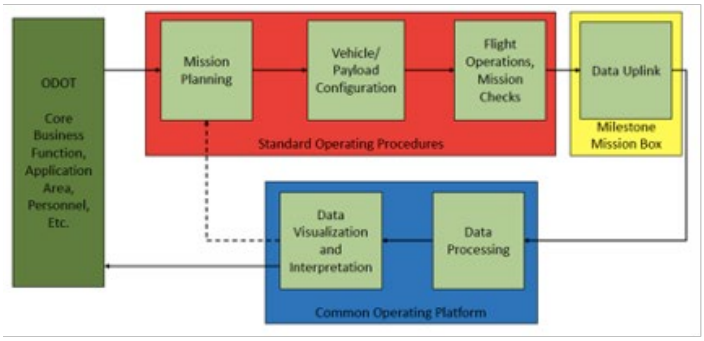
Unmanned Aerial Systems (UAS) for Transportation, Incident Management, and Infrastructure Assessment



Prepared by:
Arthur Helmicki, Victor Hunt, Bryan Brown

Prepared for:
The Ohio Department of Transportation,
Office of Statewide Planning & Research

Project ID Number: 102141

May 2021

Final *Report*

University of CINCINNATI

RESEARCH

U.S. Department of Transportation
Federal Highway Administration

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FHWA/OH-2021-16		

4. Title and Subtitle	5. Report Date
Unmanned Aerial Systems (UAS) for Transportation, Incident Management, and Infrastructure Assessment	**May 2021**
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Arthur Helmicki, Victor Hunt, Bryan Brown, Mahdi Norouzi, Kelly Cohen, Manish Kumar	

9. Performing Organization Name and Address	10. Work Unit No. (TRAIS)
University of Cincinnati **Cincinnati, Ohio 45221**	
	11. Contract or Grant No.
	27899

12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
Ohio Department of Transportation **1980 West Broad Street** **Columbus, Ohio 43223**	**Final Report**
	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

Unmanned aerial systems (UAS) have received significant attention in recent years due to substantial advances in capabilities. The goal for this project has been to work with Drive Ohio and the OH UAS Center: (1) to identify those ODOT core business functions which can be improved/enhanced by application of UAS technologies, (2) to identify the corresponding necessary UAV system configurations and missions and (3) to begin developing, testing, and documenting vehicle system configurations and capabilities to perform these missions. In collaboration with ODOT, the following application areas were selected for in-depth investigation: Bridge and facility inspection, aerial mapping, construction monitoring, and traffic monitoring and management. Additional integrative, crosscutting technologies to support operations were also considered. A variety of UAS vehicle systems were acquired along with associated cameras and supporting hardware and software. In addition, custom hardware and software were dev eloped for systems integration and operation, image processing and tele-remote UAV video streaming. Several hundred hours of flight operations were conducted at test sites as well as in actual ODOT field operational setting across the state of Ohio. The limits of UAS operations and applications in these areas were explored and documented. Based on the results obtained, a set of 7 Standard Operations Procedures (SOPs) were developed and delivered and associated training session were conducted. The research showed that UAS operations are poised to dramatically impact several areas, and that ODOT, through its UAS Center, is poised to capitalize on many of these trends leading to improved operational efficiency, increased safety and mobility, and reduced costs.

17. Keywords	18. Distribution Statement
UAS, UAV, Law Enforcement, Survey	**No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161**

19. Security Classification (of this report)	20. Security Classification (of this page)	21. No. of Pages	22. Price
Unclassified	**Unclassified**	**190**	

Form DOT F 1700.7 (8-72) **Reproduction of completed pages authorized**

Credits and Acknowledgments Page

Prepared in cooperation with the Ohio Department of Transportation
and the U.S. Department of Transportation, Federal Highway Administration

The authors wish to acknowledge the support, assistance, and guidance of the Ohio UAS Center (especially Richard Fox, Jamie Davis, David Gallagher, and Helen McCreary), ODOT Central Office and District personnel (especially Brandon Collett, Tommy Arnold, Michael Brokaw, Dirk Gross, Dave Holstein, Dave Gardner, Sandra Mapel, Jamie Hendershot, Layth Istefan, Jamie Henderson), and our Technical Liaison (Fred Judson) in conducting this research.

We also wish to thank Jill Martindale and Vicky Fout of ODOT's Office of Statewide Planning and Research for their time and assistance.

Finally, we also wish to acknowledge the hard work done by our students (especially, Niranjan Krishnan, Arjun Chiddarwar, Aswin Balasubramanian, Austin Wessels, Rumit Kumar, Sohan Karkera, Nikita Saraf, and Pranav Khekare) in the laboratory and in the field as well as their contributions in supporting the development of the documentation for this project in the form of papers, theses, reports, and standard operating procedures.

