Flight Set" while the Quad-8 uses two LiPo batteries to comprise a set. When not in use and being stored, the batteries need to be kept at a storage charge of 50-80% to retain performance and longevity. Batteries also need to be stored in an ammo can or equivalent storage container. If a battery fails, it produces a significant amount of heat which could cause nearby objects to ignite.

Batteries also need to be inspected to ensure that they are not damaged or "puffy". If a battery has a bloated appearance, it is not to be used. A bloated battery is caused by damage to the cell membrane, which induces outgassing. This is a good indicator that the battery could fail and/or rupture causing loss of control or damage to the UAV. Improper storage of batteries can impact the performance of the UAV and could result in a battery failure while in flight.

9.2.2 Bergen Hexacopter and Quad-8 operations

9.2.2.1. Preflight Checks

For safe operation of the Bergen Hexacopter and Quad-8, a preflight check must be performed to ensure it is in flight worthy condition. Preflight procedure:

1. Unfold the arms and tighten them in place. There are three thumbscrews for each of the folding arms that need to be tightened prior to flight. The motors must be oriented vertically with the propellers parallel to the ground.
2. Make sure the frame and arms are not damaged
3. Ensure that all the wire connections are securely fastened
4. Ensure that the GPS antenna is securely fastened and if it was removed for transportation, the red arrow on top must face the front of the UAV.

5. Check the propellers for damage. If cracks or chips are present, they must be replaced. Also, check that they are securely fastened to the motor.
6. Check each of the batteries to ensure they are fully charged and not damaged.
7. Attach the batteries and tightly fasten the Velcro strap so that the batteries do not move easily.
8. Turn on the controller then plug in the Hexacopter or Quad-8 batteries (Follow start up procedures in the Normal Flight Operations section) and allow it to lock onto GPS satellites. The LED Indicator will display progress on acquiring satellite lock as well as whether the main controller and IMU are functioning correctly. In GPS mode, the LED Indicator will quickly flash purple three times with a 1-second pause and then repeat when the Hexacopter or Quad-8 has full GPS lock and is ready to fly.
9. Perform a compass calibration if needed.

9.2.2.2 Performing a Compass Calibration

A compass calibration is needed (Figure A-1):

1. Before the UAV's very first flight and if the position of the GPS antenna has been moved
2. If the UAV has traveled at least 100 km (Distance determined by MTRI staff based on research) from the last takeoff location to the new location
3. The UAV has not been operated for an extended period of time
4. The UAV has been stored or packed near large metal objects
5. The LED indicator shows a compass error by flashing alternating orange and green lights
6. If the UAV appears to be drifting while it should be flying in a straight line

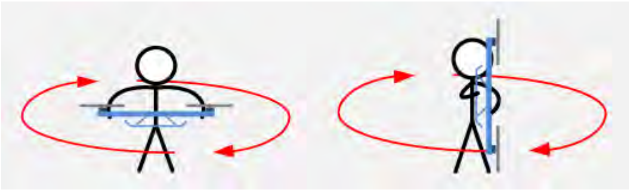
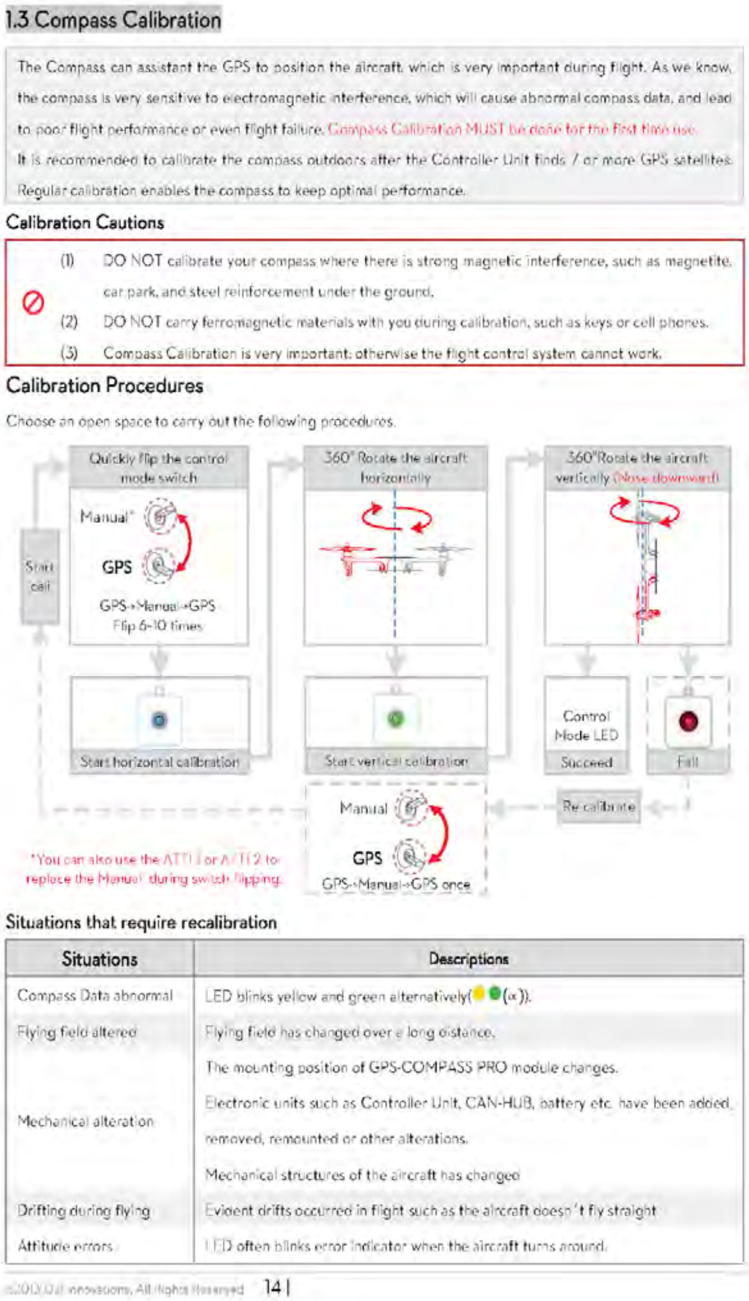


Figure A1: General UAV compass calibration procedure. (Source: http://wiki.dji.com)

Here are the steps to complete a compass calibration:

1. Quickly flip the Mode Controller switch on the remote control between Manual Mode and GPS Mode at least 10 times.
2. The LED Indicator light will turn solid blue. Hold the UAV level and spin one full rotation clockwise until the light turns green. Shown in Figure A2.
3. Flip the UAV so that the front is facing the ground and spin one full rotation clockwise until the light changes.
4. If the LED Indicator light turns white, the compass calibration was successful.
5. If the LED Indicator light quickly flashes red, the calibration failed and needs to be repeated.

Compass Calibration Diagram



1.3 Compass Calibration

The Compass can assistant the GPS to position the aircraft, which is very important during flight. As we know, the compass is very sensitive to electromagnetic interference, which will cause abnormal compass data, and lead to poor flight performance or even flight failure. Compass Calibration MUST be done for the first time use. It is recommended to calibrate the compass outdoors after the Controller Unit finds 7 or more GPS satellites. Regular calibration enables the compass to keep optimal performance.

Calibration Cautions

🚫

(1) DO NOT calibrate your compass where there is strong magnetic interference, such as magnetite, car park, and steel reinforcement under the ground.

(2) DO NOT carry ferromagnetic materials with you during calibration, such as keys or cell phones.

(3) Compass Calibration is very important; otherwise the flight control system cannot work.

Calibration Procedures

Choose an open space to carry out the following procedures.

Start cali

Quickly flip the control mode switch

Manual*

GPS

GPS→Manual→GPS
Flip 6-10 times

Start horizontal calibration

360° Rotate the aircraft horizontally

Start vertical calibration

360°Rotate the aircraft vertically (Nose downward)

Control Mode LED

Succeed

Fail

Re-calibrate

Manual

GPS

GPS→Manual→GPS once

*You can also use the ATTI 1 or ATTI 2 to replace the Manual during switch flipping.

Situations that require recalibration

Situations	Descriptions
Compass Data abnormal	LED blinks yellow and green alternatively(🟡🟢(∞)).
Flying field altered	Flying field has changed over a long distance.
Mechanical alteration	The mounting position of GPS-COMPASS PRO module changes. Electronic units such as Controller Unit, CAN-HUB, battery etc. have been added, removed, remounted or other alterations. Mechanical structures of the aircraft has changed
Drifting during flying	Evident drifts occurred in flight such as the aircraft doesn't fly straight
Attitude errors	LED often blinks error indicator when the aircraft turns around.

©2013 DJI Innovations. All Rights Reserved 14 |

Figure A2: Compass Calibration Instructions (source: dji.com)

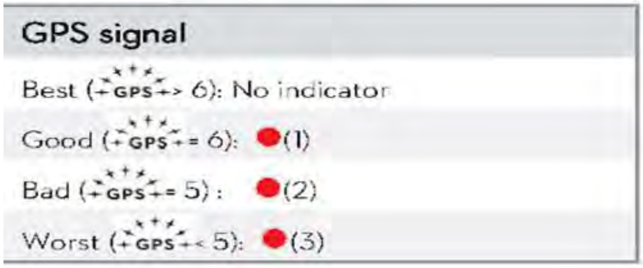
9.2.3 Normal Flight Operations-Bergen Hexacopter

Turning on the Hexacopter

1. On the remote control –

a. Make sure all toggle switches are up or back with the exception of the Mode selector switch. The Mode selector switch should be in GPS Mode which is in the down position.
b. The throttle is all stick down in the 0% position
c. The two side levers are up
d. The throttle trim should be all the way down
e. Turn on the remote control

2. On the Hexacopter
   a. Check battery voltage to ensure batteries are fully charged.
   b. Attach the batteries
   c. Plug in the negative leads (black wires) for all four.
   d. Make sure the UAV is on level ground and is in a clear open area before starting. It will assign its home position as "Home" based on where it first acquires GPS lock on startup.
   e. Plug in the positive lead (red wires) for the first battery.
   f. This will turn on the copter and the operator should hear four beeps in quick succession followed by a pause as it calibrates for level. If the calibration is successful, there will be a single beep and the LED Indicator will start normal operations. If there is no beep at the end, the calibration was unsuccessful. If this happens, unplug the positive lead and wait 5 seconds then power on again.
   g. Once successful, plug in all other leads.
   h. Check for full GPS lock which is at least 7 satellites (Figure A3). On the LED Indicator the Mode flash will be followed by at least one red flash if more satellites are needed. When there are no more red flashes, the UAV has full GPS lock and is ready to fly.



GPS signal	
Best (GPS > 6):	No indicator
Good (GPS = 6):	●(1)
Bad (GPS = 5) :	●(2)
Worst (GPS < 5):	●(3)

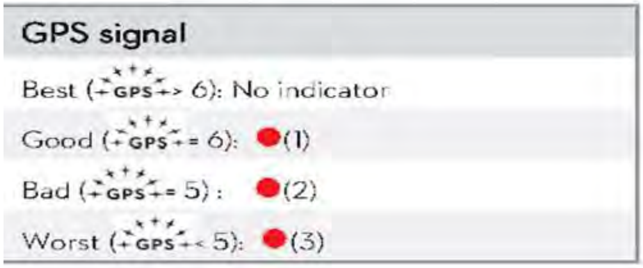
Figure A3: GPS signal key.

   i. Move both control sticks on the remote control down and inwards to start the motors. Once motors are started, allow the right stick to go to center. The left stick will remain down but centered.
   j. Slowly increase the throttle to at least 75% to spin up the motors for takeoff then bring the throttle down to 50% once the desired altitude has been obtained to level off.
   k. At any point during the flight, the left switch on the remote control can be flipped down to initiate a "Return to Home".

9.2.4 Normal Flight Operations-Bergen Quad-8

Turning on the Quad-8

1. On the remote control –
   a. Make sure all toggle switches are up or back with the exception of the Mode selector switch. The Mode selector switch should be in GPS Mode, which is in the down position.
   b. The throttle is all stick down in the 0% position
   c. The two side levers are up
   d. The throttle trim should be all the way down
   e. Turn on the remote control
2. On the Quad-8
   a. Check battery voltage to ensure batteries are fully charged.
   b. Attach the batteries
   c. Make sure the UAV is on level ground and is in a clear open area before starting. It will assign its home position as “Home” based on where it first acquires GPS lock on startup.
   d. Plug in one of the yellow battery leads.
   e. This will turn on the copter and the operator should hear four beeps in quick succession followed by a pause as it calibrates for level. If the calibration is successful, there will be a single beep and the LED Indicator will start normal operations. If there is no beep at the end, the calibration was unsuccessful. If this happens, unplug the yellow lead and wait 5 seconds then power on again.
   f. Once successful, plug in all other leads.
   g. Check for full GPS lock, which is at least seven satellites. On the LED Indicator, the Mode flash will be followed by at least one red flash if more satellites are needed. When there are no more red flashes, the UAV has full GPS lock and is ready to fly (Figure A4).



GPS signal
Best (GPS > 6): No indicator
Good (GPS = 6): ●(1)
Bad (GPS = 5) : ●(2)
Worst (GPS < 5): ●(3)

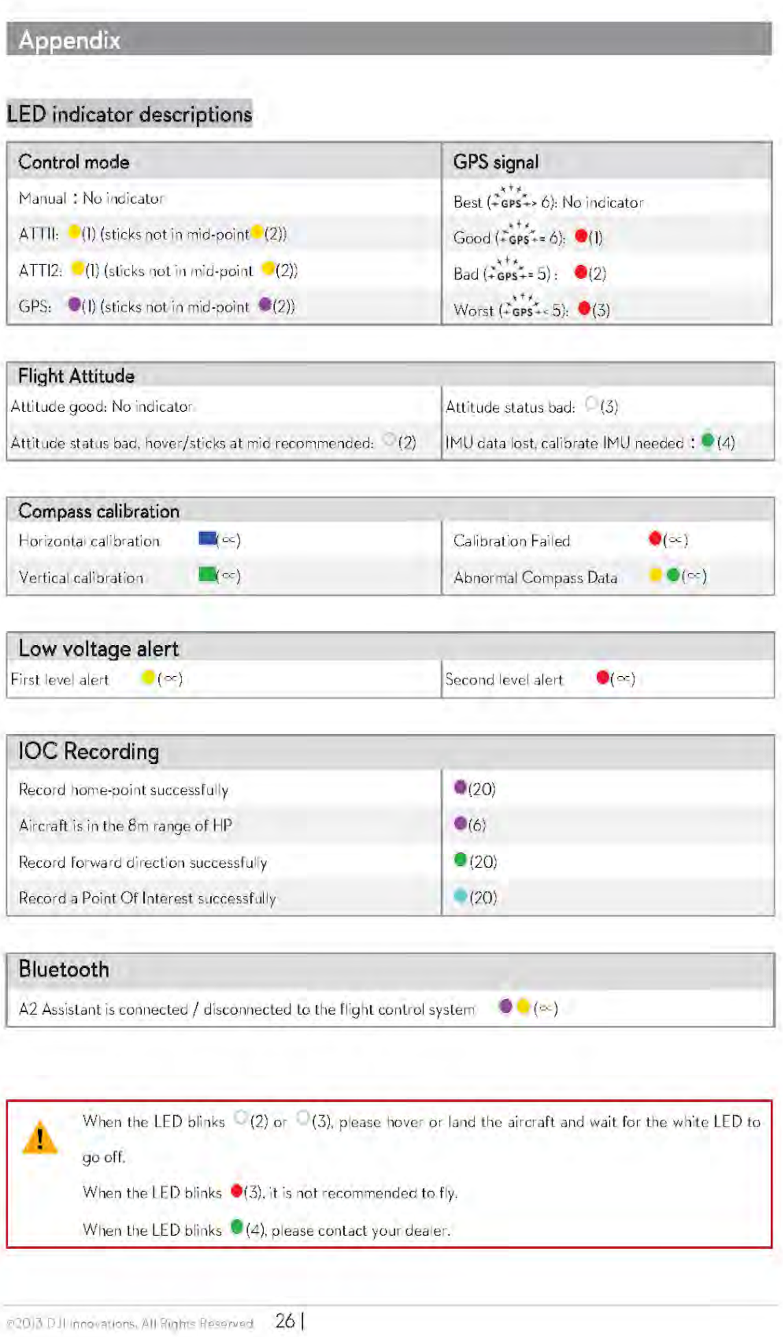
Figure A4: GPS signal key.

   h. Move both control sticks on the remote control down and inwards to start the motors. Once motors are started, allow the right stick to go to center. The left stick will remain down but centered.

i. Slowly increase the throttle to at least 75% to spin up the motors for takeoff then bring the throttle down to 50% once the desired altitude is obtained to level off.
j. At any point during the flight, the left switch on the remote control can be flipped down to initiate a “Return to Home”.

9.2.5 LED Indicator Description-DJI A2 Flight controller

See Figure A5 below.



Appendix

LED indicator descriptions

Control mode	GPS signal
Manual : No indicator	Best (GPS > 6): No indicator
ATTI1: 🟡(1) (sticks not in mid-point 🟡(2))	Good (GPS = 6): 🔴(1)
ATTI2: 🟡(1) (sticks not in mid-point 🟡(2))	Bad (GPS = 5) : 🔴(2)
GPS: 🟣(1) (sticks not in mid-point 🟣(2))	Worst (GPS < 5): 🔴(3)

Flight Attitude	
Attitude good: No indicator	Attitude status bad: ⚪(3)
Attitude status bad, hover/sticks at mid recommended: ⚪(2)	IMU data lost, calibrate IMU needed : 🟢(4)

Compass calibration			
Horizontal calibration	🟦(∞)	Calibration Failed	🔴(∞)
Vertical calibration	🟩(∞)	Abnormal Compass Data	🟡🟢(∞)

Low voltage alert			
First level alert	🟡(∞)	Second level alert	🔴(∞)

IOC Recording	
Record home-point successfully	🟣(20)
Aircraft is in the 8m range of HP	🟣(6)
Record forward direction successfully	🟢(20)
Record a Point Of Interest successfully	🔵(20)

Bluetooth	
A2 Assistant is connected / disconnected to the flight control system	🟣🟡(∞)

⚠ When the LED blinks ⚪(2) or ⚪(3), please hover or land the aircraft and wait for the white LED to go off.

When the LED blinks 🔴(3), it is not recommended to fly.

When the LED blinks 🟢(4), please contact your dealer.

©2013 DJI Innovations. All Rights Reserved. 26 |

Figure A5: General LED Codes for Bergen Hexacopter and Quad-8 (source: dji.com)

9.2.6 Nikon D810 Operation

9.2.6.1 Camera Setup

The Nikon D810 will be operated in manual mode. This means that all of the camera and lens settings (shutter speed, aperture, ISO, and focus) will be set manually prior to mounting it to the Bergen Hexacopter. The camera settings, shutter speed, aperture and ISO are all dependent of each other. Therefore, adjusting one will impact the others. Figure A6 shows how these settings are connected.

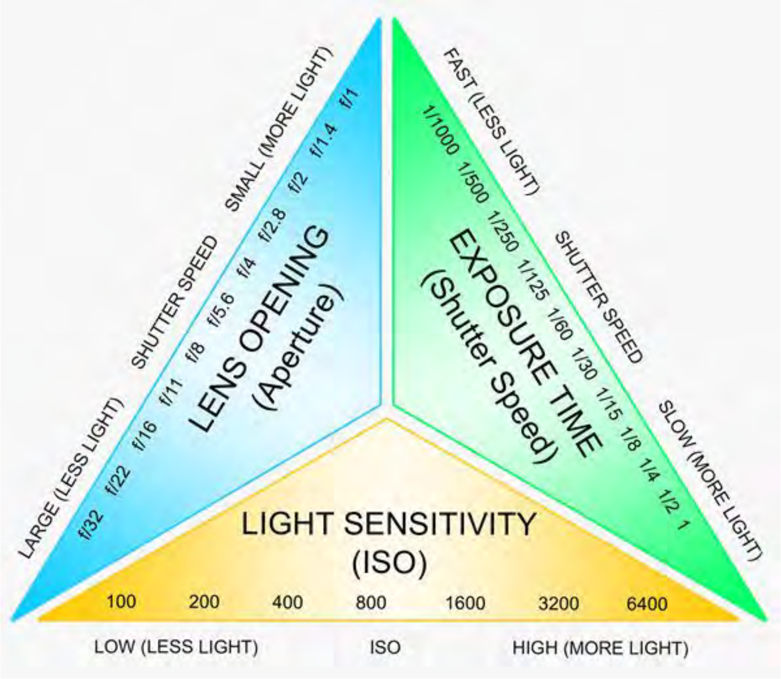


Figure A6: Relationship of camera settings to each other and relative impact on light exposing the sensor.

Shutter speed should be set first since this parameter is critical for capturing sharp imagery from a drone that's constantly moving. As shown in Figure A7, setting the shutter speed too low will cause motion blur of the imagery. Shutter speed is the amount of time the camera sensor is exposed to light and is measured in seconds. Faster shutter speeds reduce motion blur because the sensor is exposed for a shorter amount of time, but less light is allowed into the sensor. Slower shutter speeds allow for more light to the sensor, but it would allow for more movement to occur while the shutter is open. For a standard 3D photogrammetry collect, it is recommended for the shutter speed to be at least 1/640 sec and the Bergen Hexacopter to be flown at a maximum of 2 m/s (4.5 mph) from a minimum altitude of 100 ft (30.5 m). Shutter speed will have to be increased if the drone is operating while gusting winds are over 10 mph as the turbulence will cause the drone to make sudden movements and cause motion blur. Figure A8 shows an example of how shutter speed impacts the brightness of the picture.