Table of Contents

1.0 Problem Statement/Project Objectives

Unmanned aerial systems (UAS) have received significant attention in recent years due to substantial advances of their capabilities and the wide range of applications to which they have been applied. In addition, changes made in 2016 on the part of the FAA to the laws and regulations which govern the use of these technologies has also radically changed to landscape and rapidly accelerated the availability, implementation, capabilities, and ease of use of UASs across a broad swath of the commercial market as well as society in general. The term drone is now commonplace and whole new economic sectors of goods and services which were previously unthought of are now emerging. Within this setting, the goal for this project has been to help ODOT capitalize on this technology opportunity by working with ODOT's UAS Center (Figure 1):

(1) to identify those ODOT core business functions which can be improved/enhanced by application of UAS technologies,

(2) to identify the corresponding necessary UAS system configurations and missions, and

(3) to begin developing, testing, and documenting prototype vehicle system configurations and capabilities to demonstrate how these missions can be performed.

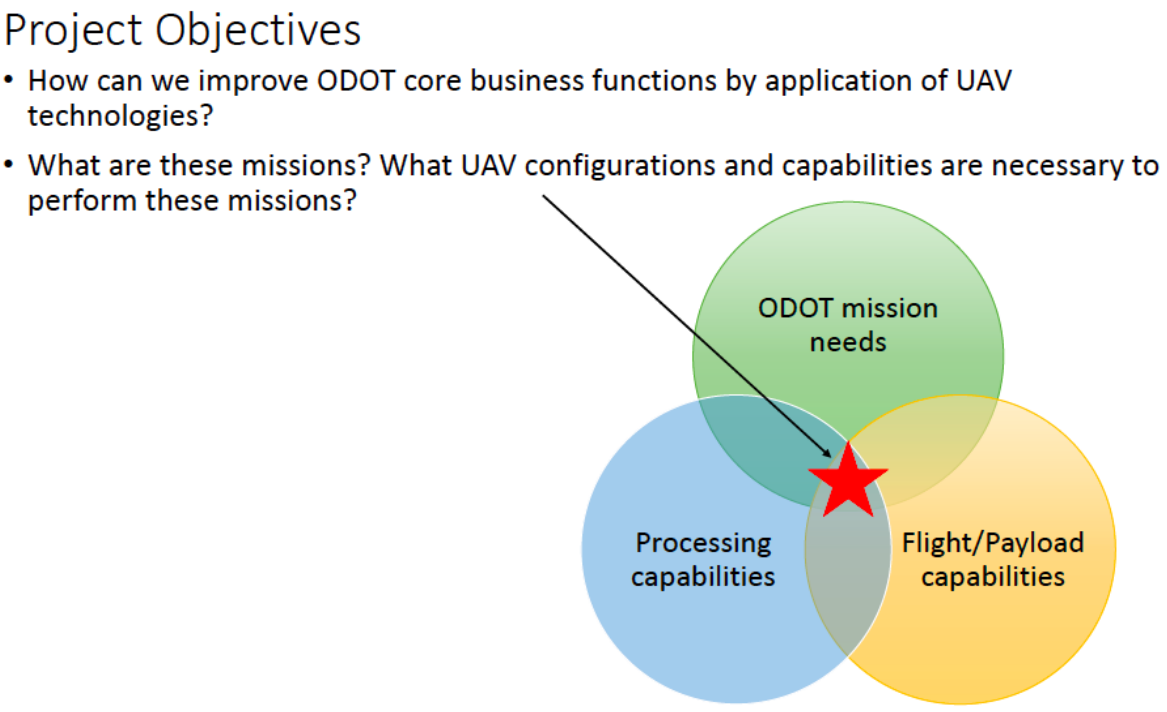


Figure 1: Project Objectives

2.0 Research Background and Approach

2.1 Research Background

Before we begin with a detailed description of the project and its findings and recommendations, we provide here a brief history in order to put this report and its contents into context. At the time ODOT published its original RPF for this work in 2015, the FAA had not yet issued its Part 107 revisions. This notwithstanding, there was a keen interest in and understanding of the fact that airborne sensor platforms had the potential to greatly assist with and enhance capabilities of the DOT in certain transportation-related activities (e.g., situational awareness, construction site/infrastructure observation and monitoring, incident management, assessment and inspection, etc.). Indeed, ODOT maintains a fleet of manned aircraft with specialized payloads to facilities, among other tasks, aerial mapping. Unmanned airborne platforms have additional distinct advantages in terms of rapid deployment, mobility, ease of use, cost, maintenance, etc.

Correspondingly, against this backdrop, the original research plan was to target the use of Moored Aerostat Systems (MAS), which fall under the category of lighter-than-air systems, coupled with various specialized sensor packages/payloads as a means for investigating such task areas of interest to the DOT. Being tethered, such a system also had advantages in terms of FAA regulation and its coordination/operation within the existing airspace.

In fact, in response to this opportunity, the Ohio Department of Transportation (ODOT) had acquired a moored aerostat, or "Blimp in a Box" (BiB), from Lighter Than Air Systems (LTAS) (then named the Drone Aviation Corp) [1]. This system was thought to be flexible enough for multiple inflation/deflation cycles but rugged enough for long term deployment and had the ability to remain stationary in reasonable weather conditions.

The MAS model BiB250 was acquired by ODOT in 2013 from LTAS. Subsequently, LTAS changed its name to Drone Aviation Corp and the BiB250 was renamed the WASP. The Bib250 system, shown in Figure 3, had 3 basic subsystems and the following technical specs:

- Aerostat
    - Inert helium lift gas
    - 15ft diameter nylon and polyurethane bladder
    - Days flight duration
    - 1,000ft operating altitude
    - Payload lift capacity 20-45lbs
- Launcher
    - Setup to altitude in 20min
    - Recovery in 6 min

    - Simple deployment 2-3 operators
    - Winched tether with power and data lines and Kevlar sheathing
- Payload
    - Dual gimbaled, auto stabilized sensor platform
    - Electro-optical+LWIR thermal camera in harsh weather enclosure
    - Portable laptop ground control station with 72hr DVR
    - 5°-35° field of view with 7x zoom
    - 640x512 pixels
    - 3-5km detection range for humans and vehicles

Figure 3: LTAS/Drone Aviation Bib250/WASP system [1]

In addition to the base system several upgrades were available, including the BiB300 Day-Night camera with integrated laser rangefinder and a communications payload. Drone Aviation is also provided product support in the form of engineering to interface custom, third party payloads as well as on-site field operations training [1].

Correspondingly, the purpose of original RFP was to provide ODOT with the engineering research and development necessary to enable effective and efficient capability for the deployment of this platform with selected mobile sensors packages capable of short suspense response for monitoring transportation infrastructure during project planning, construction monitoring, search and rescue operations, system and highway incident management. The research project was therefore intended to comprehensively develop the existing ODOT MAS with additional sensors, loggers, communications, and related equipment, operating protocols, and programming such that it may be readily deployed statewide by an ODOT team to facilitate data collection with its payload of sensors, immediate processing of this data to actionable information, and/or and coordinate with ODOT field personnel and administrators in a timely manner.

While the core focus of the project, as originally envisioned, did not change, the advent of FAA's Part 107 guidelines [2,3,4] in 2016 radically shifted the focus away from moored aerostat systems (MAS) and towards achieving the stated goals using the rapidly evolving and much more capable area of unmanned aerial systems (UASs) or unmanned aerial vehicles (UAVs).

The FAA Modernization and Reform Act of 2012 [2,3,4] was the first true push to allow for non-recreational UAS flights. Up until this point public entities could fly under Certificates of Waiver or Authorization (COAs), but this process was lengthy and prohibited flights in many locations. The Reform Act set in motion that the FAA would be required to "develop a comprehensive plan to safely accelerate the integration of civil unmanned aircraft systems into the national airspace". In addition, the FAA would need to create new UAS laws for commercial users by 2015.

Section 333 of the FAA Modernization and Reform Act of 2012 allowed commercial UAS operations on a case-by-case basis. Section 333 was a way to grant approval for UAS operations while FAA gathered information and was able to begin rule making for what would become 14 CFR Part 107.

This project was introduced during the introduction of Part 107, which was released in February of 2016. While using an aerostat had advantages to monitoring traffic and carrying payloads, Part 107 allowed for the use of sUAS (<55lb UAS) to be operated in a freer way than under the previous COA or Section 333 approvals.

Part 107 also introduced many restrictions that as the project would go on would in turn become waivable under the right circumstances. Some of the highlighted rules include:

- Unmanned aircraft must weigh less than 55 lbs. (25 kg).
- Visual line-of-sight (VLOS) only; the unmanned aircraft must remain within VLOS of the remote pilot in command and the person manipulating the flight controls of the small UAS. Alternatively, the unmanned aircraft must remain within VLOS of the visual observer.
- Small unmanned aircraft may not operate over any persons not directly participating in the operation, not under a covered structure, and not inside a covered stationary vehicle.

- Daylight-only operations, or civil twilight (30 minutes before official sunrise to 30 minutes after official sunset, local time) with appropriate anti-collision lighting.
- Must yield right of way to other aircraft.
- May use visual observer (VO) but not required.
- Maximum altitude of 400 feet above ground level (AGL) or, if higher than 400 feet AGL, remain within 400 feet of a structure.
- Minimum weather visibility of 3 miles from control station.
- Operations in Class B, C, D and E airspace are allowed with the required ATC permission.
- Operations in Class G airspace are allowed without ATC permission.