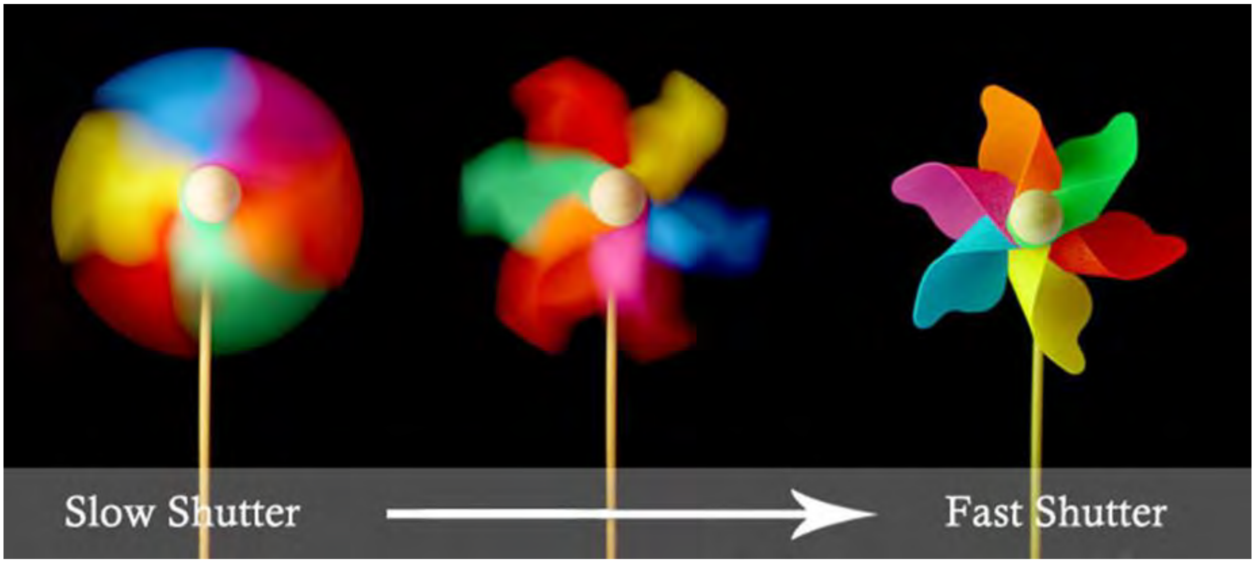


Figure A7: An example of the relationship between shutter speed and motion blur. The wheel is spinning at the same speed in all three examples, but the shutter speed is increased from left to right. Motion blur is present in the left and middle examples.

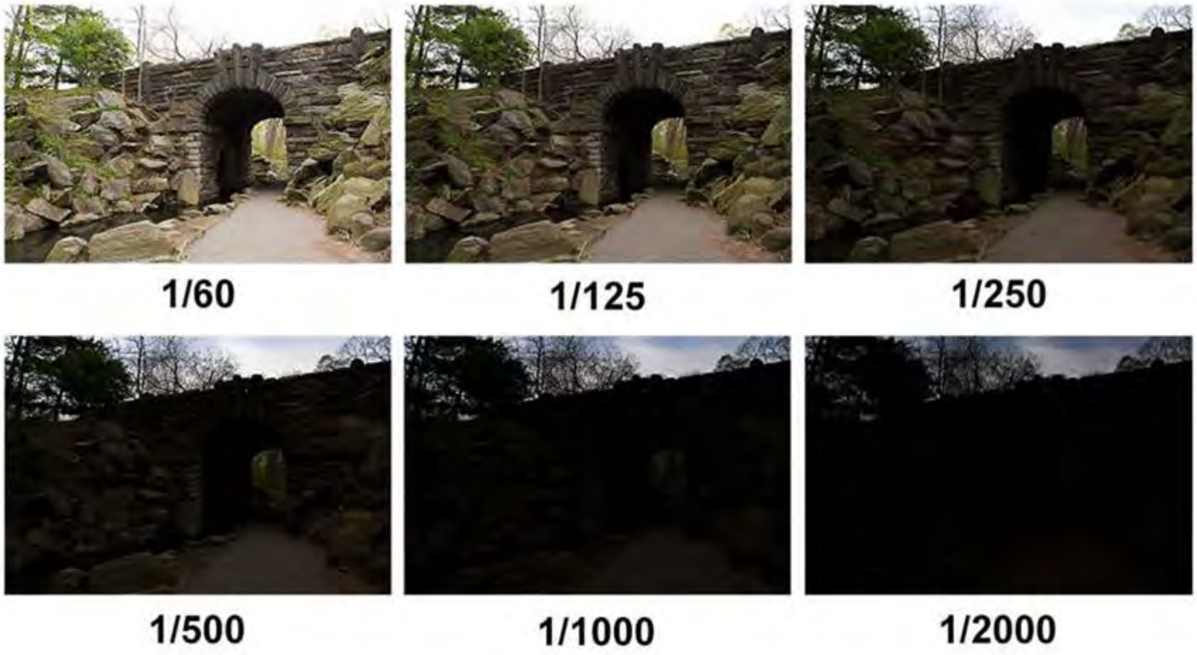


Figure A8: Increasing the shutter speed reduces the amount of light allowed to expose the sensor.

The second parameter to set is aperture. This is measured using f-stops and is related to how far the lens iris opens when a picture is captured. A lower f-stop refers to a larger iris opening and therefore more light can pass through the lens (Figure A9). For sunny days a f-stop of 8 or 10 may be suitable and cloudy days an f-stop of 4 or 5 may be more appropriate.

Changing the aperture not only controls the amount of light passing through the lens, but it also impacts the depth of field.

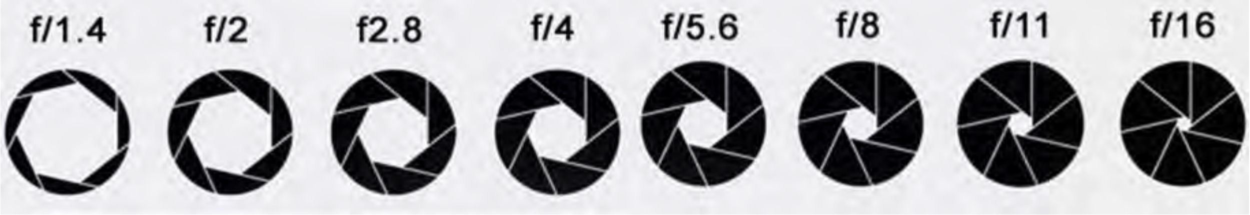


Figure A9: The relationship between f-stop and iris opening.

The depth of field refers to the area that is in focus when a picture is taken. Figure A10 shows an example of depth of field as it relates to distance from the camera. Small f-stops decrease the distance which is in focus, while larger f-stops increase the depth of field. When flying the Nikon D810 on a drone, the focus should be set for objects at the altitude the drone will be flying. If flying at 100 ft (30.5 m), the camera should also be focused at that distance. The depth of field is bounded by the near and far limits which describe the distances from the camera between which everything is in focus. Figure A11 shows the depth of field limits for a 100 ft (30.5 m) flying altitude for all aperture settings.

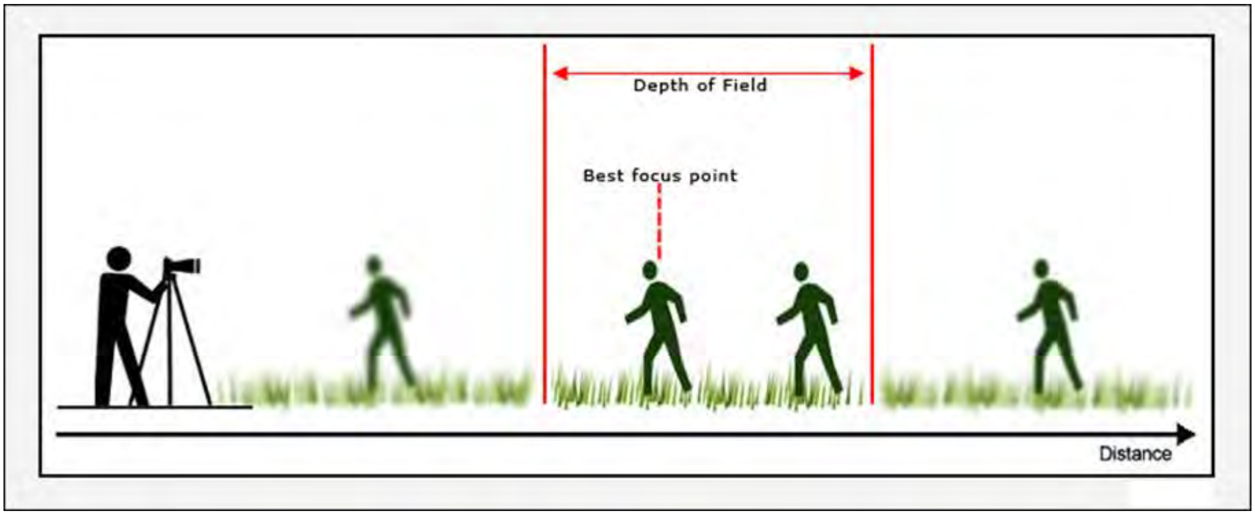


Figure A10: Everything that falls within the depth of field is in focus while everything outside of it will be blurred.

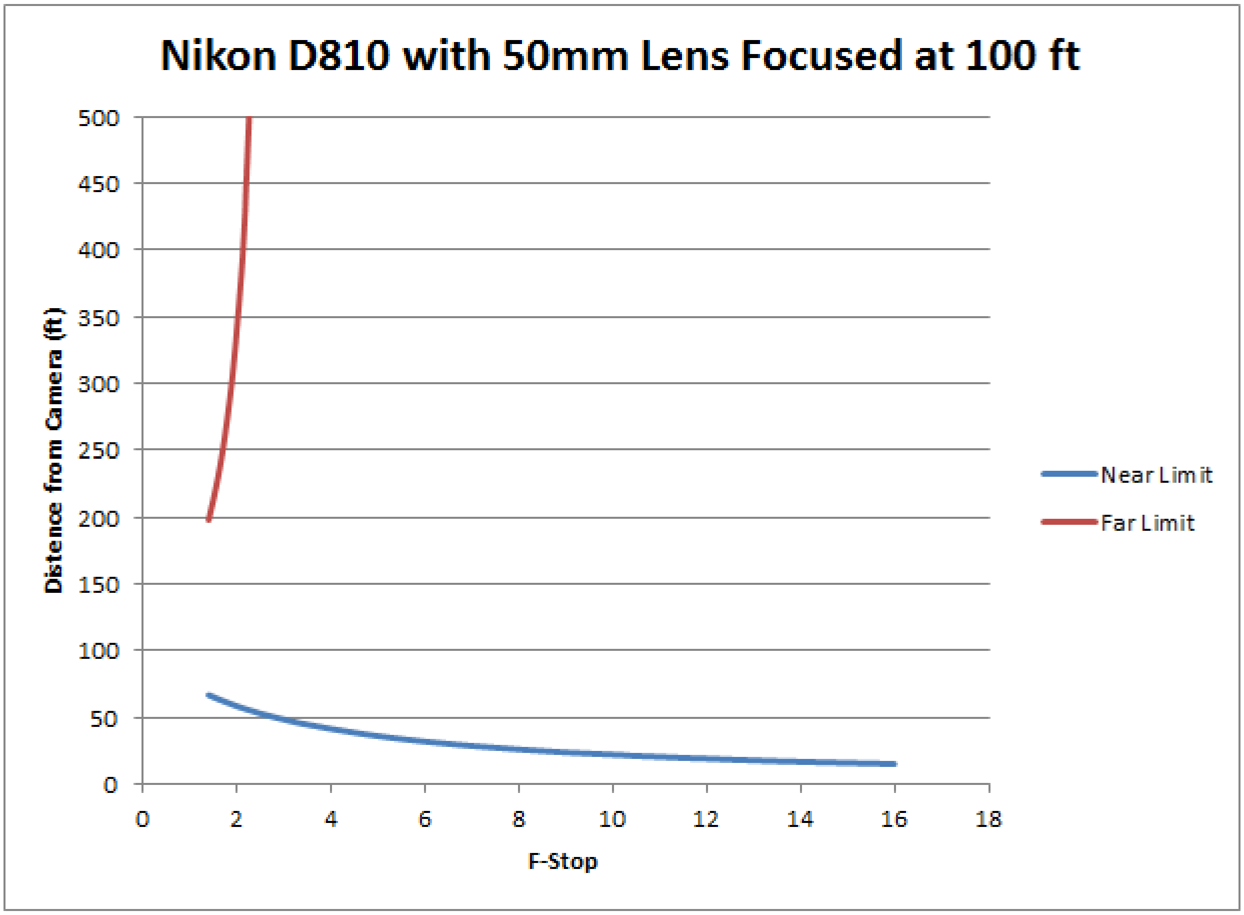


Figure A11: Example of the the near and far limits of the depth of field for a Nikon D810 with a 50mm prime lens focused to 100 ft (30.5 m). Everything between the Near and Far Limits will be in focus.

ISO is a measure of how sensitive the camera sensor is to incoming light and is the last parameter to be set. A Standard ISO setting for drone based collects with the Nikon D810 is 800. Increasing the ISO will increase the sensor's sensitivity to incoming light and will enable the user to set the desired shutter speed and f-stop to prevent motion blur and to maintain the desired depth of field for the collect. The Nikon D810 can be set with an ISO up to 12,800, but there is a disadvantage to increasing the ISO. As shown in Figure A12, increasing the ISO reduces the sharpness of the resulting image. Preserving images sharpness is essential for 3D photogrammetry applications.

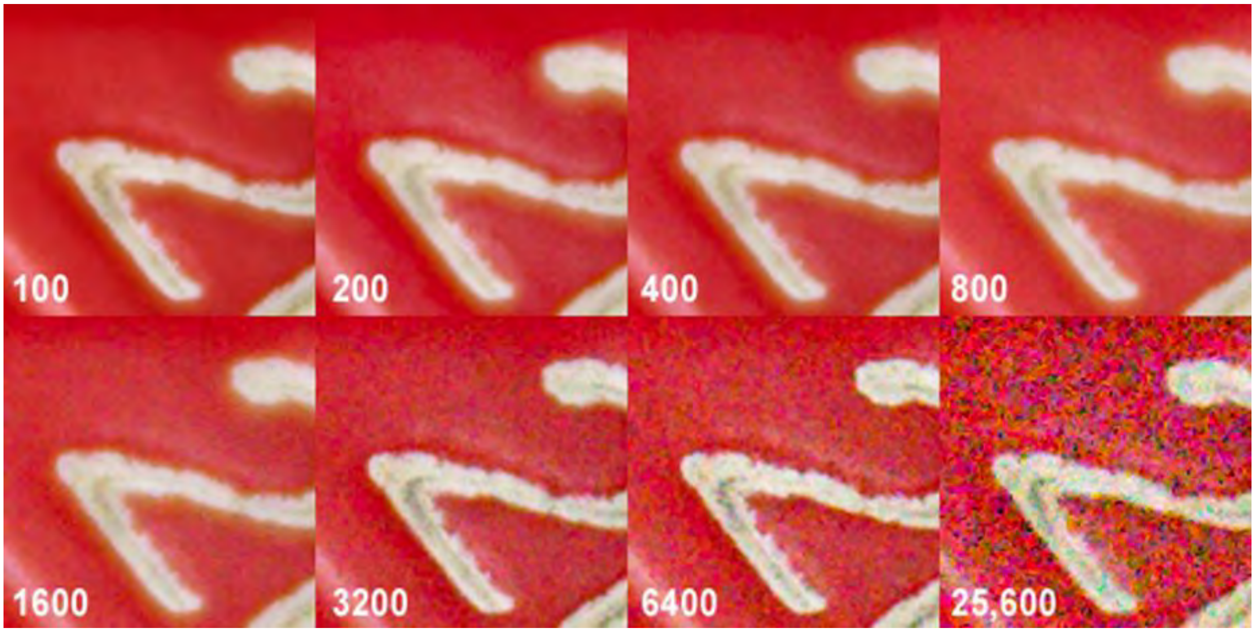


Figure A12: The relationship between ISO setting and image sharpness.

9.2.6.2 Procedure for setting up and collecting data:

1. Focus the lens to the altitude that the drone will fly. Switch the lens to "M/A" to autofocus, then switch back to "M" for manual after. This will prevent the camera from refocusing the lens to tree tops or other objects not related to the pavement surface.
2. Set Shutter Speed
   a. Standard for winds under 10 mph, flying altitude of 100 ft, and flying at 2 m/s (4.5 mph) is 1/640 sec
3. Set aperture
   a. F-Stop 2 - 4 for cloud conditions
   b. F-Stop 7 - 12 for sunny conditions
4. Set ISO
   a. Standard is ISO 800
5. Mount camera to the drone
6. Start "Interval Timer Shooting"
   a. Menu -> Shooting Menu -> Interval Timer Shooting -> Start
   b. Ensure that interval is set to 1 and the No. of intervals is set to 9,000

9.2.6.3 Processing Nikon D810 Imagery

The main processing tool used for processing Nikon D810 imagery is Agisoft Photoscan. This program imports the image set and then creates a 3D point cloud, Mesh, DEM, and Orthomosaic from the given images.

9.2.6.4 Agisoft Photoscan

To processes images through Agisoft Photoscan, first import the images into the program. Before proceeding further, it is important to scan for any images that may have been taken with motion blur or appear visually inconsistent with the other images (Figure A13). You need to remove these images before processing. Otherwise, it can cause a loss in resolution in addition to artifacts in the final output.

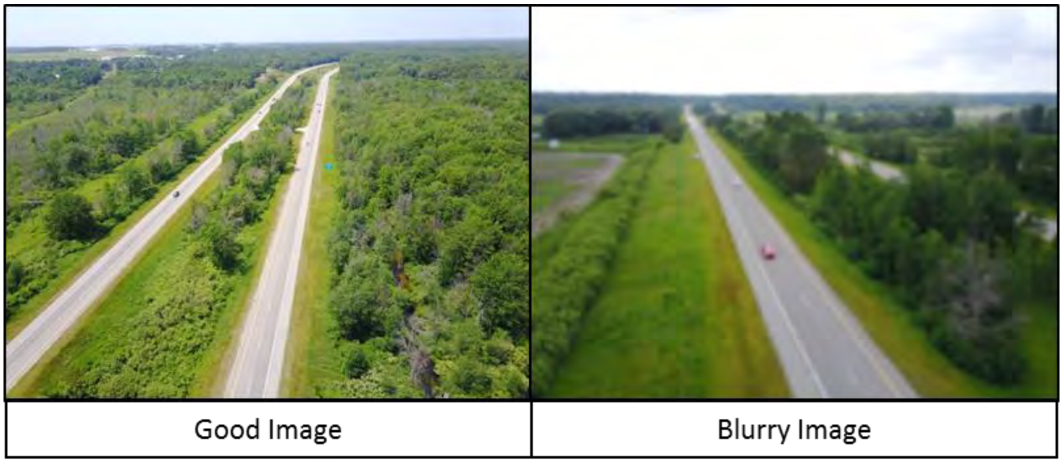


Figure A13: Example of good (left) and blurry (right) imagery.

After verifying the quality of the images you have selected, next press the align photos button. This will cause the program to build a sparse point cloud model using the images you have provided. If successful, the next step is to build the dense cloud. If both of these are successful, you will then build a mesh. It is important on this step to set the 3D mesh generation method to height field. Once you have created a mesh, you can then build a DEM and orthomosaic. Higher mesh, sparse and dense cloud settings will create DEMs and Orthomosaics that have higher resolutions, however processing time will be greatly increased at higher settings. In addition, if your images do not have geotagging and if you have not taken ground control points, you will be unable to create a DEM with accurate height values or take distance measurements off the orthomosaic. Therefore, including valid GPS information with your imagery is crucial for accurate results.

9.2.7 DJI Mavic Pro Operation

9.2.7.1 Preflight Checks and Initial Setup

The following checks should be performed before turning on or operating the DJI Mavic Pro.

1. Ensure the battery for the UAV and Remote are fully charged. If the battery power is too low, the UAV will not allow you to takeoff.
2. Inspect each propeller for damage. If there is damage of any kind to the rotors, remove that rotor and install a new one. Also, ensure that each propeller is installed in the correct location. The rotors with white circles in their centers should be placed on the motors

with white markings on them. Improper installation of the props will make it impossible to successfully takeoff.

3. Ensure the gimbal cover and gimbal control lock are removed from the UAV.
4. Plug in your phone to the remote and open the DJI Go 4 application.
5.

9.2.7.2 Normal Operation

1. To turn on the remote, press the power button once quickly and then hold until the remote turns on. **Always turn on the remote before the UAV.**
2. Next, ensure the DJI Go 4 application is running. Once running, you are now ready to turn on the UAV.
3. Press the power button on the UAV once quickly and then hold it down until the UAV emits a tone. Once on, the gimbal will move through a startup routine and the propellers will momentarily 'twitch'.
4. The remote and DJI Go app should now report that the UAV is connected. Press Go Fly on your phone screen.
5. In the top left of your screen, you may see a red or yellow banner indicating a compass error, magnetic interference, or other problem.
6. If there is a compass error, press the red banner with the message on it and then select calibrate under the compass menu. It will then walk you through the steps required to calibrate the compass.
7. If it indicates that there is magnetic interference, move the UAV to a new location away from metal objects. Do not takeoff if there is magnetic interference as the UAV may begin to drift of loose GPS connection after takeoff due to the interference.
8. If you receive any other type of error message, press on the message and follow on screen prompts to correct the problem.
9. On the top of your phone screen, centered, but slightly to the right, there will be an indicator displaying the number of satellites the UAV is currently tracking. Wait until the UAV has received at least seven satellites. Once this occurs, the words Ready to go GPS Mode should appear on your remote's screen. Once you see this message, the UAV is ready for takeoff.
10. Press the two control sticks together and down, this will unlock the motors, the propellers will begin to rotate but the UAV will not takeoff. If you do not takeoff in several seconds, the rotors will stop spinning.
11. To takeoff, press up slowly on the left control stick.
12. To land the aircraft, press down slowly on the left control stick. Once the UAV has landed, the rotors will automatically stop spinning.

9.2.7.3 Processing Mavic Pro Imagery

The main program utilized for processing DJI Mavic imagery is Agisoft Photoscan. The steps used to process this imagery are the same as those stated previously.

9.2.8 Flir Vue Pro and Flir Vue Pro R Operation

9.2.8.1 Powering the Camera

In order to operate the Flir Vue Pro or Flir Vue Pro R, an external battery pack must be connected to the mini USB port located on the side of the camera. Once this has been done, the camera will automatically turn on. In addition, you must also insert a microSD card into the memory card slot to record images and video.

9.2.8.2 Connecting the camera to the Flir Vue Pro App

To make changes to camera settings, you must download the Flir Vue Pro app on your phone or tablet. To connect the camera to the app, you must first plug in power to the camera. After about 25 seconds, the Bluetooth button will begin to glow blue. At this point, the Bluetooth has been turned on in the camera and you can now connect to your phone through the Flir Vue Pro App (remember to also turn Bluetooth on in your phone settings to connect with the camera). If at any time the blue light turns off on the camera, you can simply repress the Bluetooth button to re-activate Bluetooth on the camera. Keep in mind the camera Bluetooth will automatically turn off in not being used or if no inputs have been made through the app in several minutes.

Inside the app, you can adjust the camera settings to record video or take pictures. You can also choose the color ramp. We usually prefer to use the Fusion color ramp. To setup the camera to continuously take images, you must select the time-lapse setting and then set the period to one photo per second. Once setup, you can either close the app and start collecting data by pressing the record button (located below the Bluetooth button on the camera), or start collecting data through the app.

9.2.9 Processing Thermal Imagery

There are several programs that can be used to process thermal data. These programs include Adobe Photoshop, Agisoft Photoscan, ArcMap and Microsoft Image Composite Editor. It is usually best to start with Image Composite Editor before trying other methods. Before processing any of your data, it is extremely important to remove all blurry images from your image set. Failing to perform this initial step will make alignment much more difficult and will introduce blur into your outputs.

9.2.9.1 Adobe Photoshop

Photoshop can be used to both perform a lens correction as well as create orthomosaic images. In some cases, the imagery from the thermal camera may appear to have barrel distortion. To fix this, load one of the photos into Photoshop and then go to Filter->Lens Correction. Once this is done, a new window will appear allowing you to manually adjust one of the images until the distortion is eliminated. One you have fixed the photo, you can save those settings and run your entire image set through the batch process command within the program.

Because some of the photos may have been taken at different angles and altitudes, it may be necessary to adjust some of the photos differently than others. Once finished, you can then either use Photoshop, or one of the other programs listed to create an orthomosaic.

In order to create an orthomosaic within Photoshop, you must use the photomerge function. This function will import the desired images and then automatically align them based on the preset capture motion settings within the program. Often times, running subsets of your dataset will yield the best results. Once you have exported the orthomosaic, you must then use ArcMap to georeference the finalized orthomosaic to its actual location.

9.2.9.2 Agisoft Photoscan

In order to create orthomosaics, it is extremely helpful to take GPS ground control points that will be visible in the thermal imagery. These points will be helpful both for image alignment, in addition to georeferencing the imagery. To run thermal images through Agisoft, follow the program workflow to obtain your output. This program is specifically designed for optical imagery therefore, thermal imagery may or may not output correctly depending on the dataset. If you do not provide Agisoft with ground control points, you need to georeference the output imagery using ArcMap.

9.2.9.3 Microsoft Image Composite Editor

Using image composite editor will allow you to merge thermal images together. To run this program, select New Panorama and then insert your desired images (make sure the program sorts the images in the order they will aligned). From there, simply press next at the top of the screen and then follow the workflow to crop and export the merged photos. If the photos do not align correctly, try changing the camera motion option or try processing a subset of your images. Once successful, you will need to export the finished orthomosaic and use ArcMap to georeference it to the appropriate location. If the program cannot successfully merge your images, you may need to use one of the other listed programs to create the orthomosaic.

9.2.9.4 ArcMap

If aligning the thermal images through other methods was unsuccessful, you may be able to manually align and merge your images using the georeferencing toolbar within ArcMap. To do this, insert any ground control points you may have taken in addition to any previous optical or thermal imagery into the document. Adding a basemap layer may also be helpful. Next, add control points to each thermal image to make it align with your other imagery sources. Continue to align each subsequent image using your available imagery and previously placed thermal imagery. Once you have enough overlapping images, you can merge them together using the Mosaic to New Raster command. This function will input your georeferenced images and output a single georeferenced orthophoto.

9.2.10 Radiometric Calibration Flir Vue Pro R Imagery

In addition to creating orthomosaic imagery, you can also obtain imagery with per-pixel temperature values using the Flir Vue Pro R camera. This camera records the necessary conversion information and metadata in the radiometric jpegs it creates. To extract this data, you need to run the appropriate script to output the images in the desired temperature units. Remember, you can only extract per-pixel temperature values from the Flir Vue Pro R camera and not the Flir Vue Pro camera.

9.2.11 Velodyne LiDAR Puck Operation

9.2.11.1 Mounting the LiDAR unit to the UAV

Figure A14 below show how to properly mount the LiDAR unit to the Bergen Quad-8.

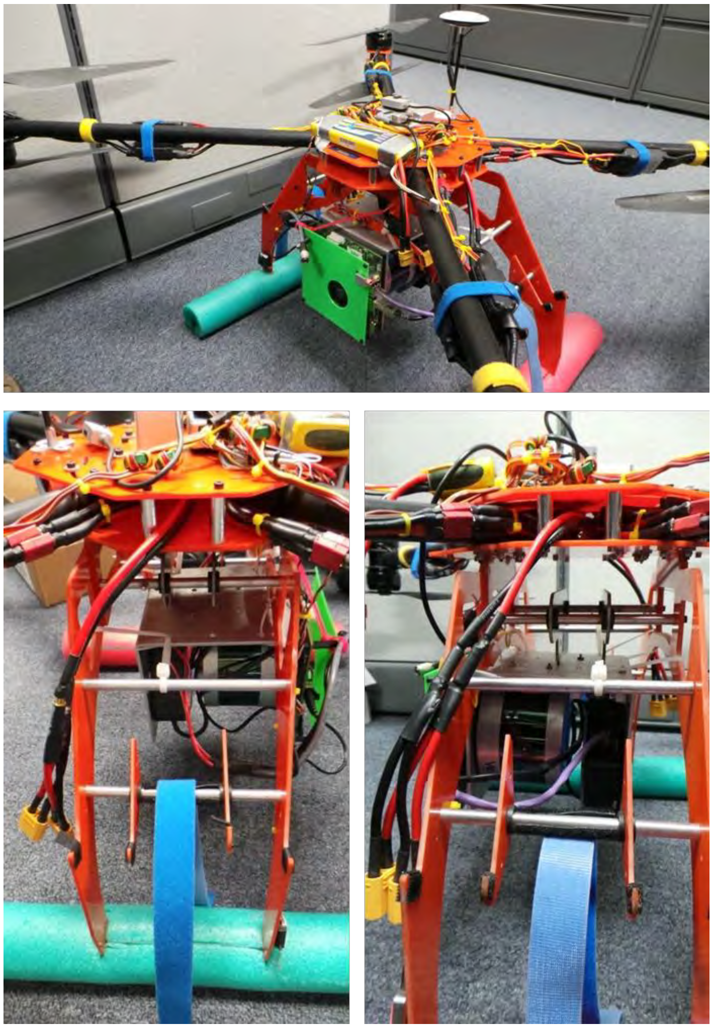


Figure A14: Bergen Q-8 with lidar attachment.

9.2.11.2 Processing LiDAR data

The lidar unit creates a point cloud measuring distances to its surroundings in .bag and .log files. The .bag files can be viewed as videos that show the generated LiDAR point cloud as the unit moves through the collection site. In order to create a raster output from this information, you must use GPS data to indicate the LiDAR unit's location throughout the collect. This information can be obtained in two different ways.

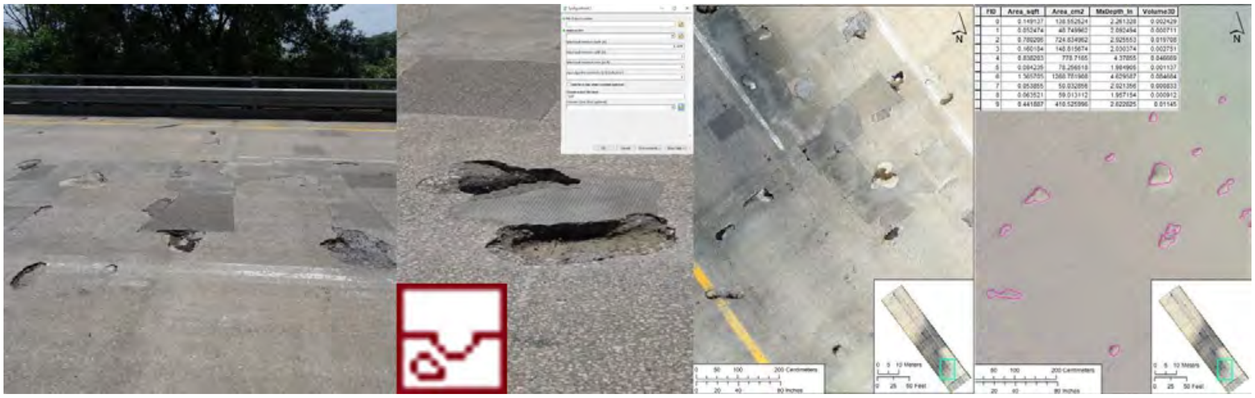
The first option is to simultaneously fly a GPS unit with the LiDAR unit during the collect. Once the GPS clock and LiDAR clock have been synchronized, you should be able to derive the location data from the GPS and apply it to the raw LiDAR points to create an output.

It is also possible to obtain GPS data from Agisoft Photoscan. To do this, you must create an orthomosaic of the study site using a standard RGB camera. From there, it is possible to extract the GPS information and apply it to the LiDAR data. This method will allow you to create an output without simultaneously flying the GPS and LiDAR unit.

9.3 “Spallgorithm” Manual

Spallgorithm 4.2 ArcGIS Desktop Tool

Operations Guide



FID	Area_sqft	Area_cm2	MxDepth_In	Volume3D
0	0.149137	138.552524	2.261328	0.002429
1	0.052474	48.749962	2.092494	0.000711
2	0.780205	724.834962	2.925553	0.019708
3	0.160184	148.815674	2.030374	0.002751
4	0.838203	778.7165	4.37855	0.046669
5	0.084235	78.256518	1.984905	0.001137
6	1.565705	1560.781968	4.629587	0.054684
7	0.053855	50.032856	2.021356	0.000833
8	0.063521	59.013112	1.957154	0.000912
9	0.441887	410.525996	2.022025	0.01145

Michigan Tech Research Institute

Michigan Technological University

9.3.1 Overview

The Spallgorithm is a arcpy script tool designed to analyze digital elevation models of road surfaces and delineate spalls. The tool is designed to be integrated into the ArcGIS Desktop environment. The tool is delivered as a zipped folder and comes with example datasets. The Spallgorithm tool comes within a custom ArcToolbox where other similar tools may be added.

9.3.2 ArcGIS Desktop Installation (see Figures A15 – A16)

The first step is to unzip the folder to a directory where the script will run from. This can be a location on either a network or local drive. Right click on the zipped folder ("Spallgorith.7z") and use select 7zip->Extract files. Choose your directory by pressing the … box in the resulting window and click OK. While the file extraction is occurring open a new instance of ArcMap.

9.3.2.1 Adding the toolbox

In ArcMap open the ArcToolbox by clicking on the icon or by clicking on Geoprocessing->ArcToolbox. When the tool box opens, right click on ArcToolbox at the top and select Add ToolBox. In the Add Toolbox window navigate to the directory where the files were extracted. The toolbox is located in the folder \Spallgorithm\ToolFiles\. Click on Bridge Evalution Tools.tbx and press open. You will see the toolbox added to the existing toolboxes. Double click on the toolbox and you will see Spallgorithm4.2. Before running the tool, the path to where the tool script exists will need to be updated. Right click on the Spallgorithm4.2 and select Properties. Click on the source tab and the folder icon . Navigate to the same location as the toolbox is found \Spallgorithm\ToolFiles\ and select SpallApp4_2.py and click Open and then OK in the Properties window.

9.3.2.2 Adding to the toolbar

From the dropdown menu click Customize->Customize Mode. In the Customize window click New… and enter a name for the new toolbar (e.g. Bridge Tools). The new toolbar will appear in the list of toolbars already checked. Note where the toolbar appears among the other toolbars. It will be a blank toolbar. Next click on the Commands tab in Customize window. Third from the bottom in the list of Categories will be [Geoprocessing Tools]. Highlight [Geoprocessing Tools] and select Add Tools… Again navigate into the Bridge Evaluation Tools toolbox, highlight Spallgorithm4.2 and click Add. Click and drag Spallgorithm4.2 from the Commands into the new Bridge Tools toolbar. While the Customize window is still open right click on the scripttool icon that was just dropped in the toolbar and select Browse for image… Navigate back to \Spallgorithm\ToolFiles\ and select Logo.png and click Open. The icon will be updated and the Customize window may be closed.

9.3.2.3 Saving toolbar customizations

Because the tool that has been added is a custom tool ArcMap will not save it in the normal template that loads each time ArcMap is started. The simpleist option is to save the changes to the document. This can be done by opening the customize window again, from the dropdown menu Customize->Customize Mode and clicking the Options tab. Check Save all customizations to the document and click OK on the pop up window. The current document can then be saved and will retain the customizations made. Click File->Save and choose a directory where you would like the document (*.mxd) to be saved. Each time the document is opened the toolbar and toolbox will have the Spallgorithm already loaded. If it is desired to keep this customization to the normal template users can follow the directions here: http://desktop.arcgis.com/en/arcmap/latest/get-started/customizing-the-ui/fundamentals-of-saving-your-customizations.htm

9.3.3 Running the tool
(See Figures A17 – A26)

9.3.3.1 Required Inputs

The only required input for the Spallgorithm is a high resolution digital elevation model (DEM) of a road surface in raster format, preferably TIFF. The DEM must be defined in linear units of feet. The State Plane coordinate system is recommended, specifically for Southern Michigan the NAD 1983 (2011) International Feet State Plane Michigan South coordinate system; FIPS 2113, WKID EPSG: 6499. However, the performance of the tool can be increased with the use of a shapefile which covers only the sections of the bridge to be analyzed. This is useful for the exclusion of bridge joints and edge effects which may lead subpar results. The addition of this exclusion file is not required to make the tool run. See below for instructions on making an exclusion zone shapefile.

9.3.3.2 User defined variables

The tool has a total of ten parameters to be specified. 1) File output location, 2) Select a DEM, 3) Select spall minimum depth, 4) width, 5) area, 6) algorithm sensitivity, 7) adding the output to the map, 8) the output file name, 9) selection of exclusion zone and 10) the option of creating a hillshade layer. The option of adding the output to the map and the selection of the exclusion zone are optional parameters. The others are required, however most come with default values.

The file output location specifies the folder where the output will be saved to. Both a spall feature class layer and a hillshade model will be produced. Select a DEM allows the user to choose a DEM layer to apply the tool to. The DEM can be in the table of contents of the current ArcMap session or selected from the folder icon. Specifying the spall minimum depth tells the tool the height deviation from the surrounding to classify as a potential spall. The default value is

3/16”. The spall minimum width specifies a value to the tool to exclude any spalls which are smaller in diameter than the defined value. The default value is 3”. The spall maximum area excludes and identified spalls that are greater than the defined value. The default value is 4 sq ft. The algorithm sensitivity is only meant to be used on DEM’s that are considered to be noisy. That is when the DEM contains exceptionally variable values across what is actually a smooth surface. The algorithm sensitivity tells the tool how many standard deviations away from the norm of the area in which a spall is to be considered a spall. Increasing the value will generally reduce the number of false positives, but may also result in true negatives. Checking the Add file to map box is optional. When the box is checked the spall layer will be added to the map document along with the hillshade if desired. Regardless of whether the box is checked the layers will be saved in the folder specified. Choosing the output filename allows the user to specify the file name that the shapefile (*.shp) will be saved as. The exclusion zone is optional and allows users to select a shapefile that covers the parts of the DEM to be analyzed. The exclusion zone layer can be in the table of contents of the current ArcMap session or selected from the folder icon. Selecting the Save Hillshade box will save a hillshade layer derived from the DEM with the same name of the spall layer plus “_hillshade.tif”.

9.3.3.3 The Workspace

ESRI geoprocessing tools often create several intermediary files that delete after the output is saved. The spallgorithm tool operates in a similar manner. When the tool is running this workspace where this occurs is in the ArcGIS scratch folder, which is usually located in the current users Documents\ArGIS folder. The scratch folder self-cleans as geoprocessing tasks proceed.

9.3.3.4 Creating an exclusion layer
(See Figures A27 – A34)

The aim of the mask layer is to define the zones which are to be included in the analysis. The sides of bridge decks often contain debris and artifacts from the railing that make the DEM less reliable. Excluding these zones generally improves performance. Using the orthomosaic raster was used to create the DEM is the best reference to use in creating a mask. Open the *.mxd document and add the orthophoto of the bridge to be analyzed. Next open ArcCatalog within ArcMap by pressing the ArcCatalog button or from the dropdown menu Windows->Catalog. Within ArcCatalog navigate to where the bridge DEM is, right click on the folder and select New->Shapefile. Type a name for the shapefile, select Polygon for the Feature type, and select the same spatial reference as the DEM is in (likely the State Plane Michigan South) and press OK. The empty shapefile will be added to the table of contents. From the editor toolbar Editor ▾ click on the dropdown menu and select Start editing. Again click the dropdown menu and select Editing Windows-> Create features. In the Create Features window click on the shapefile and in the Construction tools below click on polygon. To create the mask left click to create vertices which will form a polygon. After completing a polygon right click to bring up the menu. If multiple polygons are to be used on the bridge choose Finish part and proceed to create the next polygon. On the last polygon right click and choose Finish sketch. Go back to the editor dropdown menu and select Stop Editing and choose to save your edits. The exclusion layer has been created and may be used as a Spallgorithm input.

9.3.3.5 Understanding the output

The Spallgorithm generates a feature class file named by the user and saved to the specified directory. Additionally, if selected, a hillshade layer is also saved in the same directory. Both files will be saved in the same coordinate system as the input DEM. The resultant spall layer contains a total of nine attribute fields. The first two "FID" and "Shape" are required fields for the creation of a shapefile, indicating an identification number and the type of feature. The next two fields, "Area_sqft" and "Area_cm^2" are the area of each spall in square feet and cm^2. The next two fields "CENTROIDX" and "CENTROIDY" are the longitude and latitude, respectively, in the coordinate system of the input DEM. The field "MxDepth_In" is the maximum depth, in inches, calculated from the DEM in relation to the mean of the non-spalled area immediately surrounding the spall. The remaining fields "Cubic_in" and "Cubic_ft" are the volumes, in inches and feet, respectively, of the spalls as calculated from a triangular irregular network of the DEM from within the spall extent.

9.3.4 Screenshot workflow

9.3.4.1 ArcDesktop Installation

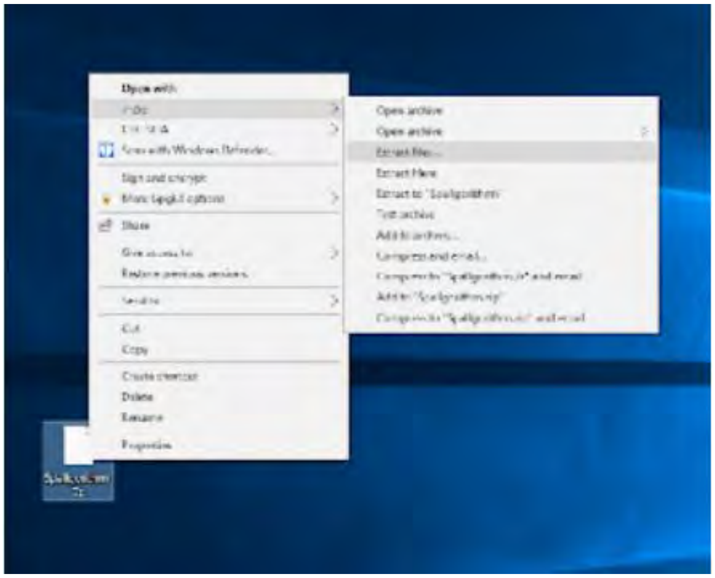


Figure A15: Extracting the files.

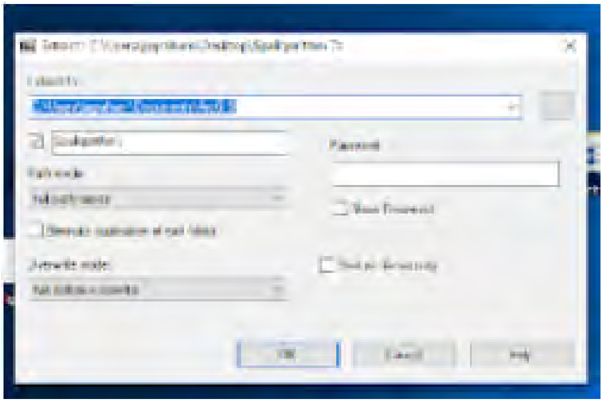
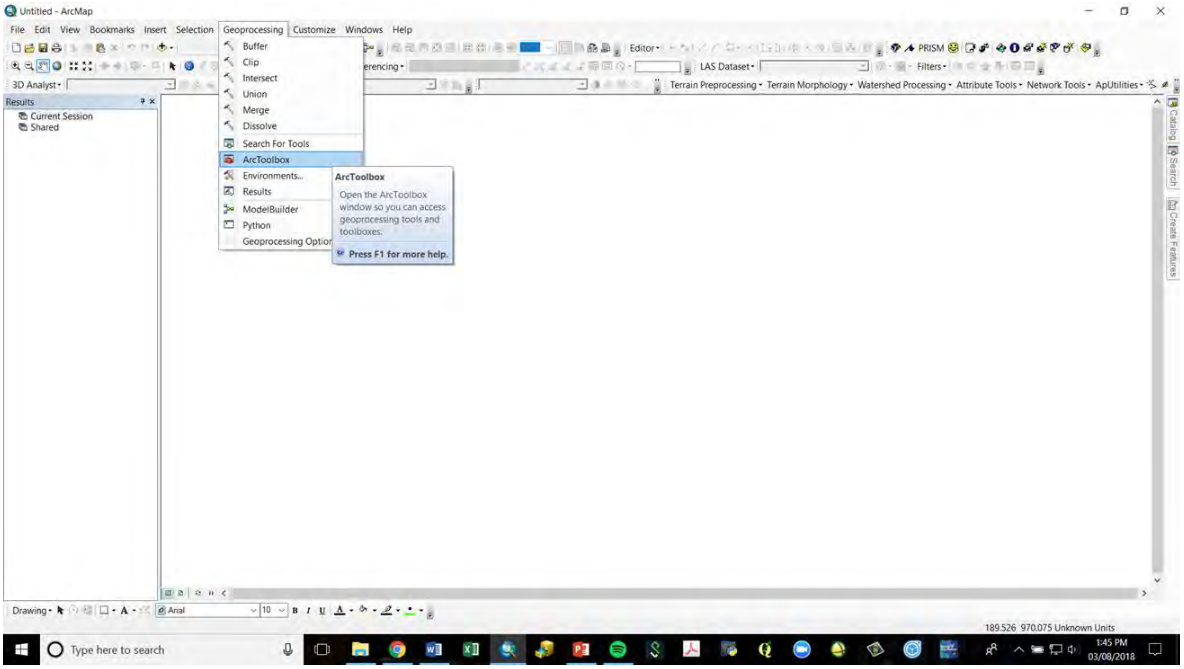


Figure A16: Choosing extraction location.