As stated above, many of these restrictions are waivable and during this project the team would act on those waivers.

The most used waivable restriction during this project was that of operations in Class B, C, D, and E airspaces. The first waiver was that used in this project was a Class D airspace waiver. This waiver process has also been improved over the last several years. The waiver process took approximately 3 months to complete. The FAA introduced the LAANC program to help enable UAS integration into the airspace. LAANC or Low Altitude Authorization and Notification Capability provides UAS pilots access to controlled airspace at or below 400 feet, awareness of where pilots can and cannot fly, and Air Traffic professionals with visibility into where and when UAS are operating. LAANC was also the first introduction of UAS Service Suppliers (USS) a building block of the Unmanned Traffic Management (UTM) system. LAANC now provides near real-time authorization around controlled airspace. This has been used multiple times during the project for monitoring traffic around the state.

The sUAS that were used for the project ranged simple flying cameras to complex inspection capable machines. The start of this project would see the use of such UA as the DJI Matrice 100 which at the time of this writing has seen 2 newer versions in the Matrice 200 and Matrice 300. The technology in these systems has improved the ease of operation as well and many of the first vehicles used have become outdated and lack the same user-friendly features that newer systems can provide. This technology improvement is also evident in the market survey conducted and show in section 3.1. Many companies saw Part 107 as a new market for them to introduce new systems and become a world leader in UAS; many have come and gone.

2.2 Research Approach

In the process of conducting this research project, the research team has strived to help the UAS Center develop a view of the application and implementation of UAS's as an end-to-end solution which includes not only consideration of the prototype vehicle system and mission demonstrations, but also includes such considerations as mission planning and flight operations particulars, data handling

(e.g., communication, storage/archival, processing and interpretation/visualization) resulting in the development of a series of Standard Operating Procedures (SOPs), additional hardware (e.g., Milestone Mission Box), and a customized, web-server based software processing platform (e.g., Common Operating Platform) which function cohesively to help the UAS Center meet ODOT needs in a number of specific application areas (e.g., bridge and facility inspection, aerial mapping and construction site monitoring, traffic monitoring and management, etc.). This report attempts documents these many facets of the research project.

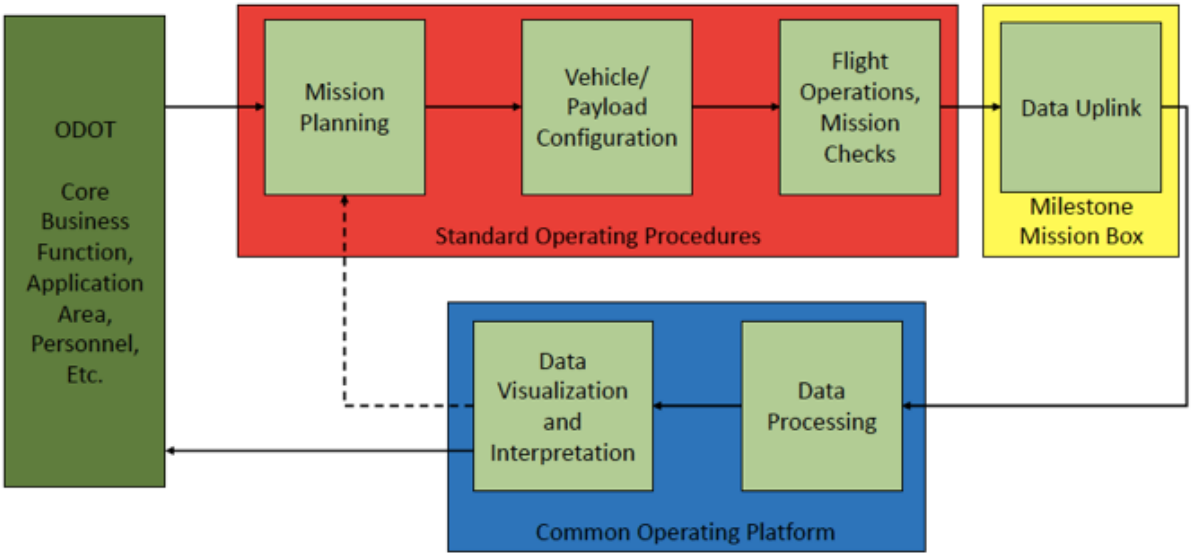


Figure 2: Project end-to-end research approach

Commensurate with the objectives outlined above in Section 1 and the Background given in Section 2, the research team followed a 4-phase approach in conducting the research:

Task 1: Identify situations within ODOTs operations that warrant use of airborne data and identify appropriate hardware and software needed to obtain, process, and interpret flight data, including flight operations considerations.