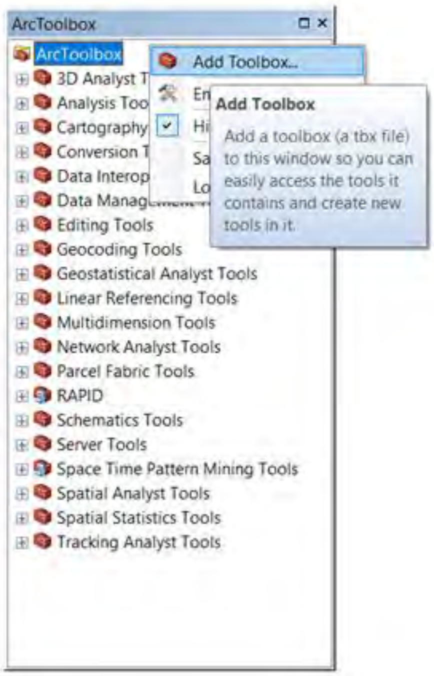


ArcToolbox

Open the ArcToolbox window so you can access geoprocessing tools and toolboxes.

Press F1 for more help.

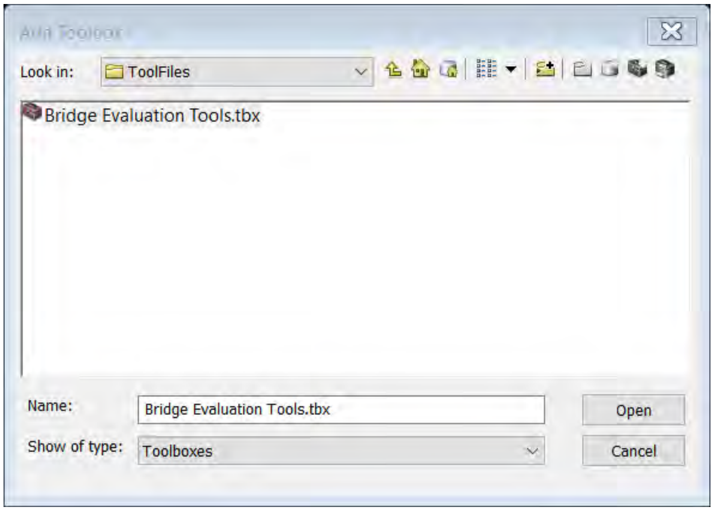
Figure A17: Opening ArcToolbox.



Add Toolbox

Add a toolbox (a tbx file) to this window so you can easily access the tools it contains and create new tools in it.

Figure A18: Adding a toolbox.

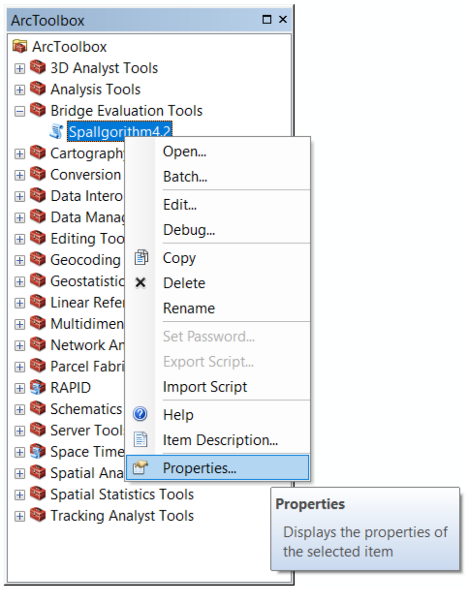


Look in: ToolFiles

Name: Bridge Evaluation Tools.tbx

Show of type: Toolboxes

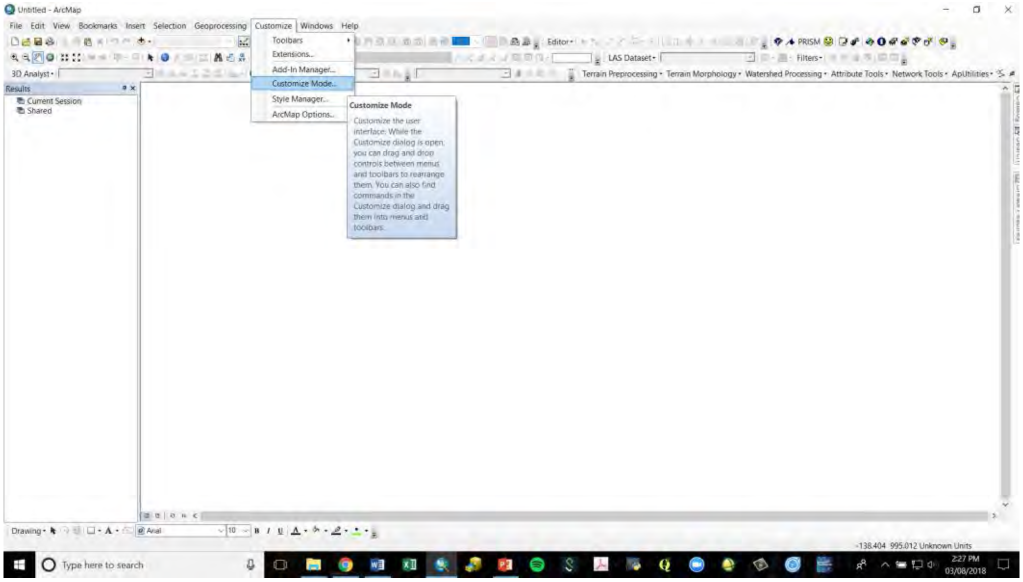
Figure A19: Finding the Bridge Evaluation Toolbox.



Properties

Displays the properties of the selected item

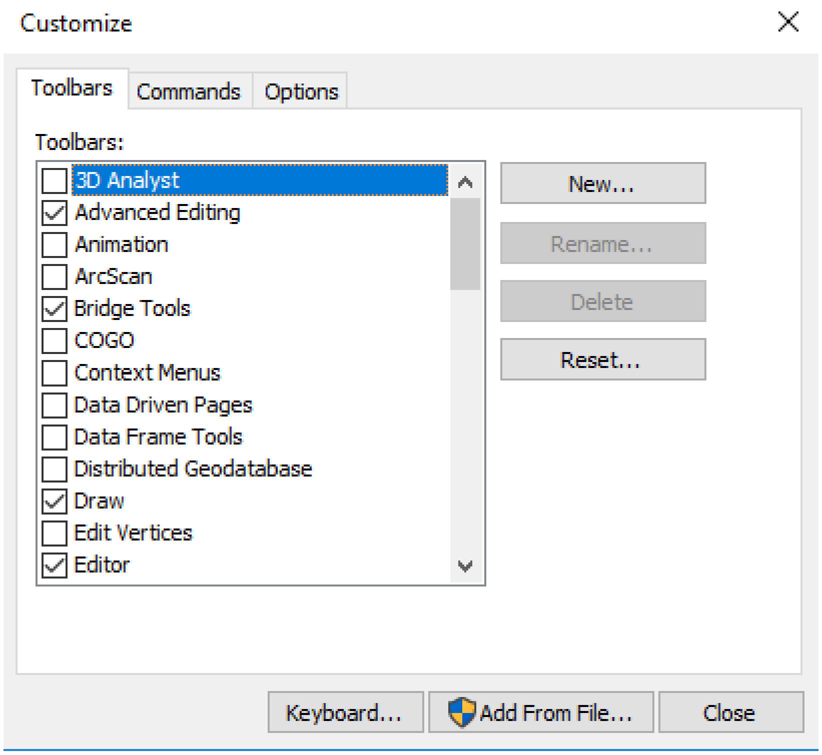
Figure A20: Updating the path to the script in the script tool properties.



Customize Mode

Customize the user interface. While the Customize dialog is open, you can drag and drop controls between menus and toolbars to rearrange them. You can also find commands in the Customize dialog and drag them into menus and toolbars.

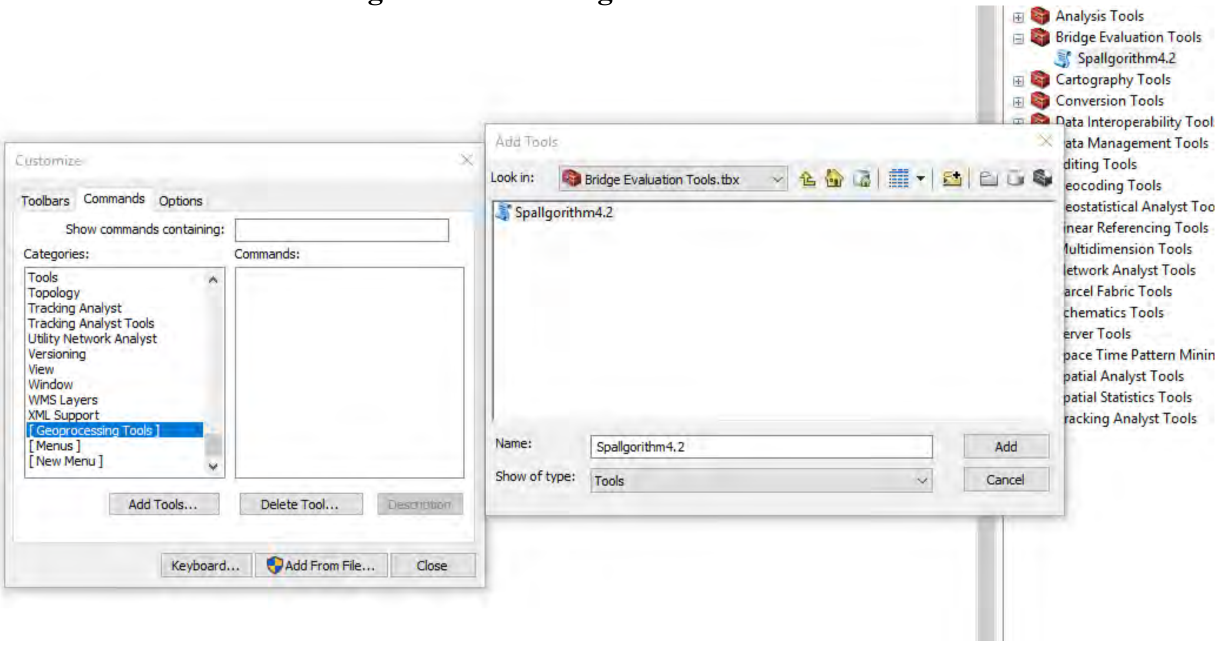
Figure A21: Opening customize mode.



Toolbars:

- □ 3D Analyst
- ☑ Advanced Editing
- □ Animation
- □ ArcScan
- ☑ Bridge Tools
- □ COGO
- □ Context Menus
- □ Data Driven Pages
- □ Data Frame Tools
- □ Distributed Geodatabase
- ☑ Draw
- □ Edit Vertices
- ☑ Editor

Figure A22: Adding the new toolbar.



Show commands containing:

Categories:

- Tools
- Topology
- Tracking Analyst
- Tracking Analyst Tools
- Utility Network Analyst
- Versioning
- View
- Window
- WMS Layers
- XML Support
- [Geoprocessing Tools]
- [Menus]
- [New Menu]

Commands:

Look in: Bridge Evaluation Tools.tbx

Spallgorithm4.2

Name: Spallgorithm4.2

Show of type: Tools

- Analysis Tools
- Bridge Evaluation Tools
  - Spallgorithm4.2
- Cartography Tools
- Conversion Tools
- Data Interoperability Tool
- ata Management Tools
- diting Tools
- eocoding Tools
- eostatistical Analyst Too
- inear Referencing Tools
- ultidimension Tools
- letwork Analyst Tools
- arcel Fabric Tools
- chematics Tools
- erver Tools
- pace Time Pattern Minin
- patial Analyst Tools
- patial Statistics Tools
- racking Analyst Tools

Figure A23: Adding the script tool to the available tools.

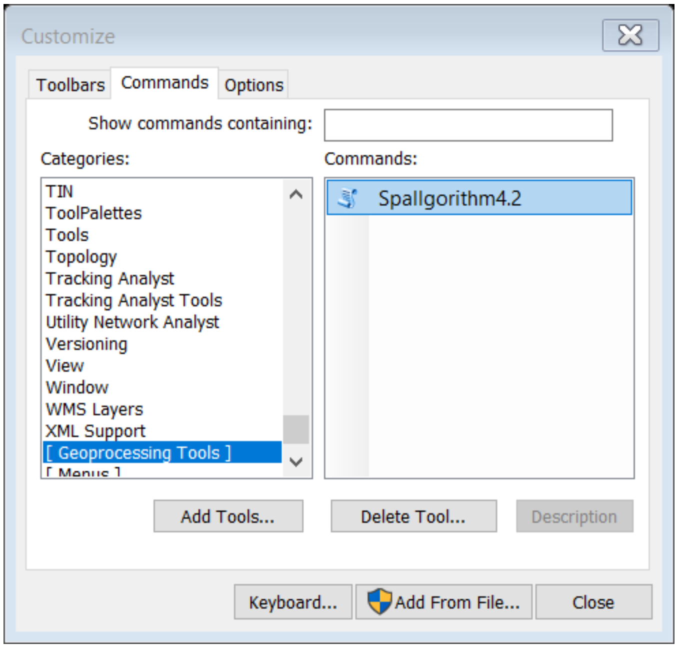


Figure A24: Adding the Spallgorithm script tool to the toolbar.

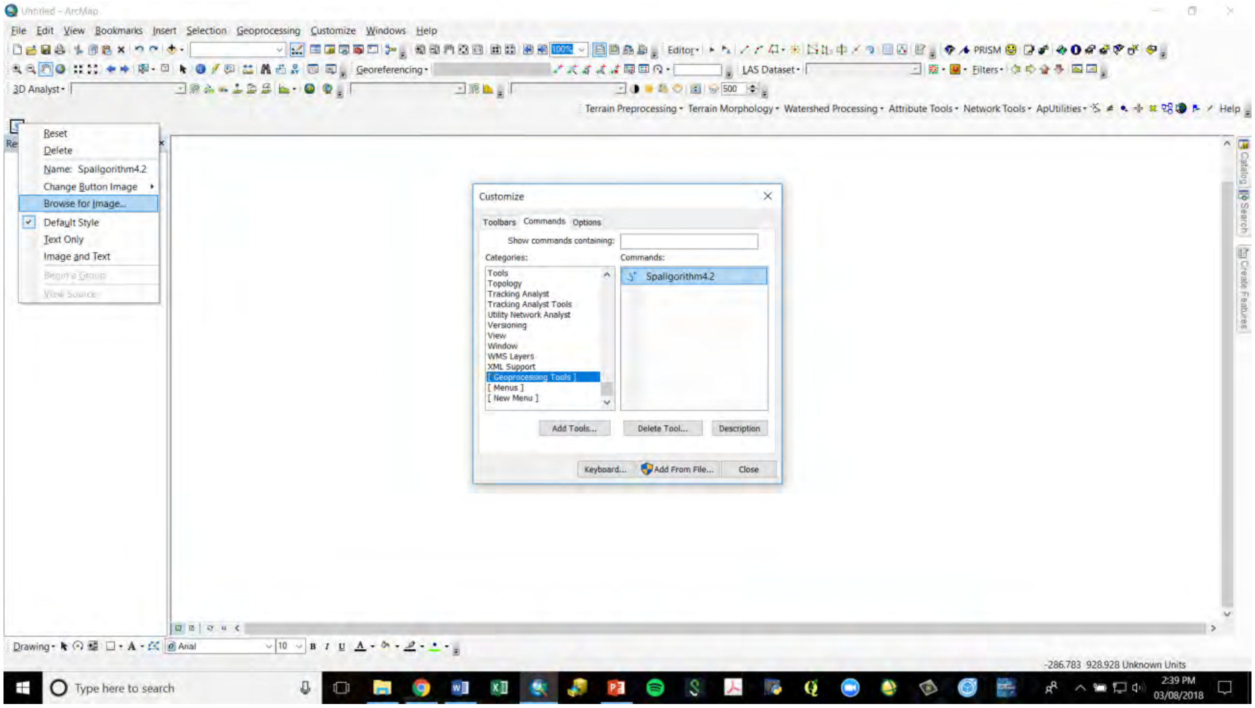
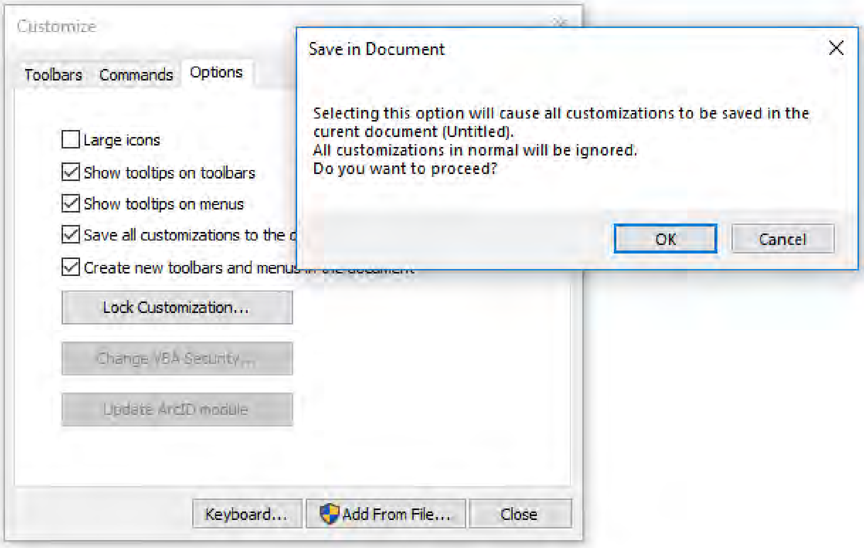


Figure A25: Adding the icon to the tool button in customize mode.



☐ Large icons

☑ Show tooltips on toolbars

☑ Show tooltips on menus

☑ Save all customizations to the d

☑ Create new toolbars and menus in the document

Save in Document

Selecting this option will cause all customizations to be saved in the curent document (Untitled).
All customizations in normal will be ignored.
Do you want to proceed?

Figure A26: Saving the customizations to the document

9.3.4.2 Creating an exclusion layer

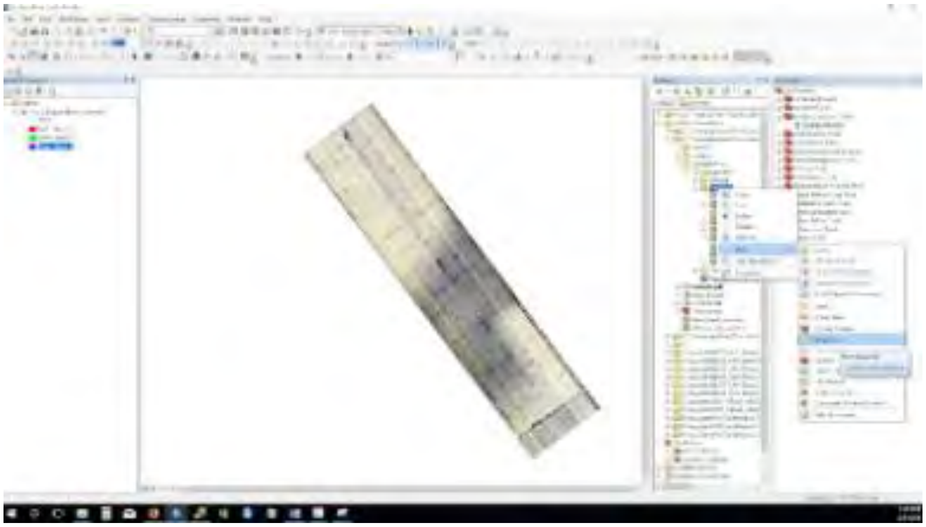
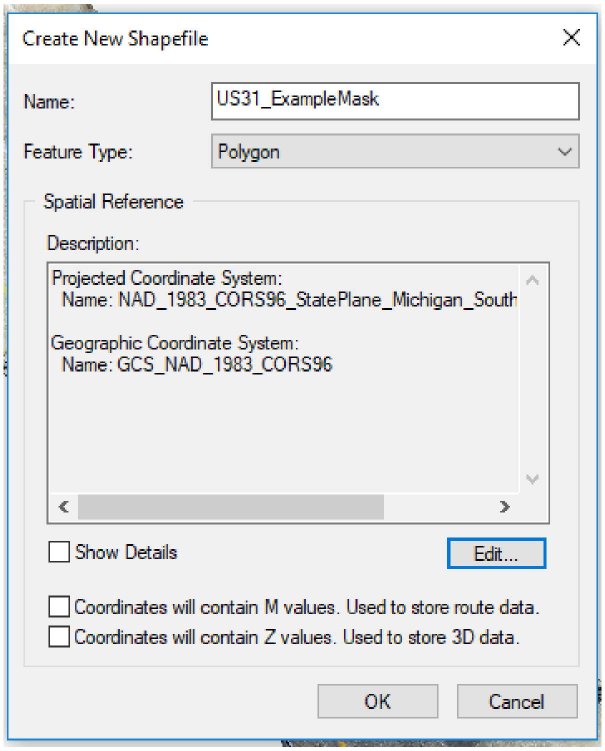


Figure A27: Creating a shapefiles for the exclusion layer.


Create New Shapefile

Name: US31_ExampleMask

Feature Type: Polygon

Spatial Reference

Description:

Projected Coordinate System:
 Name: NAD_1983_CORS96_StatePlane_Michigan_South

Geographic Coordinate System:
 Name: GCS_NAD_1983_CORS96

□ Show Details

□ Coordinates will contain M values. Used to store route data.

□ Coordinates will contain Z values. Used to store 3D data.

Figure A28: Assigning the exclusion layer a name and coordinate system.

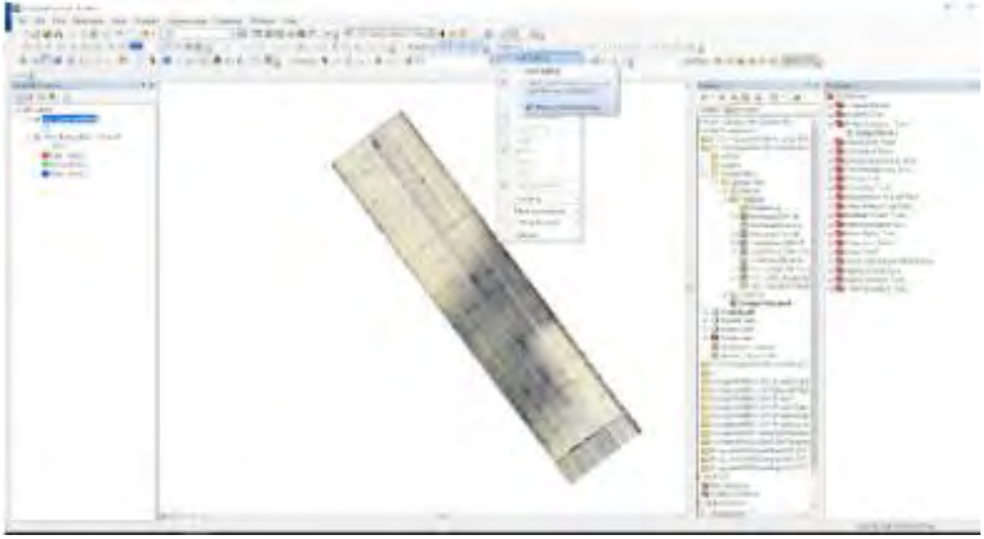


Figure A29: Opening the Editor Toolbar.

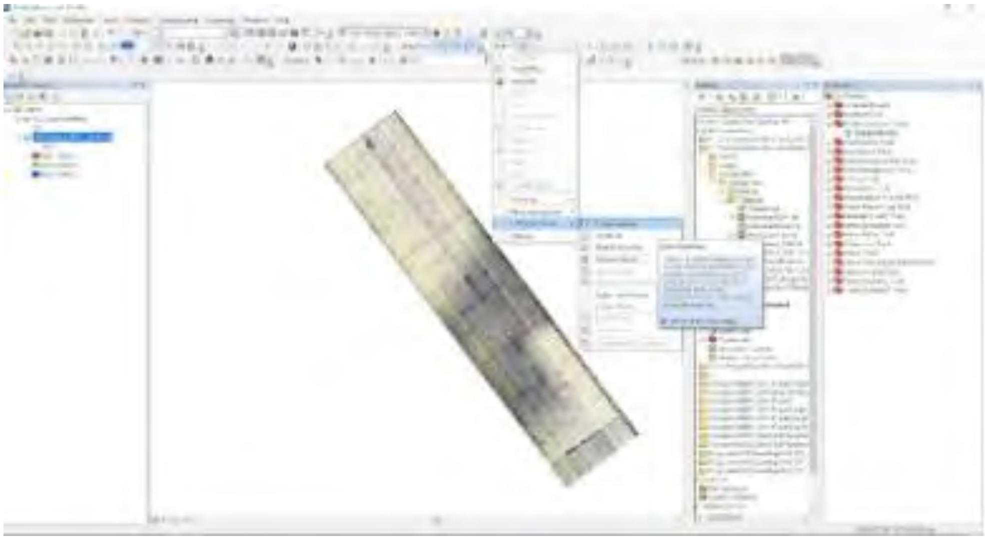


Figure A30: Opening the Create Features Window.

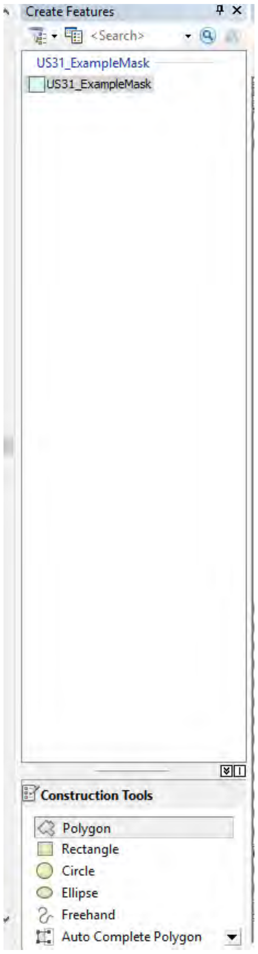


Figure A31: Selecting shapefile to edit.

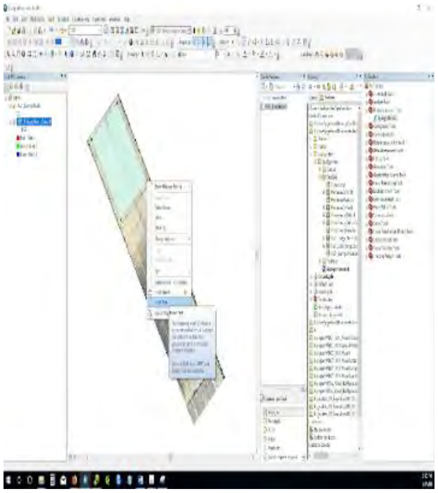


Figure A32: Creating a polygon part.

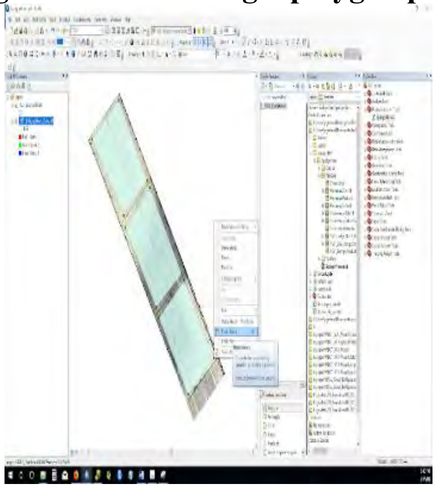


Figure A33: Continuing creating the polygon.

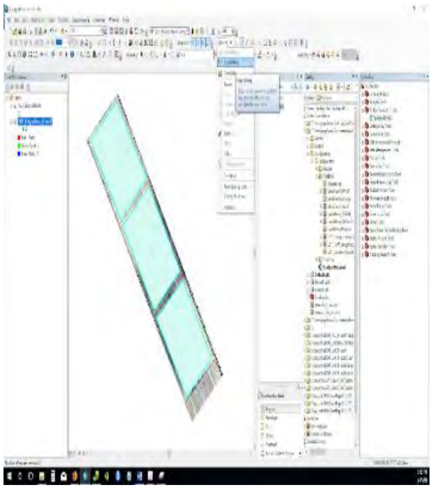


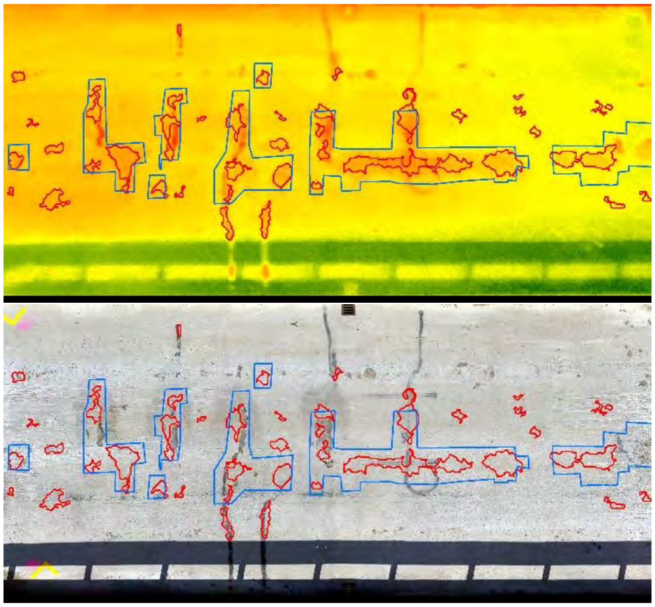
Figure A34: Saving the exclusion layer shapefile.

9.4. Thermal Delamination Detection Algorithm Manual

Thermal delamination detection algorithm 1.0

ArcGIS Desktop Tool

Operations Guide



Michigan Technological University

9.4.1 Overview

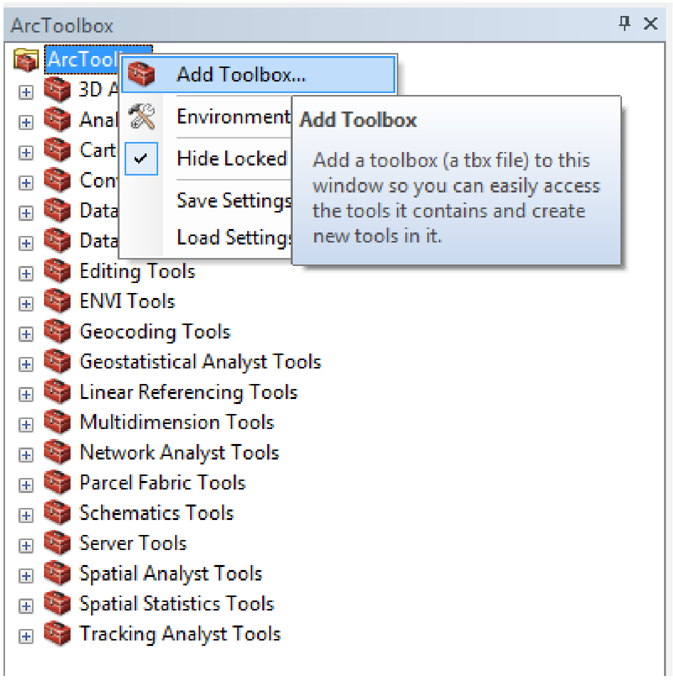
The Thermal Delamination Detection Algorithm (TDDA) is an ArcPy script tool designed to analyze thermal and visible imagery of road surfaces and delineate delaminations. The tool is designed to be integrated into the ArcGIS Desktop environment. The tool is delivered as a zipped folder and comes with example datasets. The Spallgorithm tool comes within a custom ArcToolbox where other similar tools may be added.

9.4.2 ArcGIS Desktop Installation

The first step is to unzip the folder to a directory where the script will run from. This can be a location on either a network or local drive. Right click on the zipped folder (“ThermalDelam.zip”) and use any decompression tool (e. g. 7-Zip) to decompress the files. It is recommended to use the same name for the folder in which you will decompress the .zip files (i.e. \ThermalDelam\).

9.4.2.1 Adding the toolbox

In ArcMap open the ArcToolbox by clicking on the icon or by clicking on Geoprocessing->ArcToolbox. When the tool box opens (Figure A35), right click on ArcToolbox at the top and select Add ToolBox.

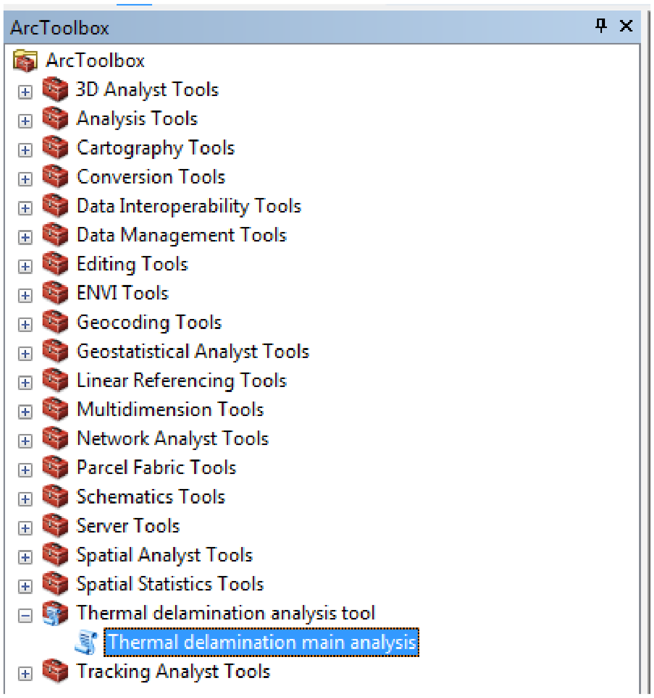


Add a toolbox (a tbx file) to this window so you can easily access the tools it contains and create new tools in it.

Figure A35: Opening the “add toolbox” dialog box to search for the tool.

In the Add Toolbox window navigate to the directory where the files were extracted. The toolbox is located in the folder \ThermalDelam\. Click on ThermalDelam.pyt and press open. You will see the toolbox added to the existing toolboxes as “Thermal Delamination Analysis

Tool" (Figure A36). Double click on the toolbox and you will see the "Thermal Delamination Main Analysis" tool.



ArcToolbox

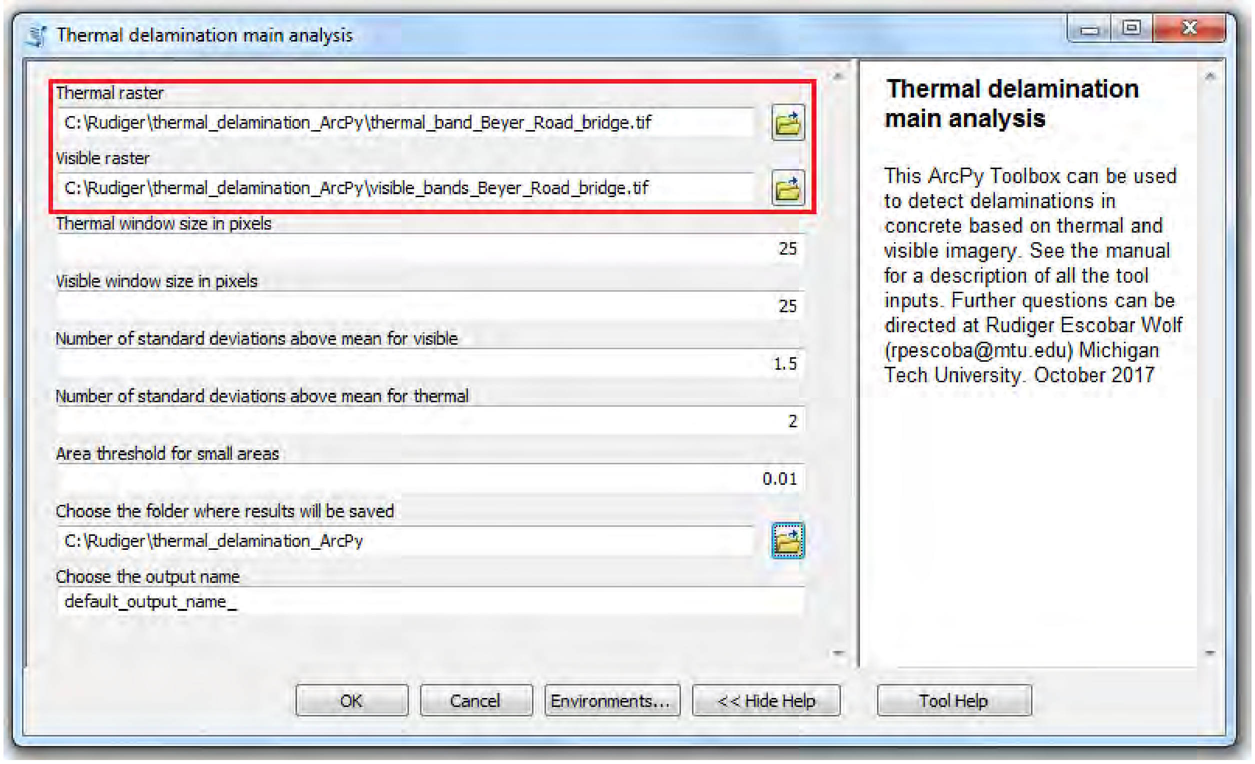
- ArcToolbox
  - 3D Analyst Tools
  - Analysis Tools
  - Cartography Tools
  - Conversion Tools
  - Data Interoperability Tools
  - Data Management Tools
  - Editing Tools
  - ENVI Tools
  - Geocoding Tools
  - Geostatistical Analyst Tools
  - Linear Referencing Tools
  - Multidimension Tools
  - Network Analyst Tools
  - Parcel Fabric Tools
  - Schematics Tools
  - Server Tools
  - Spatial Analyst Tools
  - Spatial Statistics Tools
  - Thermal delamination analysis tool
    - Thermal delamination main analysis
  - Tracking Analyst Tools

Figure A36: Loading and showing the "Thermal Delamination Analysis Tool" toolbox, and the "Thermal Delamination Main Analysis" tool.

9.4.3 Running the tool

9.4.3.1 Required Inputs

Two input datasets are required for the Thermal Delamination Analysis Tool: a thermal and a visible image of the road surface in raster format, preferably in geotif format. The user also has to choose the values for several parameters (Figure A37). Both thermal and visible image rasters have to be defined in the same plane projected coordinate system, e. g. State Plane coordinate system. Thermal images have to be single band rasters, while visible images have to be three band (common red-green-blue) rasters. Figure 3 shows the input dialog box for the tool, and the location of the thermal and visible raster file search and selection inputs (highlighted in the red box). Click on the folder icons to the right of the file path to navigate to the folder where you have the stored thermal and visible raster inputs, and once located, selected them as your respective inputs. You can also directly type, or copy-paste the raster file location and name on the corresponding input box.


Thermal delamination main analysis

Thermal raster
C:\Rudiger\thermal_delamination_ArcPy\thermal_band_Beyer_Road_bridge.tif

Visible raster
C:\Rudiger\thermal_delamination_ArcPy\visible_bands_Beyer_Road_bridge.tif

Thermal window size in pixels
25

Visible window size in pixels
25

Number of standard deviations above mean for visible
1.5

Number of standard deviations above mean for thermal
2

Area threshold for small areas
0.01

Choose the folder where results will be saved
C:\Rudiger\thermal_delamination_ArcPy

Choose the output name
default_output_name_

Thermal delamination main analysis

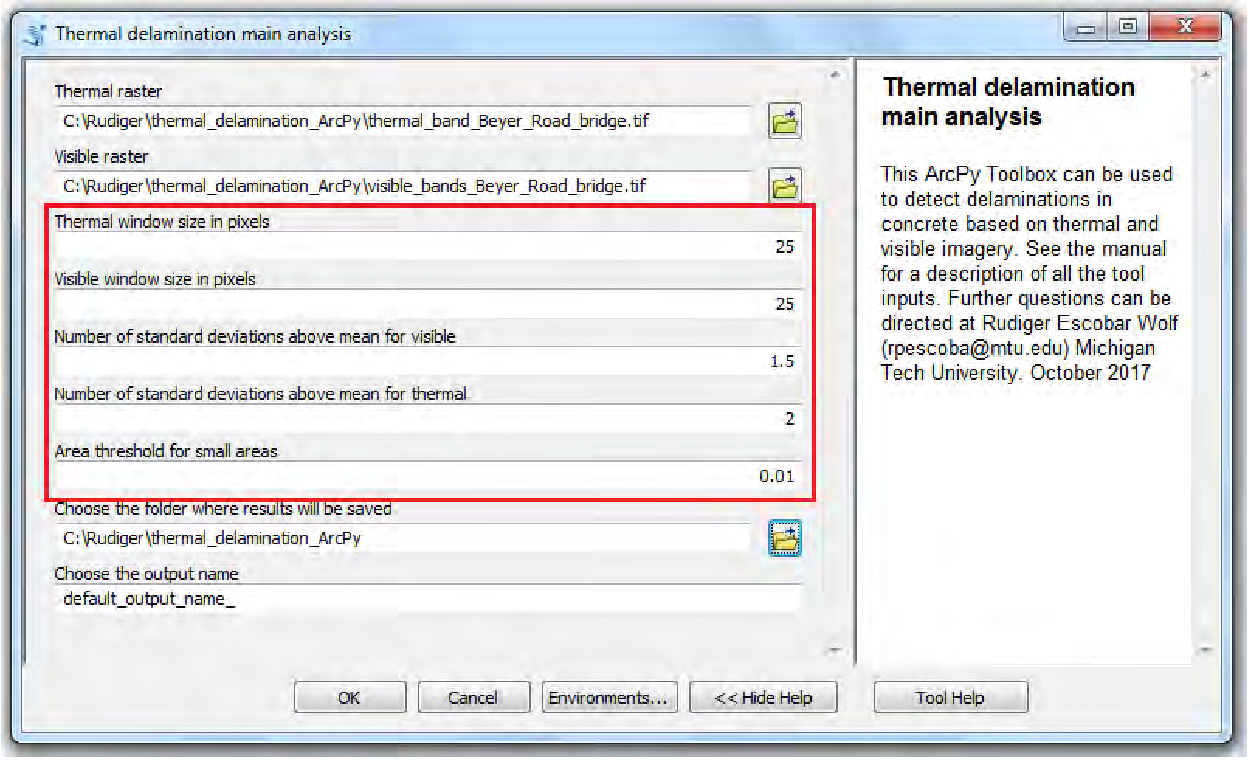
This ArcPy Toolbox can be used to detect delaminations in concrete based on thermal and visible imagery. See the manual for a description of all the tool inputs. Further questions can be directed at Rudiger Escobar Wolf (rpescoba@mtu.edu) Michigan Tech University. October 2017

Figure A37: Tool input dialog box, highlighting the input of thermal and visible raster files in the red box.