Here the research team relied on input from the Ohio UAS Center Technical Advisory Committee as well as meetings and discussions with ODOT personnel at all levels in in all areas of interest to ODOT's mission. This included personnel all the way from Central Office, District Offices, down to County Garages.

In addition, the research team conducted an extensive market survey in order to capture the state of the UAV market as of 2016 in terms of capabilities. This included an understanding of hardware (i.e., vehicles, sensors, camera systems, etc.), software (Flight control and planning, image processing, 2D and 3D modelling, etc.).

Details of these activities and findings are contained in Section 3.1.

Using the data above, the research team worked with Ohio UAS Center personnel to identify selected mission areas for further in-depth investigation. As detailed below in Sections 3.2-3.4 these included:

- Bridge and facility inspection,
- Aerial mapping and construction site monitoring, and
- Traffic monitoring and management.

Additional integrative, crosscutting technologies to needed support these operations were also considered (sees Sections 3.2, 3.5-3.6):

- Common Operating Platform (COP), a software platform to allow collection, storage, exchange and processing of UAV camera data across ODOT offices and across UAV hardware types.
- Augmented reality such as the Microsoft Hololens in order to aid in data visualization.
- Milestone Mission Box to allow tele remote viewing of real-time UAV video using ODOT's Milestone traffic monitoring platform.
- Investigation of AI and computer vision capabilities in automating the interpretation of UAV traffic video and for the extraction of higher level traffic parameters such as count, flow, density, headway, etc.

Task 2: Acquire the hardware and develop supporting software and integrate into prototype systems. Both tethered and untethered systems and commercially-off-the-shelf (COTS) and custom built systems were considered.

After consultation with Ohio UAS personnel, several UAS vehicle systems were acquired for further testing and investigation. As detailed in Section 3.1, these included:

- 3 DJI Matrice 100's which allowed a great deal of hardware and software configurability and interoperability.
- 1 DJI Matrice 210 RTK which provided a great deal of configurability in camera placement, proximity sensing for flying in tight quarters, and advance GPS capabilities.
- 2 Intel Aero's which provided and open flight control architecture.
- 1 custom UAV developed and fabricated with shrouding and 2 independent cameras in order to operate in closed and dark spaces.
- 1 Hoverfly tethered UAV based on the Typhoon Yuneec vehicle platform
- 1 Powerline tethered system which made use of one of the DJI M100's.

In addition, several Zenmuse optical (still and video) and infrared (IR) cameras, were acquired for pairing with the fleet of DJI UAV vehicles. Finally, 3 UAS image processing and modelling software packages were investigated:

- Pix4D MapperPro and Engine
- Bentley Context Capture
- OpenDroneMap

Task 3: Test and demonstrate these prototype systems for select identified applications.

Several hundred hours of flight operations were conducted at test several sites as well as under actual ODOT operational/field settings. See details in Sections 3.2-3.6. This included operations in 7 of ODOT's districts as well as in conjunction with ODOT personnel from the areas of traffic, structures, operations, construction, CADD mapping/surveying, and the UAS Center. The limits of UAS operations and applications in the selected areas were explored and documented.

Task 4: Develop procedures for system implementation and produce training materials.

Based on the results obtained from R&D efforts above, a set of 7 SOPs were developed and delivered and associated training session were conducted. The SOP's are contained as Appendices in this report and outline in detail the results, findings and recommendations resulting from this research.

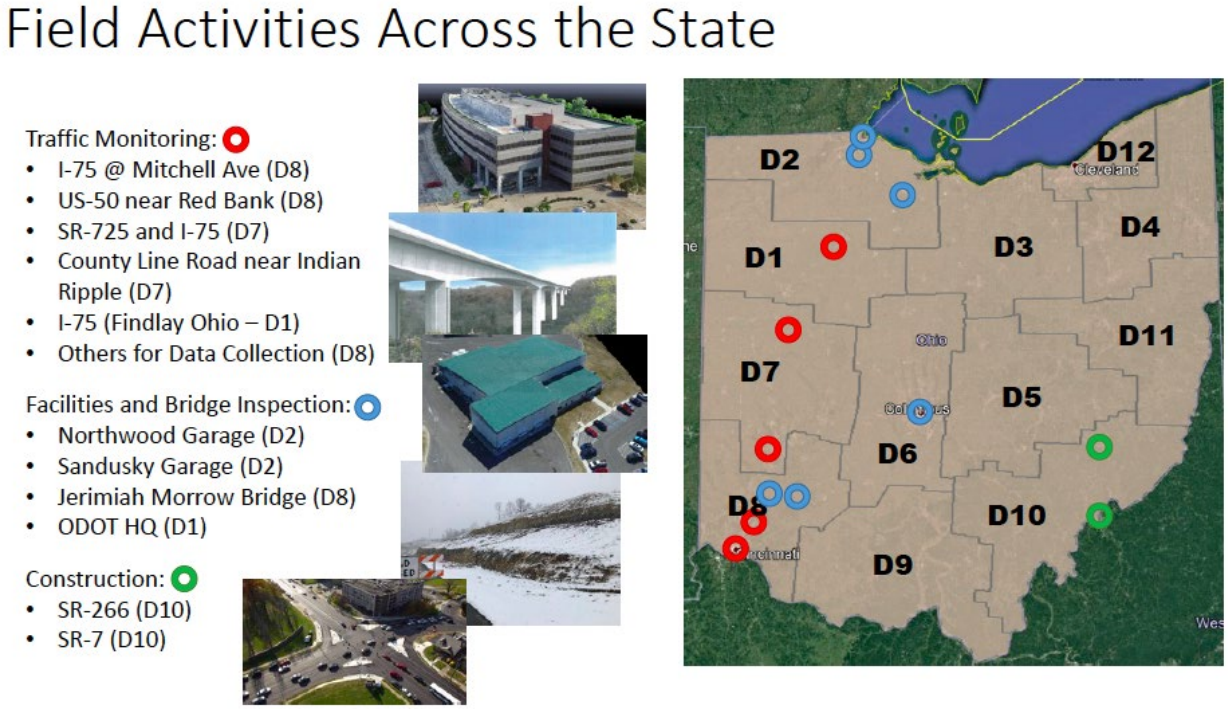


Figure 4: Snapshot of research field activities conducted across Ohio.

3.0 Research Findings and Conclusions

3.1 Market Survey

As described in Section 1 above, especially Figure 1, the goal for this project was to explore the application of UAS technologies in a way that was informed by and synergistic with both ODOT core business functions as well as current and near term UAS capabilities. Further, we were tasked with this effort under limited time and resources. As such, the first task was to identify both desirable mission areas and technology readiness. This was accomplished by conducting surveys in both the transportation area and the marketplace. The results of these efforts will be described in this section.

3.1.1 ODOT Mission Profile Planning

As discussed above, the possible range of applications area for UAS is extremely large and growing on a daily basis. In order to identify application areas that might provide the best return on investment for this project, the research team, working with Ohio UAS Center personnel developed a survey at was distributed to three target audience: Officials at sate DOT across the country, an expanded set of administrators within ODOT, and members or the Ohio UAS Center's Technical Advisory Committee (TAC). The latter two of these groups constitute key stakeholders for this project.

The survey was developed jointly between the research team the Ohio UAS Center and distributed via Survey Monkey in the December 2016-January 2017 timeframe. The survey requested respondents to rate the usefulness of UASs, various sensor packages, etc. in their respective areas as well as to rate the level of importance UASs have in supporting various DOT functions. Responses were gathered, analyzed, and presented to ODOT and the UAS Center TAC in February 2017 where the findings were discussed and an initial set of target mission profiles were selected for further investigation and focus as part of this project. Some key interesting observation emerged from this process.

First, survey responses regarding application areas tended to fall into loose categories which we were able to aggregate into the following 11 mission profiles:

1. **Aerial photography and GIS** – use of UAVs in obtaining aerial imagery such as photo, video, optical, IR, etc., possibly indexed for use with geographical information systems.
2. **External inspection** – use of airborne vehicle to support, augment, and/or replace manual external inspection of bridges, facilities, etc.
3. **Interior inspection** – use of airborne vehicle to support, augment, and/or replace manual internal inspection of buildings, bridges, etc.

4. **Traffic monitoring** – use of airborne video to obtain and extract information and parameters relevant to traffic and roadways thereby providing alternatives to embedded roadway sensors.
5. **Construction site inspection** – use of airborne vehicles in monitoring construction sites possibly in combination advanced GPS in order to obtain survey grade models
6. **Remote inspection** – use of airborne vehicles and cameras to inspect remote areas, especially flying in autopilot and beyond visual line of site (BVLOS).
7. **Accident scene and Quick-clear** – use of airborne vehicles and cameras in detecting, monitoring and documenting accident sites. This would include facilitating ODOT Quick-clear program by monitoring the time it takes tow truck crews to respond and clear roadways.
8. **Search and rescue and disaster events** – use of airborne vehicles to support search and rescue operations. This might include optical as well as IR imagery and like mission 6 above, could include flying in autopilot and BVLOS.
9. **Fire monitoring and hazardous materials** – use of airborne vehicles and cameras to support first responders in various situation such as fire and hazardous materials where humans could be in danger.
10. **Lidar** – use of airborne vehicle as a platform to carry and operate Lidar scans of various sites such as construction sites, structures, etc.
11. **Communications/MARCS** – use of airborne vehicles to implement and maintain ad hoc communications networks. This could be used in conjunction with mission profiles 7, 8, and 9 above.