9.4.3.2 User defined variables

The tool has a total of five parameters to be specified. 1) Thermal window size in pixels, 2) Visible window size in pixels, 3) Number of standard deviations above mean for visible, 4) Number of standard deviations above mean for thermal, 5) Area threshold for small areas. Input values for all these parameters are required, but default values are already given when the tool is opened, and the user can choose to use those values or change them according to his or her needs. Figure A38 shows the input dialog box for the tool, and the location of the user defined variables (highlighted in the red box).

The size of the thermal and visible windows has to be chosen depending on the size of the expected target features (delaminations) that we want to identify and the dimensions of the road surface. For pixel ground resolutions on the order of a few cms, and two lane road widths, pixels windows on the order of 25 pixels worked well, and for these reason those are the default values chosen. But the user may have to test different options for these values and see what results in a better algorithm performance, depending on the case being analyzed.



Thermal raster
C:\Rudiger\thermal_delamination_ArcPy\thermal_band_Beyer_Road_bridge.tif

Visible raster
C:\Rudiger\thermal_delamination_ArcPy\visible_bands_Beyer_Road_bridge.tif

Thermal window size in pixels
25

Visible window size in pixels
25

Number of standard deviations above mean for visible
1.5

Number of standard deviations above mean for thermal
2

Area threshold for small areas
0.01

Choose the folder where results will be saved
C:\Rudiger\thermal_delamination_ArcPy

Choose the output name
default_output_name_

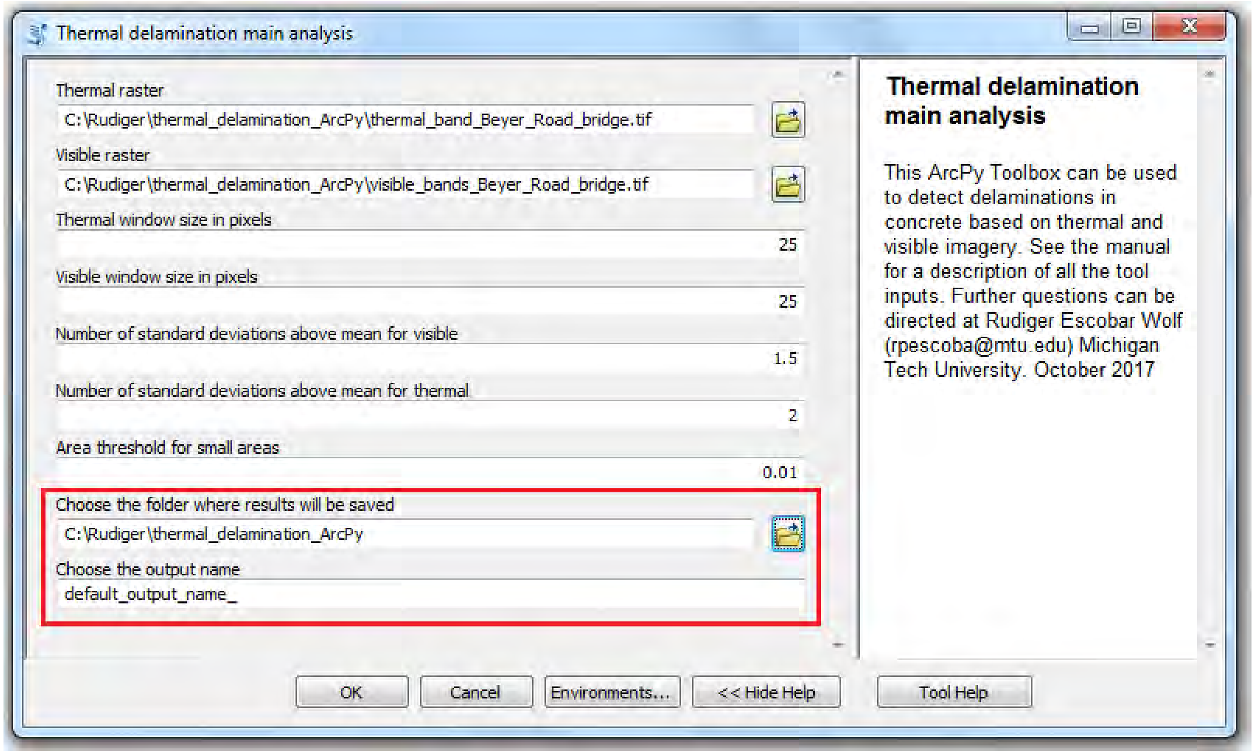
Thermal delamination main analysis

This ArcPy Toolbox can be used to detect delaminations in concrete based on thermal and visible imagery. See the manual for a description of all the tool inputs. Further questions can be directed at Rudiger Escobar Wolf (rpescoba@mtu.edu) Michigan Tech University. October 2017

Figure A38: Tool input dialog box, highlighting the user defined variable inputs in the red box.

The values for the standard deviations above the mean for the visible and thermal datasets can also vary, but values between 1 and 2.5 are usually effective at detecting and isolating potential delaminations. The default value for the visible case is 1.5 and the default for the thermal case is 2. Higher values correspond to a less sensitive setting and will generally result in less delamination areas, resulting in less false positives but also more false negatives. The user may have to go through a trial and error process to find the optimum combination of threshold values for each particular case.

Additionally, the user will have to specify the folder output location and name for the output files. The output will be an ESRI shapefile (.shp) containing polygons for all the mapped delamination areas, each of which will have an area calculated, in the mapping units of the original coordinate system in which the input layers are defined. Figure A39 shows the input dialog box for the tool, and the location of the user defined folder location and output file name (highlighted in the red box). Make sure to use a folder location that avoids having blank character or spaces in the folder names, as this may cause problems for ArcMap finding and reading the files. Also notice that it is not necessary to add the file extension (i. e. .shp) to the output name, and the tool does that automatically.



Thermal delamination main analysis

Thermal raster
C:\Rudiger\thermal_delamination_ArcPy\thermal_band_Beyer_Road_bridge.tif

Visible raster
C:\Rudiger\thermal_delamination_ArcPy\visible_bands_Beyer_Road_bridge.tif

Thermal window size in pixels
25

Visible window size in pixels
25

Number of standard deviations above mean for visible
1.5

Number of standard deviations above mean for thermal
2

Area threshold for small areas
0.01

Choose the folder where results will be saved
C:\Rudiger\thermal_delamination_ArcPy

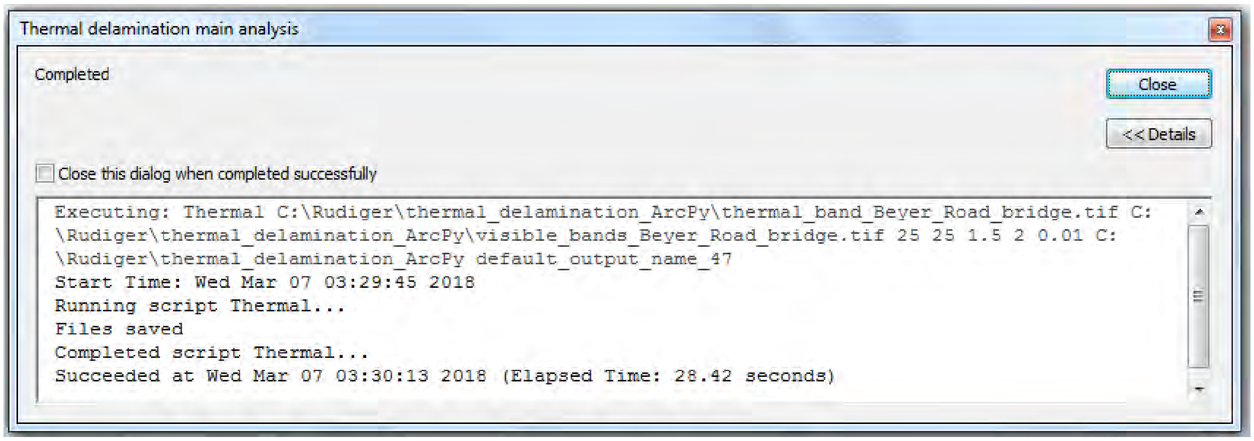
Choose the output name
default_output_name_

Thermal delamination main analysis

This ArcPy Toolbox can be used to detect delaminations in concrete based on thermal and visible imagery. See the manual for a description of all the tool inputs. Further questions can be directed at Rudiger Escobar Wolf (rpescoba@mtu.edu) Michigan Tech University. October 2017

Figure A39: Tool input dialog box, highlighting the user defined folder location and file name for the output file.

Running the tool may take a few minutes, depending on the size of the input raster datasets. Once the tool has run a result display box will be shown (Figure A40), indicating whether the tool has run without problems, or giving an error message. If the tool ran successfully, the result display box will display the name and location of the outputs, the time it took the tool to run, and other processing information.



Thermal delamination main analysis

Completed

□ Close this dialog when completed successfully

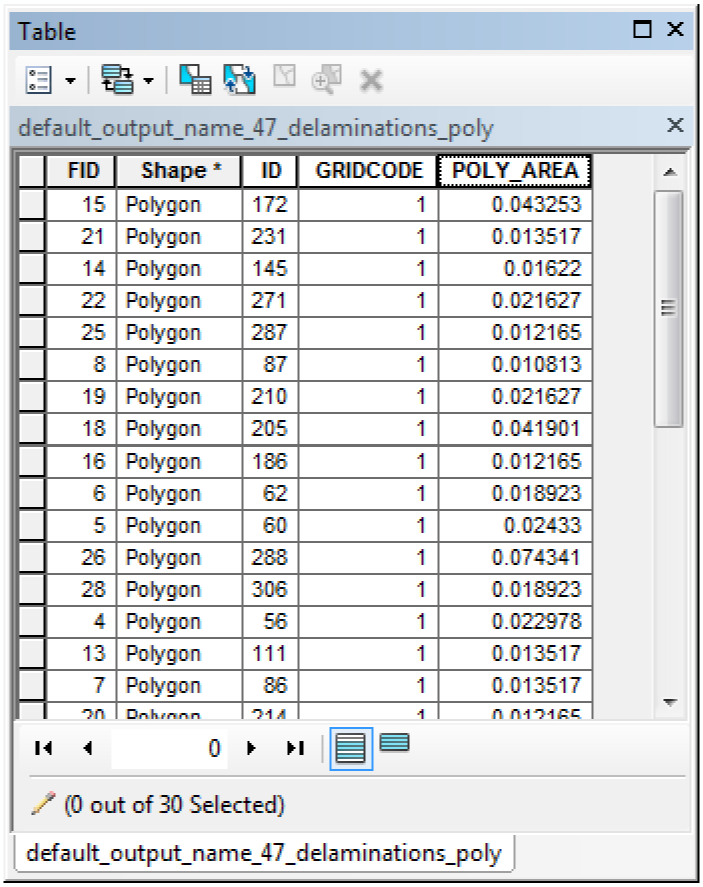
```
Executing: Thermal C:\Rudiger\thermal_delamination_ArcPy\thermal_band_Beyer_Road_bridge.tif C:
\Rudiger\thermal_delamination_ArcPy\visible_bands_Beyer_Road_bridge.tif 25 25 1.5 2 0.01 C:
\Rudiger\thermal_delamination_ArcPy default_output_name_47
Start Time: Wed Mar 07 03:29:45 2018
Running script Thermal...
Files saved
Completed script Thermal...
Succeeded at Wed Mar 07 03:30:13 2018 (Elapsed Time: 28.42 seconds)
```

Figure A40: Tool results display box.

The results of running the tool will be stored in the folder given as an input by the user, and can be loaded onto ArcMap after the tool finished running.

9.4.4 Understanding the output

The tool generates a shapefile named by the user and saved to the specified folder. The shapefile contains polygons with areas mapped as potential delaminations, with a total of five attribute fields (Figure A41). The first two "FID" and "Shape" are required fields for the creation of a shapefile, indicating an identification number and the type of feature. The next two fields, ID and GRIDCODE correspond the polygon individual identification numbers and the inherited values of the raster processing steps in the algorithm, and will usually not be of interest for the user. The fifth field, POLY_AREA, contains the areas for each of the polygons.



Table

default_output_name_47_delaminations_poly

FID	Shape *	ID	GRIDCODE	POLY_AREA
15	Polygon	172	1	0.043253
21	Polygon	231	1	0.013517
14	Polygon	145	1	0.01622
22	Polygon	271	1	0.021627
25	Polygon	287	1	0.012165
8	Polygon	87	1	0.010813
19	Polygon	210	1	0.021627
18	Polygon	205	1	0.041901
16	Polygon	186	1	0.012165
6	Polygon	62	1	0.018923
5	Polygon	60	1	0.02433
26	Polygon	288	1	0.074341
28	Polygon	306	1	0.018923
4	Polygon	56	1	0.022978
13	Polygon	111	1	0.013517
7	Polygon	86	1	0.013517
20	Polygon	214	1	0.012165

0

(0 out of 30 Selected)

default_output_name_47_delaminations_poly

Figure A41: Example of the attribute table for the shapefile resulting from running the tool. See description in the text.

Figure A42 shows the graphical display of the tool's output, compared with the raster inputs (thermal and visible). The upper panel shows a section of the visible imagery for the Beyer Road Bridge test site. The middle panel shows a thermal image for the same section of the Beyer Road Bridge, and the lower panel shows the resulting polygons (black outline) from the thermal delamination detection analysis.

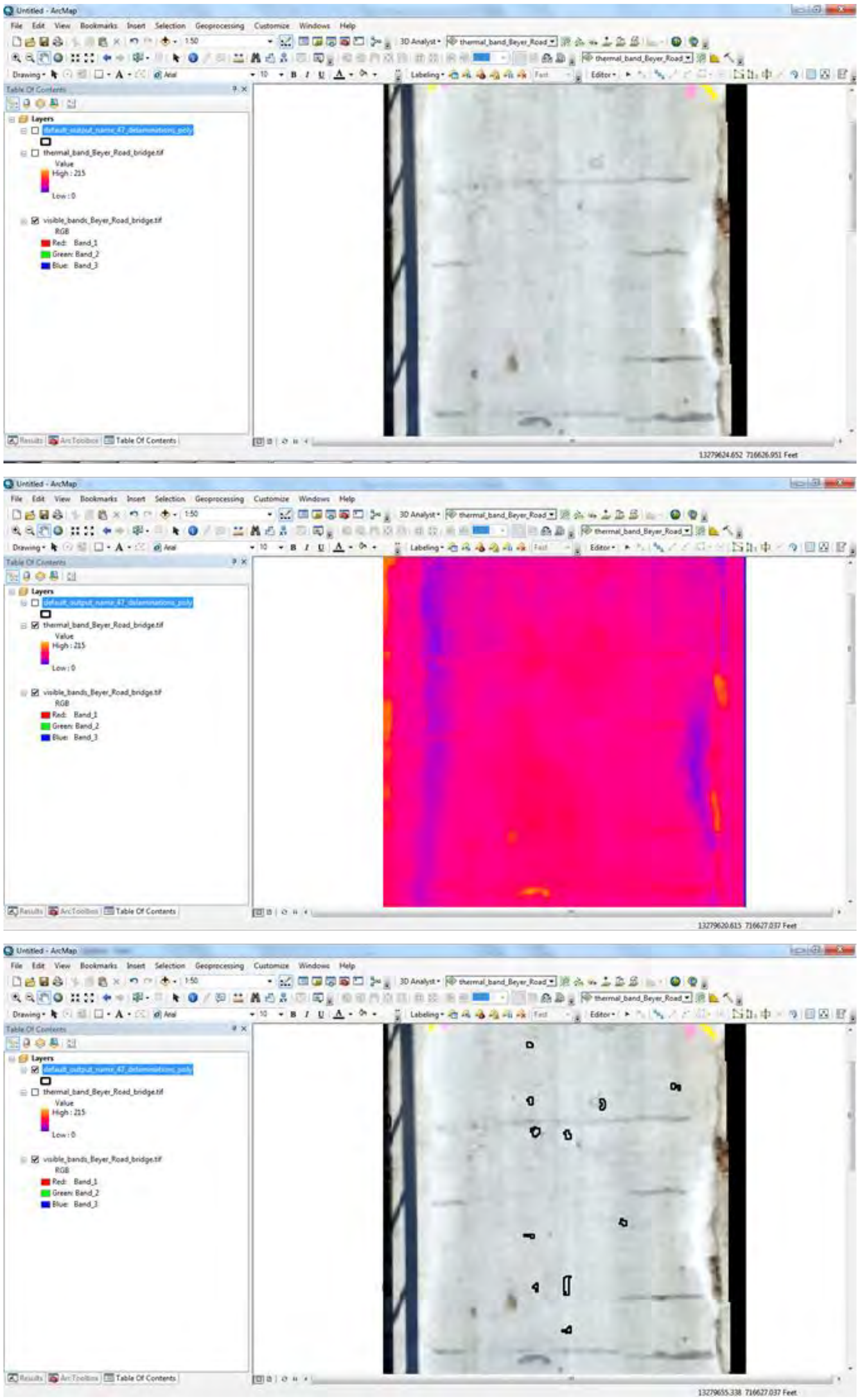

Untitled - ArcMap
File Edit View Bookmarks Insert Selection Geoprocessing Customize Windows Help
Table Of Contents
Layers
default_output_name_47_delaminations_poly
thermal_band_Beyer_Road_bridge.tif
Value
High : 215
Low : 0
visible_bands_Beyer_Road_bridge.tif
RGB
Red: Band_1
Green: Band_2
Blue: Band_3
Results
ArcToolbox
Table Of Contents

Figure A42: Example of the input datasets used in the thermal delamination detection tool, and the respective output results. See text for details.

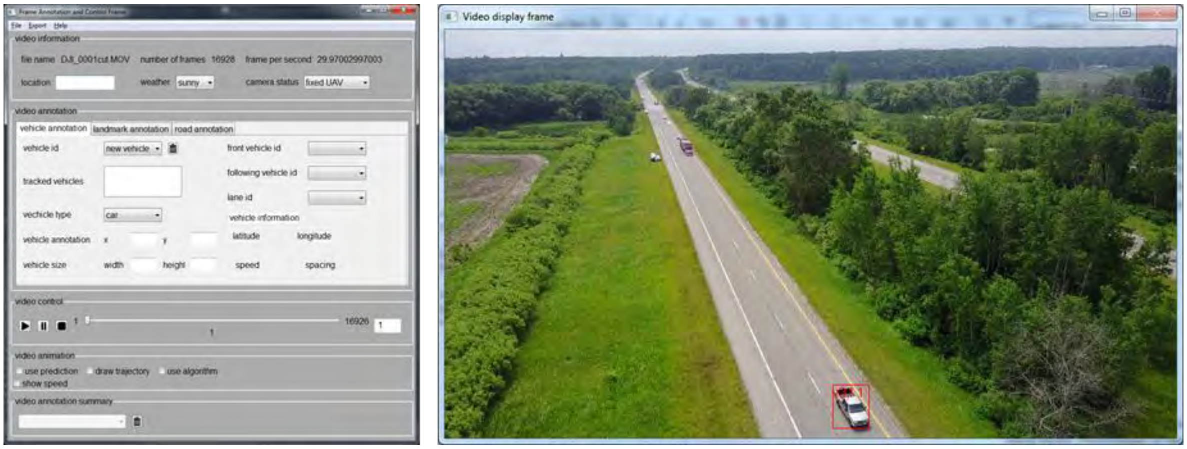
9.5 Traffic Monitoring

This section describes the semi-automated UAV traffic monitoring tool and the whole procedure to use the tool for traffic data processing and analysis. Section 9.5.1 is an overview. Section 9.5.2 introduces the steps to use the tool. Section 9.5.3 introduces methods and algorithms. Section 9.5.4 shows the results from video taken at US23 (Ann Arbor, MI) and US31 (Muskegon, MI). Section 9.5.5 gives a brief expectation of the future work.

9.5.1 Overview of the semi-automated UAV traffic monitoring tool

The semi-automated UAV traffic monitoring tool is developed using C/C++ and wxWidget libraries for a Graphical User Interface (GUI), and Open Computer Vision (OpenCV) libraries for non-parametric algorithms to detect vehicles and roads in a UAV video with minimal manual annotations. By using this tool, users can add landmark information (for geo-registration due to remote sensing), label an object (e.g. individual vehicle driving states and road characteristics) of interest, and conduct traffic data analysis (e.g. traffic studies such as volume and speed studies etc.) semi-automatically. This tool also outputs frame-by-frame estimated annotations of individual vehicle driving states (e.g. location and speed) using a homogeneous perspective projection algorithm to overcome the UAV vibration issue with very few manual annotations (e.g. five to ten annotations per vehicle). The automatic frame-by-frame estimated annotations can provide a labeled dataset for training advanced machine learning algorithms to automatic vehicle detection and tracking, which is more efficient than pure manual annotations.

The GUI includes two parts, one is an Annotation and Control Panel for presenting the annotation information and controlling the video; the other one is a Video Display Window for annotation operations in a video (e.g. selecting the tracked object, adding projection points, labeling the objects of interest) and animation traffic flow dynamics (e.g. volume, speed, and individual vehicle trajectory). Figure A43 shows the GUI of the semi-automated traffic monitoring tool.


Frame Annotation and Control Frame

video information

file name DJI_0001cut.MOV number of frames 16928 frame per second 29.97002997003

location

weather sunny

camera status fixed UAV

video annotation

vehicle id new vehicle

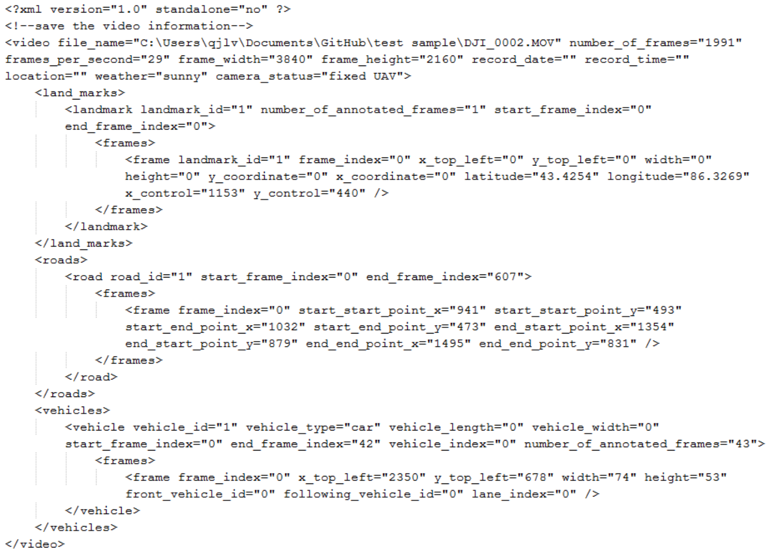
vechicle type car

video control

1 16926 1

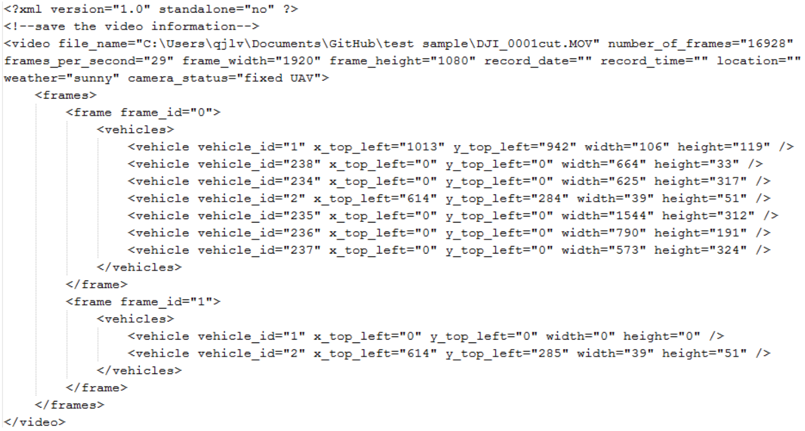
Figure A43: GUI of the semi-automated UAV traffic monitoring tool.

The semi-automated UAV traffic monitoring tool will output four files (Figure A44): (1) (video_name)_manual_annotation.xml for recording manual annotations using the GUI, (2) (video_name)_autmatic_annotation.xml for storing estimated frame-by-frame annotations, (3) (video_name)_vehicle_trajectory.csv for visualizing the trajectory of all vehicles in the video using vehicle trajectory analysis and visualization tool - Mobile Sensor Data Analyzer, and (4) (video_name)_traffic_variable.csv for analyzing the traffic flow dynamics. Extensible Markup Language (XML) and Comma-Separated Values (CSV) formats are chosen to define these files. The content of XML is meaningful and easy to understand for both computer and human and CSV file is suitable for data analysis. We define different formats for different files.



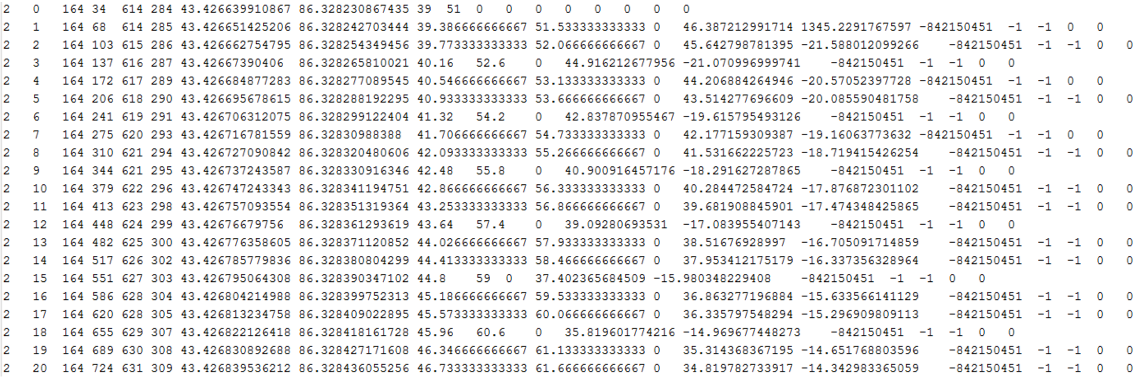
```
<?xml version="1.0" standalone="no" ?>
<!--save the video information-->
<video file_name="C:\Users\qjlv\Documents\GitHub\test sample\DJI_0002.MOV" number_of_frames="1991"
frames_per_second="29" frame_width="3840" frame_height="2160" record_date="" record_time=""
location="" weather="sunny" camera_status="fixed UAV">
    <land_marks>
        <landmark landmark_id="1" number_of_annotated_frames="1" start_frame_index="0"
        end_frame_index="0">
            <frames>
                <frame landmark_id="1" frame_index="0" x_top_left="0" y_top_left="0" width="0"
                height="0" y_coordinate="0" x_coordinate="0" latitude="43.4254" longitude="86.3269"
                x_control="1153" y_control="440" />
            </frames>
        </landmark>
    </land_marks>
    <roads>
        <road road_id="1" start_frame_index="0" end_frame_index="607">
            <frames>
                <frame frame_index="0" start_start_point_x="941" start_start_point_y="493"
                start_end_point_x="1032" start_end_point_y="473" end_start_point_x="1354"
                end_start_point_y="879" end_end_point_x="1495" end_end_point_y="831" />
            </frames>
        </road>
    </roads>
    <vehicles>
        <vehicle vehicle_id="1" vehicle_type="car" vehicle_length="0" vehicle_width="0"
        start_frame_index="0" end_frame_index="42" vehicle_index="0" number_of_annotated_frames="43">
            <frames>
                <frame frame_index="0" x_top_left="2350" y_top_left="678" width="74" height="53"
                front_vehicle_id="0" following_vehicle_id="0" lane_index="0" />
        </vehicle>
    </vehicles>
</video>
```

(a) Format (video_name)_manual_annotation.xml – manual annotations of road, landmark, and vehicle characteristics.



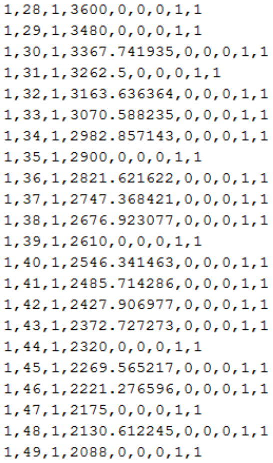
```
<?xml version="1.0" standalone="no" ?>
<!--save the video information-->
<video file_name="C:\Users\qjlv\Documents\GitHub\test sample\DJI_0001cut.MOV" number_of_frames="16928"
frames_per_second="29" frame_width="1920" frame_height="1080" record_date="" record_time="" location=""
weather="sunny" camera_status="fixed UAV">
    <frames>
        <frame frame_id="0">
            <vehicles>
                <vehicle vehicle_id="1" x_top_left="1013" y_top_left="942" width="106" height="119" />
                <vehicle vehicle_id="238" x_top_left="0" y_top_left="0" width="664" height="33" />
                <vehicle vehicle_id="234" x_top_left="0" y_top_left="0" width="625" height="317" />
                <vehicle vehicle_id="2" x_top_left="614" y_top_left="284" width="39" height="51" />
                <vehicle vehicle_id="235" x_top_left="0" y_top_left="0" width="1544" height="312" />
                <vehicle vehicle_id="236" x_top_left="0" y_top_left="0" width="790" height="191" />
                <vehicle vehicle_id="237" x_top_left="0" y_top_left="0" width="573" height="324" />
            </vehicles>
        </frame>
        <frame frame_id="1">
            <vehicles>
                <vehicle vehicle_id="1" x_top_left="0" y_top_left="0" width="0" height="0" />
                <vehicle vehicle_id="2" x_top_left="614" y_top_left="285" width="39" height="51" />
            </vehicles>
        </frame>
    </frames>
</video>
```

(b) Format of (video_name)_automatic_annotation.xml – estimated annotations, vehicle trajectory and traffic flow data.



```
2   0   164 34  614 284 43.426639910867 86.328230867435 39  51  0   0   0   0   0   0   0   0
2   1   164 68  614 285 43.426651425206 86.328242703444 39.386666666667 51.533333333333 0   46.387212991714 1345.2291767597 -842150451  -1  -1  0   0
2   2   164 103 615 286 43.426662754795 86.328254349456 39.773333333333 52.066666666667 0   45.642798781395 -21.588012099266    -842150451  -1  -1  0   0
2   3   164 137 616 287 43.42667390406  86.328265810021 40.16   52.6    0   44.916212677956 -21.070996999741    -842150451  -1  -1  0   0
2   4   164 172 617 289 43.426684877283 86.328277089545 40.546666666667 53.133333333333 0   44.206884264946 -20.57052397728 -842150451  -1  -1  0   0
2   5   164 206 618 290 43.426695678615 86.328288192295 40.933333333333 53.666666666667 0   43.514277696609 -20.085590481758    -842150451  -1  -1  0   0
2   6   164 241 619 291 43.426706312075 86.328299122404 41.32   54.2    0   42.837870955467 -19.615795493126    -842150451  -1  -1  0   0
2   7   164 275 620 293 43.426716781559 86.32830988388  41.706666666667 54.733333333333 0   42.177159309387 -19.16063773632 -842150451  -1  -1  0   0
2   8   164 310 621 294 43.426727090842 86.328320480606 42.093333333333 55.266666666667 0   41.531662225723 -18.719415426254    -842150451  -1  -1  0   0
2   9   164 344 621 295 43.426737243587 86.328330916346 42.48   55.8    0   40.900916457176 -18.291627287865    -842150451  -1  -1  0   0
2   10  164 379 622 296 43.426747243343 86.328341194751 42.866666666667 56.333333333333 0   40.284472584724 -17.876872301102    -842150451  -1  -1  0   0
2   11  164 413 623 298 43.426757093554 86.328351319364 43.253333333333 56.866666666667 0   39.681908845901 -17.474348425865    -842150451  -1  -1  0   0
2   12  164 448 624 299 43.42676679756  86.328361293619 43.64   57.4    0   39.09280693531  -17.083955407143    -842150451  -1  -1  0   0
2   13  164 482 625 300 43.426776358605 86.328371120852 44.026666666667 57.933333333333 0   38.51676928997  -16.705091714859    -842150451  -1  -1  0   0
2   14  164 517 626 302 43.426785779836 86.328380804299 44.413333333333 58.466666666667 0   37.953412175179 -16.337356328964    -842150451  -1  -1  0   0
2   15  164 551 627 303 43.426795064308 86.328390347102 44.8    59  0   37.402365684509 -15.980348229408    -842150451  -1  -1  0   0
2   16  164 586 628 304 43.426804214988 86.328399752313 45.186666666667 59.533333333333 0   36.863277196884 -15.633566141129    -842150451  -1  -1  0   0
2   17  164 620 628 305 43.426813234758 86.328409022895 45.573333333333 60.066666666667 0   36.335797548294 -15.296909809113    -842150451  -1  -1  0   0
2   18  164 655 629 307 43.426822126418 86.328418161728 45.96   60.6    0   35.819601774216 -14.969677448273    -842150451  -1  -1  0   0
2   19  164 689 630 308 43.426830892688 86.328427171608 46.346666666667 61.133333333333 0   35.314368367195 -14.651768803596    -842150451  -1  -1  0   0
2   20  164 724 631 309 43.426839536212 86.328436055256 46.733333333333 61.666666666667 0   34.819782733917 -14.342983365059    -842150451  -1  -1  0   0
```

(c) Format of (video_name)_vehicle_trajectory.csv – file for visualizing the time-space trajectory.



```
1,28,1,3600,0,0,0,1,1
1,29,1,3480,0,0,0,1,1
1,30,1,3367.741935,0,0,0,1,1
1,31,1,3262.5,0,0,0,1,1
1,32,1,3163.636364,0,0,0,1,1
1,33,1,3070.588235,0,0,0,1,1
1,34,1,2982.857143,0,0,0,1,1
1,35,1,2900,0,0,0,1,1
1,36,1,2821.621622,0,0,0,1,1
1,37,1,2747.368421,0,0,0,1,1
1,38,1,2676.923077,0,0,0,1,1
1,39,1,2610,0,0,0,1,1
1,40,1,2546.341463,0,0,0,1,1
1,41,1,2485.714286,0,0,0,1,1
1,42,1,2427.906977,0,0,0,1,1
1,43,1,2372.727273,0,0,0,1,1
1,44,1,2320,0,0,0,1,1
1,45,1,2269.565217,0,0,0,1,1
1,46,1,2221.276596,0,0,0,1,1
1,47,1,2175,0,0,0,1,1
1,48,1,2130.612245,0,0,0,1,1
1,49,1,2088,0,0,0,1,1
```

(d) Format of (video_name)_traffic_variable.csv – for traffic flow analysis.

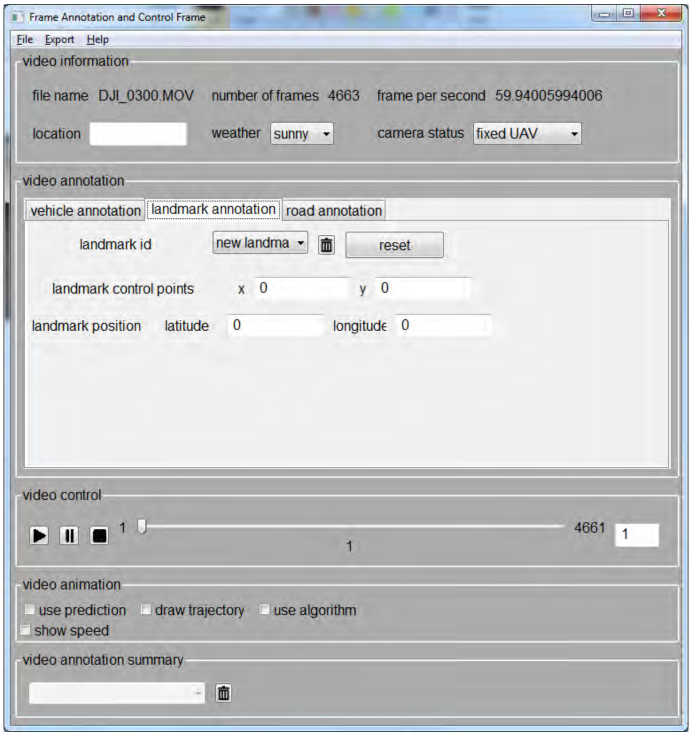
Figure A44. Input and Output file format.

9.5.2 Procedure of using the semi-automated UAV traffic monitoring tool

There are six steps to use the semi-automated UAV traffic monitoring tool for traffic data analysis: (1) landmark annotation, (2) road annotation, (3) vehicle annotation, (4) frame-by-frame annotation estimation, (5) estimated annotations output, and (6) traffic analysis and animation.

9.5.2.1 Landmark Annotation

Landmark annotation aims to get the world coordinate of a point on a frame, which is used to calculate actual speed and distance in the real world from the remote sensing data - UAV video frames. In order to build the relationship between points (defined as control points) on each frame, the tool uses two types of geo-registered images as reference images: (1) orthomosaic photos, and (2) Google satellite images. The landmark annotation is to select four corresponding points on both reference image and a video frame (Figure A45).



file name DJI_0300.MOV number of frames 4663 frame per second 59.94005994006

location weather sunny camera status fixed UAV

landmark id new landma

landmark control points x 0 y 0

landmark position latitude 0 longitude 0

1 4661 1

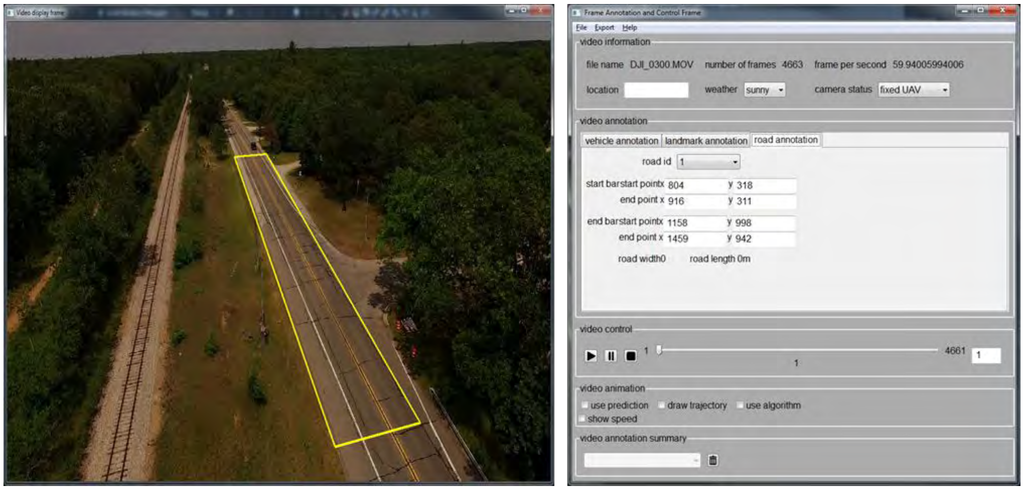
1

□ use prediction □ draw trajectory □ use algorithm
□ show speed

Figure A45: Landmark Annotation.

9.5.2.2 Road Annotation

Road annotation is the second procedure for collecting traffic data. Users only need to annotate the road on the first frame, the software will generate the road annotations on the following frames automatically. The tool also provides a manual way to annotate a road or revise a road annotation. Figure A46 shows the GUI for road annotation.



file name DJI_0300.MOV number of frames 4663 frame per second 59.94005994006

location weather sunny camera status fixed UAV

road id 1

start barstart pointx 804 y 318
end point x 916 y 311

end barstart pointx 1158 y 998
end point x 1459 y 942

road width0 road length 0m

1 4661 1

1

□ use prediction □ draw trajectory □ use algorithm
□ show speed

Figure A46: Road Annotation.

9.5.2.3 Vehicle Annotation

Vehicle annotation aims to provide a semi-automatic way for users to annotate vehicles. To train machine learning algorithms to detect and track individual vehicle driving dynamics, very large number of frame-by-frame ground truth data points are required for constructing a training and validation dataset. It is time intensive work to annotate all the vehicles frame-by-frame manually. The key idea of this tool is to use a very few sample annotations of a vehicle object to estimate frame-by-frame annotations of the vehicle object in a video. Similar to landmark and road annotation, users only need to annotate a vehicle on several key frames (e.g. when a vehicle appears or disappears in the video and when the appearance or position changes suddenly). An automatic annotation algorithm is used to generate annotations between manually annotated frames.

9.5.2.4 Frame-by-Frame Annotation Estimation

Frame-by-frame annotation estimation (i.e. landmark, road, and vehicle) uses manually annotations of landmark, road, and vehicles from a sample of frames in a video to automatically estimate all the annotations of landmark, road, and vehicles frame-by-frame in the video. This automatic frame-by-frame annotation estimation can deal with UAV video vibration issues.

9.5.2.5. Estimated Annotation Outputs

Estimated annotation outputs include three files: (1) frame-by-frame estimated annotations of landmark, road, and vehicles, (2) vehicle trajectories, and (3) traffic variables. Frame-by-frame annotation estimation file records all generated vehicle annotations in the video. The information in this file will be used in Section 9.5.2.6 to show the annotation on the screen. Vehicle trajectory file can be visualized as time-space diagram in Mobile Sensor Data Analyzer. Time-space diagram is a way to analyze and show traffic condition. The traffic variable file is used to analyze the traffic flow statistics on roads in a video. The data can be analyzed using other software (e.g. Excel).

9.5.2.6. Traffic Analysis and Animation

In this step, the data we collect in the previous steps will be presented on the screen directly. Users can use the animation to watch the results such as vehicle driving trajectory and speed, traffic volumes, number of vehicles on the road. Figure A47 shows the trajectory and speed of each vehicle, and traffic volumes in the video.

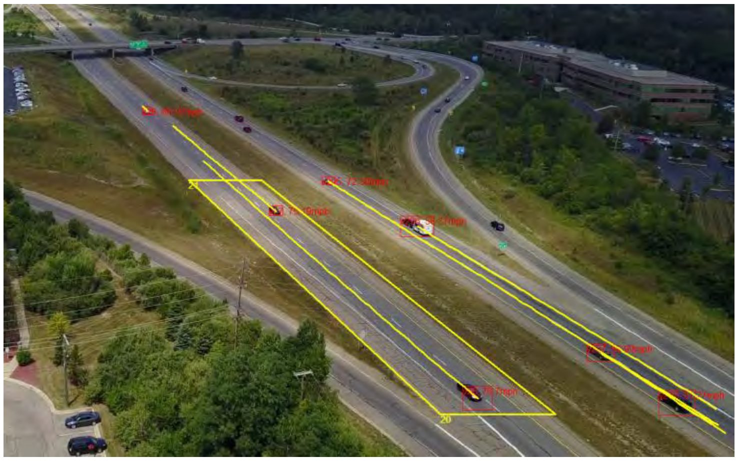


Figure A47: Vehicle Trajectory and Speeds and Traffic Volumes.

9.5.3 Methods and Algorithms of the Automatic Frame-By-Frame Annotation Estimation and Traffic Data Analysis

This section presents methods and algorithms used in the automatic frame-by-frame annotation estimation step in Section 9.5.2.4 and traffic data analysis in Section 9.5.2.6. The landmark, road and vehicle manual annotations are used for landmark annotation estimation frame-by-frame annotation estimation, a homogeneous perspective projection method is used in Section 9.5.3.1. For road annotation estimation, the k-means algorithm (Hartigan and Wong, 1979) and Hough transform algorithm (Duda and Hart, 1972) are combined to find a road segment is Section 9.5.3.2. For vehicle annotation estimation, both linear interpolation and a mean shift algorithm (Comaniciu and Meer, 2002) based on frame-by-frame difference image are implemented in Section 9.5.3.3. Section 9.5.3.4 presents the methods to analyze traffic dynamics using automatic estimated vehicle trajectories.