Second, based on further discussion about mission needs and vehicle capabilities, these prototype mission profiles were subsequently clustered into mission groupings based on common platform needs, piloting, etc. The latter issue of vehicle/payload selection was also examined via a market survey. The results of this activity is discussed in detail below in the remaining sections of Section 3.1 of this report. The following specific mission groupings were identified and are depicted in Figure 5:

Group1: Missions 1, 2, 5, and 6 all requiring a vehicle platform that could be flown manually or in pre-programmed modes and equipped with either image or video cameras

Group 2: Missions 4, 5, and 7 which could be accomplished using a tethered vehicle platform with camera capable of loitering aloft in one spot of long periods.

Group 3: Mission 3 (interior inspection) which would require special navigation skills due to the operation in GPS denied environments.

Group 4: Mission 10 (Lidar) which requires significant basic engineering to mate/merge/interface Lidar systems with UAS platforms, synchronize Lidar and GPS data, and work through operational details.

Other: Missions 8, 9, and 10 which as connected with first responders and may have specialized vehicle platform and/or payload requirements (e.g., chemical sensors, high temp, flying BVLOS, etc.)

Third, it was determined that Groups 1 and 2 would provide the most immediate benefit to ODOT, seem to be capable of being conducted with a clearly defined set of platforms and payloads, and could reasonably to explored within the timeframe of this project. It was determined the natures of Groups 3 and 4 should necessitate them being be considered for their own targeted research projects. The Other group, while important, was determined to have the lowest priority for this project. Further, additional subsequent discussions and preliminary research identified Mission 7, accidents/Quick-clear as needing additional information and resources in order to properly tackle (e.g., autonomous UAV in a roadside box, coordination to ODOT's Traffic Management Center and Milestone camera system, etc.) and so were subsequently moved down the list of topics considered as part of this research effort.

Finally, the team identified the need for a software platform which could operate across all UAV manufacturers; have the ability to accept, archive, and transfer large image data sets within and across ODOT offices and districts; and be able to support the processing and interpretation of UAV data (e.g., generation of 2D and 3D models, etc.) as a key success factor for the incorporation of UAVs and UAV operations across the ODOT organization. As such, the research team was also tasked with developing a prototype software platform which could operate with the ODOT IT ecosystem. This platform came to be known as the Common Operating Platform (COP).

In the sections which follow, the results of several application areas that were explored are reported on. These include: traffic monitoring, construction site mapping and aerial mapping, Inspection of bridges and facilities, development of the Common Operating system, and a set of other integrative technologies that were explored (e.g., virtual reality and off site real-time video streaming).

In order to meet these mission profiles, the team next turned to a survey to market. At the time this project began in 2016, over 300 companies were involved with sUAS hardware and software. Since then, this number has since grown several-fold and many more companies now produce their own systems. The sUAS market is constantly shifting with new companies, sensors and intelligence being developed each week. Many of the companies included in our initial market survey have come and gone. Our initial requirements for sUAS platforms included four main considerations:

1. how easy the sUAS would be to be operate,
2. how quickly we could deploy it,
3. how much maintenance would be involved and
4. how weather ready the systems were.

To narrow down our selections the team limited the sUAS selection to companies that manufactured and sold complete systems that were ready to fly. The

team also only included multirotor systems as needing quick deployment cut many fixed wing aircraft. Our final version of the market survey included 54 sUAS platforms and included non-tethered and tethered systems. Figure 6 shows an overview of the final market survey.

			Platform	Test Location	Application Location	Subject Matter Expert
Focus	Group 1	Mission 1: Aerial Photography/GIS	DJI Matrice 100/210, VGCS Prototype	Summit Park, Wilmington Airpark, MUTC, UC Victory Pkwy Labs	Sites TBD by Subject Matter Experts, candidate sites under discussion	Fred Judson, TAC, Mike Brokaw, Brandon Collett, Stephan Mather
		Mission 2: Exterior Inspection				
		Mission 5: Construction Site Inspection				
		Mission 6: Remote Inspection				
		Mission 7b: Accident Scene				
		Platform Development, Sensor testing and Software integration	Intel Aero, Flamewheel	UC Victory Pkwy Labs, Summit Park	Wilmington Air Park	Fred Judson
	Group 2	Mission 4: Traffic Monitoring	Hoverfly Typhoon	Summit Park, Wilmington Airpark, MUTC, UC Victory Pkwy Labs	Sites TBD by Subject Matter Experts, candidate sites under discussion	Fred Judson, TAC, Carl Merckle, John McAdam, Bryan Comer
		Mission 5: Construction Site Inspection				
		Mission 7a: Quick Clear				
	Group 3	Mission 3: Interior Inspection	VGCS Prototype, Intel Aero, Hybrid	UC Victory Pkwy Facility	JM, FWW, Other	Fred Judson, TBD
	Group 4	Mission 10: LIDAR	TBD	Wilmington Airpark	TBD	Fred Judson, TBD
	Other/Secondary	Mission 8: Search and Rescue	Matrice, Hoverfly, Aero, other	UC Victory Pkwy Labs, Summit Park, Wilmington Airpark	WVa, MUTC, Other	Fred Judson, TBD
		Mission 9: Fire Monitoring & Chemical Cloud Detection				
		Mission 11: MARCS/Communications				

Figure 5: Mission profile survey results and focus areas.