9.5.3.1 Homogeneous perspective projection method using landmark annotation

In our software, the homogeneous perspective projection method is used to build the relationship between the coordinates of control points on a frame in the video and the real world coordinates (e.g. latitude and longitude). The input of homogeneous perspective projection method is the pairs of control points and its corresponding real world coordinates. The algorithm will output a transformation matrix indicating the relationship between coordinates on image and real world coordinates.

The coordinates on frames can only provide the relative speed and location information of vehicles. However, such information may be not reliable because of UAV vibrations. In order to derive the relationship between video coordinates and world coordinates, four control points are needed. We use a homogeneous perspective projection method to build the relationship. The method is as follows. We define (x,y) as the coordinate on a video frame, and (u,v) is the corresponding real world coordinate. In projective geometry, homogeneous coordinates is used to describe points. Using homogeneous coordinate, the coordinate (x,y) can be represented (x,y,1) and (u,v) can be represented with (hu,hv,h) , h is a homogeneous parameter. A 3×3 transformation matrix can be applied to represent the transformation between these two homogeneous coordinates. When considering rotation, scaling, sheer and translation, the general transformation matrix is defined as:

$$T = \begin{pmatrix} m_1 \, m_2 \, m_3 \\ m_4 \, m_5 \, m_6 \\ m_7 \, m_8 \, 1 \end{pmatrix}$$

Where m_1 m_2 $m_3 m_4$ m_5 $m_6 m_7$ m_8 are parameters representing rotation, scaling, sheer and translation.

The relation between world coordinate and the coordinate on the video frame can be expressed as:

$$\begin{pmatrix} hu \\ hv \\ h \end{pmatrix} = T \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

The four control points' coordinates are used to estimate the transformation matrix. After getting the transformation matrix *T*, the real world coordinate can be calculated according to the coordinate on video frames.

9.5.3.2 Road identification method using road annotation

A clustering algorithm is used for road identification and the Hough transform transformation algorithm is used to detect the edges of a road. The input of the algorithm is the annotation of road in the previous estimated annotated frame and the output is the estimated annotation on current frame. It is not necessary to repeat the process frame by frame. The time span between two successive frames is short, so the change between two successive frames can be neglected. Therefore, we only apply this algorithm every 100 frames and annotations on other frames are estimated by linear interpolation algorithm.

Unlike complex appearance of vehicles, the color of road surface is always homogeneous, so the value of each pixel can be a good feature to distinguish road and non-road part. By applying the k-means algorithm, the pixels are clustered into two clusters according to its pixel value: one contains road surface pixels and the other contains non-road surface pixels. Then Hough transform algorithm for detecting lines is applied to find the edges of a road after some morphological processing of the clustering result. There are three steps in the process: (1) choice of a color space, (2) clustering pixels using the k-means algorithm, (3) detecting edges of road using the Hough transform algorithm.

9.5.3.2.1 Choice of a Color Space

A pixel can be represented in multiple color spaces (e.g. RGB and YUV). We use RGB and YUV color space. In both color spaces, every pixel is represented by three 8-digit binary values. In RGB color space, the three values represent the intensity of the red (R), green (G) and blue (B) color components. In YUV space, the three values represent one luma (Y) and two chrominance (UV) components. They are suitable for different scenarios. For example, representation in YUV color space is more insensitive to variation of illumination compared to RGB color space. The images in RGB and YUV space are shown in Figure A48 respectively.

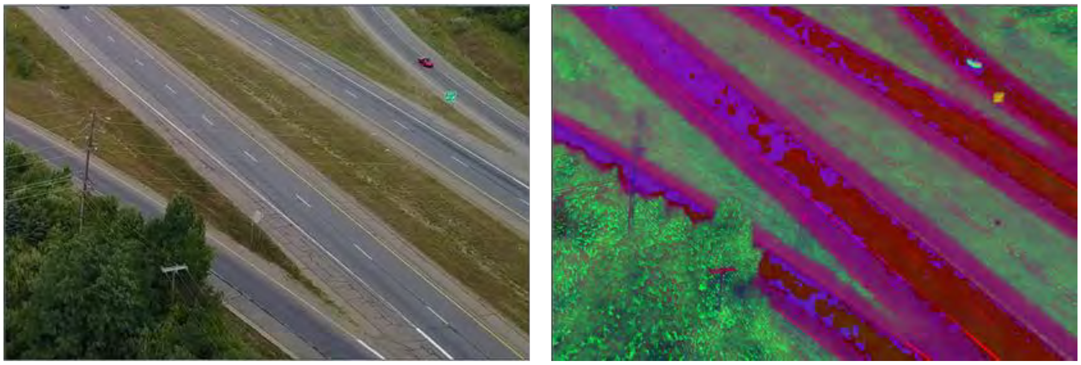


Figure A48: The Image in RGB (left) and YUV Color Space (right).

9.5.3.2.2 The k-means Clustering Algorithm

The k-means is an efficient heuristic clustering algorithm. A pixel will be assigned a label according the nearest centroid. The k-means algorithm aims to minimize the accumulative distance (e.g. Euclidean distance, Manhattan distance, etc.) between data points and centroids. Figure A49 is the clustering result of the k-means algorithm using Euclidean distance.

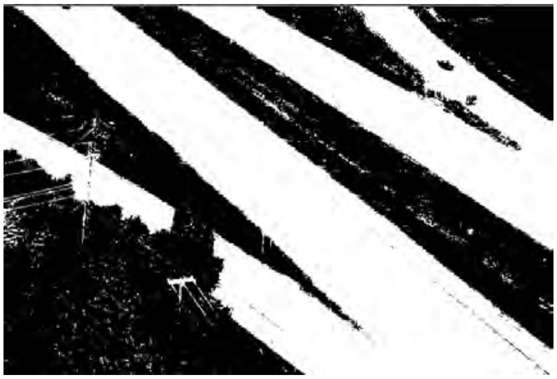


Figure A49: Pixel Clustering Result.

9.5.3.2.3 Road Edge Detection

A short segment of road can be viewed as a straight road, so the edges of the road could be estimated using two straight lines. The Hough transform algorithm can be used to find straight lines in an image. Therefore, it is used for road edge detection in this software.

The Hough transform algorithm may detect two or more lines for every edge in the image. The result is shown in Figure A50. The two lines represented the road are selected by comparing with the road annotation on the former annotated frame. Assuming the video vibration is not severe, then the road edge should be close the current edge. Figure A50 presents the results of the Hough transform algorithm and Figure A50 shows the determined edges on the new frame.

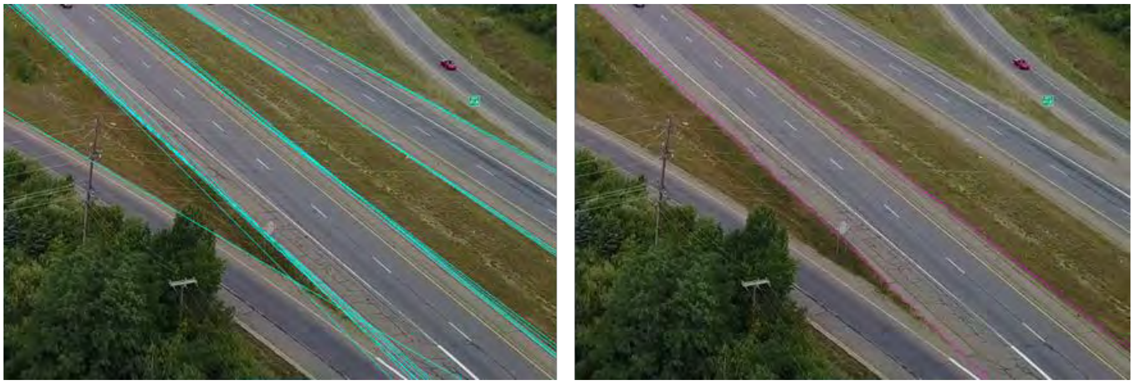


Figure A50: Identified Road

9.56.3.3 Vehicle detection method

Vehicle detection is based on frame-by-frame difference and the mean shift algorithm. There are two steps in the proposed vehicle detection method: (1) frame-by-frame difference image processing, and (2) the mean shift algorithm for vehicle detection.

9.5.3.3.1 Frame-by-frame difference image

Frame by frame difference image is obtained by the subtraction of two successive frames. The difference image need morphologic operation (e.g. erosion and dilation) to remove the noise in the difference image and expand the interested part in the image. The processed frame-by-frame difference image of a vehicle is shown in Figure A51.

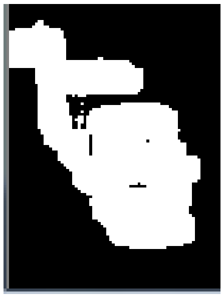


Figure A51: An Example of Processed Frame-by-Frame Difference Image

9.5.3.3.2 Mean shift algorithm

The mean shift algorithm is a non-parametric iterative clustering algorithm. It is a fast algorithm to estimate kernel density of samples. The mean shift algorithm is widely used in object detection. The mean shift algorithm tries to find the position of search window to maximize the total weight of samples in the search window.

The notions used in the algorithm is defined as follows:

- S : The search window. S= $\left(x_{top}, y_{top}, w, h\right)$, x_{top}, y_{top} are the top-left x coordinate and top-left y coordinate of a search window S, w is the width of search window, and h is the height of search window.
- c: The center of a search window S.
- (x, y): The coordinate of pixel in search window $x_{top} \leq x \leq x_{top} + w, y_{top} \leq y \leq y_{top} + h$.
- $I(x, y)$: The value of a pixel(x, y).
-

Mean shift Algorithm in object detection:

Step 1: Calculate zeroth and first moment

$$M_{00} = \sum_{x=x_{top}}^{x_{top}+w} \sum_{y=x_{top}}^{y_{top}+h} I(x, y)$$

$$M_{01} = \sum_{x=x_{top}}^{x_{top}+w} \sum_{y=x_{top}}^{y_{top}+h} yI(x, y)$$

$$M_{10} = \sum_{x=x_{top}}^{x_{top}+w} \sum_{y=x_{top}}^{y_{top}+h} xI(x, y)$$

Where, M_{00} is the zeroth moment and M_{01} and M_{10} are the first moments.

Step 2: Calculate the mass center (x', y') of the search window:

$$x' = \frac{M_{10}}{M_{00}}, y' = \frac{M_{01}}{M_{00}}$$

Step 3: If the center of a search window S, c, equals to the mass center (x', y'), then stop; otherwise, set $c = (x', y')$, and go to Step 1.

The result is shown in Figure A52.

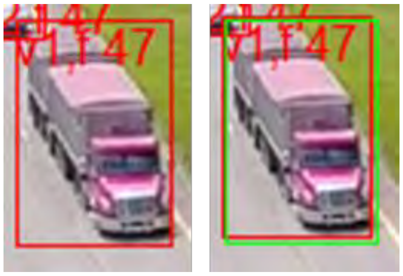


Figure A52: Vehicle Detection Result

9.56.3.4 Traffic data analysis

The software itself also collects data and displays the results on the screen. The software can analyze traffic variables such as the vehicle speed, cumulative number of vehicles entering the road, cumulative number of vehicles exiting the road, traffic in-flow rate, traffic out-flow rate and traffic density.

9.5.3.4.1 Traffic Volume and Flow Rate

Traffic volume is the number of vehicles passing by a specific point during a specific time period. Flow rate is the number of vehicles passing by a specific point during in an hour. Typically, the time period used to calculate the flow rate is 1 minute, 5 minutes, 10 minutes, and 15 minutes. Since our video are almost less than 10 minute, the flow rate is only calculated at the end of the video.

We define N is the number of vehicles and T is the observed time at a point, then the traffic flow is calculated by:

$$q = \frac{N}{T}$$

9.5.3.4.2 Density (Occupancy)

Density is defined as the number of vehicles on a mile.

We define and l is the length of road, and n is the number of vehicles on the road, then the density is calculated by:

$$k = \frac{N}{l}$$

9.5.3.4.3 Space Mean Speed

There are three ways to calculate the space mean speed during the time. The definition of space-mean speed is the ratio between total distance of vehicle and total time of vehicles passing the road. There are two ways to calculate the space mean speed in our software.

We define N is the number of vehicles, l_i is the total travel distance of the ith vehicle, t_i is the travel time of the ith vehicle, v_i is the instantons speed of the ith vehicle.

(1) The space mean speed can be calculated by total distance dividing total travel time:

$$\bar{v_s} = \frac{\sum_{i=1}^{N} l_i}{\sum_{i=1}^{N} t_i}$$

(2) The space mean speed is often calculated by arithmetic instantaneous speed of all vehicles over the road on each frame.

$$\bar{v_s} = \frac{1}{N}\sum_{i=1}^{N} v_i$$

In animation step, the space mean speed is calculated by average instantaneous speed of on-road vehicles. Space mean speed also can be calculated according to the data in traffic variable file.

9.5.3.4.4 Time-space trajectory

Time-space trajectory shows the vehicle trajectory through time as it moves on a road. Some traffic data can be derived from time-space trajectory such as vehicle speed, average density between two points and average flow between two time points. It's also a direct way to reflect some special traffic phenomenon, for example, shock waves. In order to visualize the time-space trajectory, the software will output a file which can be fed into the Mobile Sensor Data Analyzer. Figure A53 is an example of time-space trajectory.

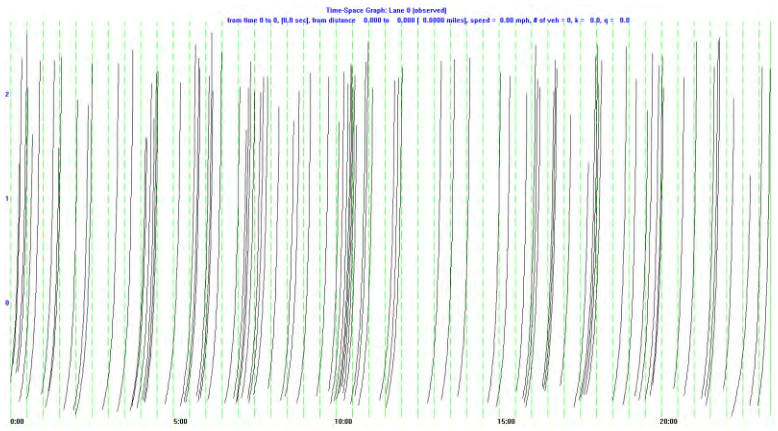


Figure A53: An Example of Time-space Trajectory

9.5.4 The result of traffic monitoring

This section shows the results of US 23 (Ann Arbor) and US31 (Muskegon). Each road has several videos. We deal with these videos separately.

9.5.4.1 Summary of data

The following table is the summary of the data collected and used in the project.

Table A1: Summary of Collected Videos

Video name	Duration	Number of frames	Frames per second	Number of vehicles	Number of roads	UAV status	Loca tion	Road name
DJI_0058.MOV	05:15	9,446	29.97	304	2	static	Ann arbor	US23
DJI_0059.MOV	03:47	6,814	29.97	410	2	static	Ann arbor	US23
DJI_0060.MOV	04:02	7,268	29.97	46	2	static	Ann arbor	US23
DJI_0061.MOV	01:59	3,581	29.97	214	2	static	Ann arbor	US23
DJI_0069.MOV	03:45	6,751	29.97	324	2	static	Ann arbor	US23
DJI_0070.MOV	03:50	6,903	29.97	423	2	static	Ann arbor	US23
DJI_0071.MOV	00:30	924	29.97	62	2	static	Ann arbor	US23

9.5.4.2 Traffic analysis on different roads: US23 (Ann Arbor)

The following table is the summary of videos taken at US23.

Table A2: Summary of videos of US23

Video name	Road No.	Duration (min)	Volume (veh)	Flow (veh/hr)	Space mean speed (mph)	Speed limit(mph)	Road length(feet)
DJI_0058.MOV	1	03:47	85	1348.02	65.97	70	400
DJI_0058.MOV	2	03:47	TBD	TBD	TBD	70	TBD
DJI_0060.MOV	1	04:02	TBD	TBD	TBD	70	TBD
DJI_0060.MOV	2	04:02	TBD	TBD	TBD	70	TBD
DJI_0061.MOV	1	01:59	TBD	TBD	TBD	70	TBD
DJI_0061.MOV	2	01:59	TBD	TBD	TBD	70	TBD
DJI_0069.MOV	1	03:45	TBD	TBD	TBD	70	TBD
DJI_0069.MOV	2	03:45	TBD	TBD	TBD	70	TBD
DJI_0070.MOV	1	03:50	TBD	TBD	TBD	70	TBD
DJI_0070.MOV	2	03:50	TBD	TBD	TBD	70	TBD
DJI_0071.MOV	1	00:30	TBD	TBD	TBD	70	TBD
DJI_0071.MOV	2	00:30	TBD	TBD	TBD	70	TBD

9.5.4.3 Traffic analysis on different roads: US31 (Muskegon)

The following table (and Figures A54) is the summary of video taken at US31.

Table A3: Summary of Videos of US31

Video name	Road No.	Duration (min)	Volume (veh)	Flow (veh/hr)	Space mean speed (mph)	Speed limit	Road length(feet)
DJI_001.MOV	1	09:24	113	721.2766			341

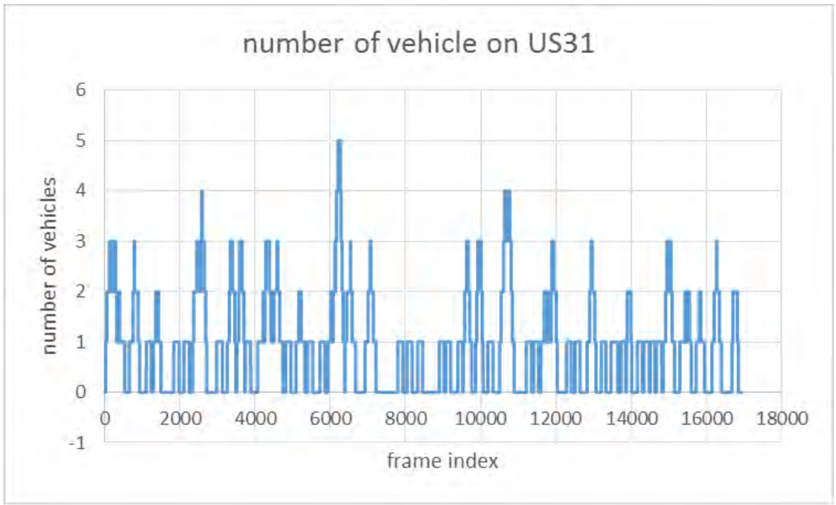


Figure A54: Number of vehicles on road US31

The number of vehicles on road US31 is much smaller than US23. For most time there is only one vehicle on the selected segment of road (Figure A55).

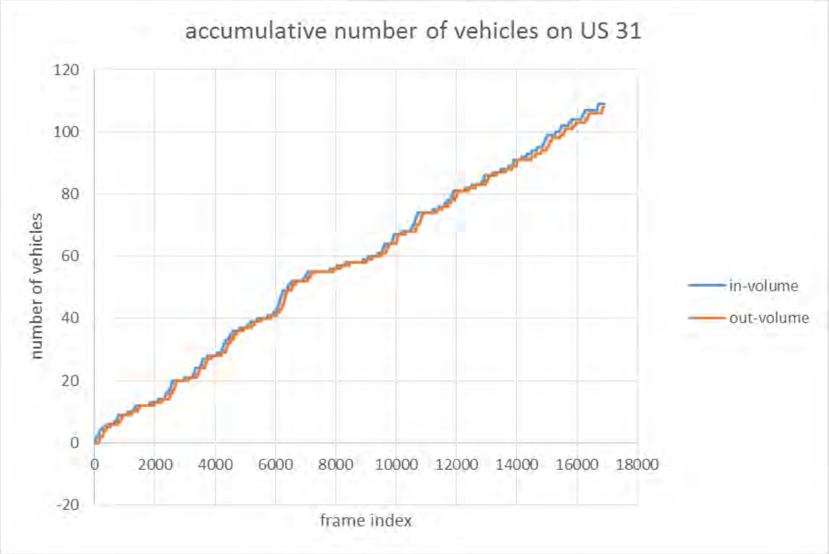


Figure A55: Number of Vehicles on the Road US31 per Frame

There is no other entrance or exit of the selected segment. Therefore, the trends of in-volume and out volume are similar. The difference is the number of vehicle on the road, which represents the density of the road segment.

Future work

The potential future work is to build an automatic UAV video-based traffic monitoring tool that can be used by MDOT traffic engineers to collect traffic data using UAV for daily operations including traffic studies (e.g. volume studies, speed and travel time studies etc.), capacity analysis, traffic simulation calibration and validation, traffic safety analysis, and Intelligent Transportation Systems operations etc. The automatic UAV video-based Traffic monitoring tools requires a large sample of annotation samples from real world for offline deep learning model training.

To create an automatic UAV video traffic data analysis tool, we need to do the following tasks.

1. Collect more UAV videos for reoccurring and non-reoccurring traffic conditions at different scenarios (such as traffic accidents and work zone etc.) and different types of roads, as well as weather conditions.
2. Use the semi-automated traffic monitoring tool in this project to annotate more data samples, especially for some vehicle overlapping cases under stop-and-go traffic conditions. This will create a database for UAV vehicle and road annotation data samples.
3. Develop a fully automatic vehicle detection and tracking algorithm using deep learning and conduct model training.
4. Develop an advanced projection (or geo-registration) algorithm (due to the remote sensing) to provide precise vehicle positions to estimate vehicle speed, size, and distance gap etc., and road information.
5. Validate and calibrate the fully automatic vehicle detection and tracking algorithm for traffic analysis.
6. Consideration of developing an on-line real-time fully automatic UAV video traffic monitoring algorithm that could be developed.
7. Demonstrate deployment of the automatic UAV traffic data analysis tool at a MDOT transportation operations center.