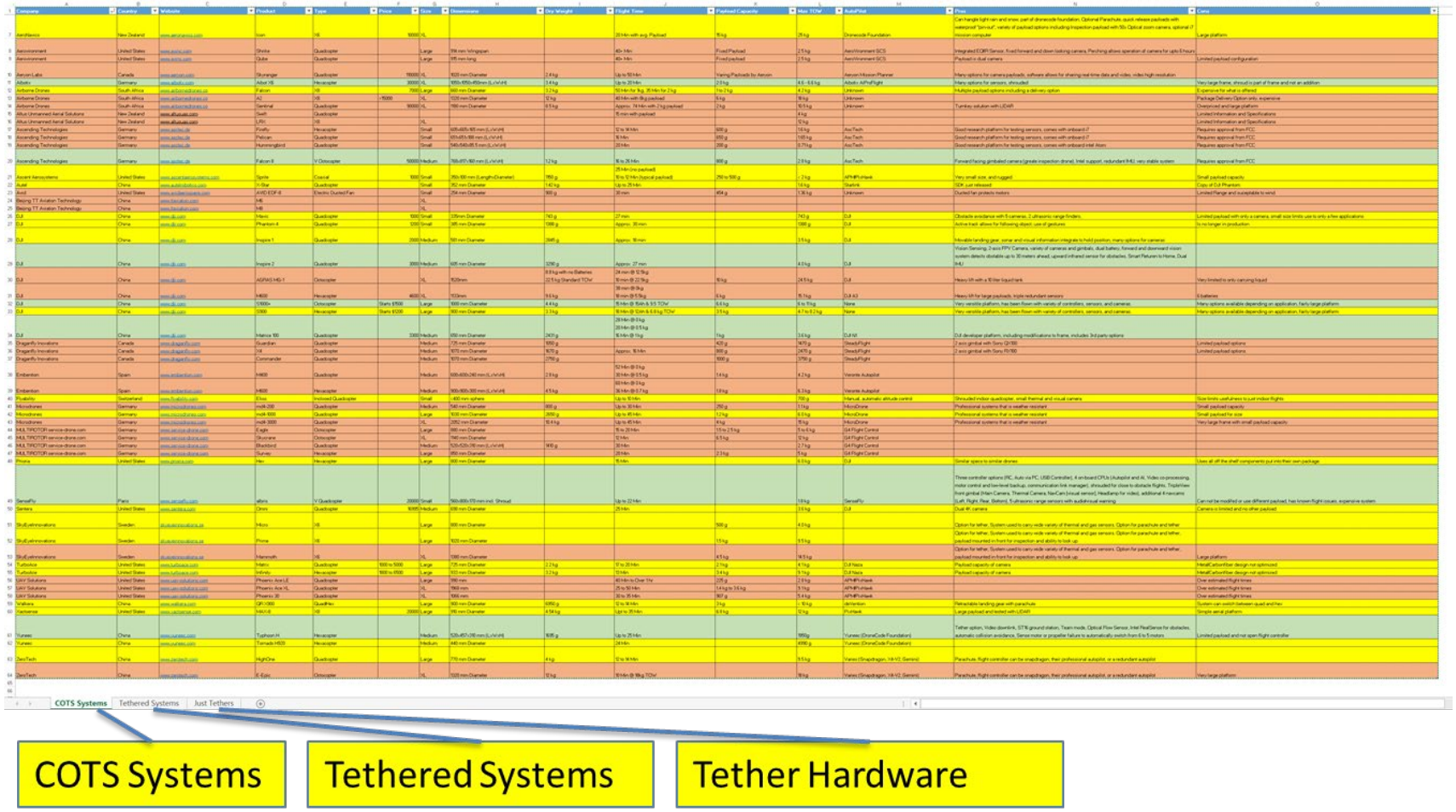


Figure 6: Market survey spreadsheet.

Particular specifications and features of several of these sUAS helped narrow the selection. A few of these included:

- Sizing (Dimensions, Takeoff Weight)
- Payload (Integration, Capacity, Range of Options, Placement, Field Swappable)
- Flight Time
- Autopilot and Level of Autonomy
- Country of Origin

3.1.2 UAV Vehicle Platform - Untethered

The first sUAS selected would be our primary systems for the project capable of free flight. We also determined that having developmental platforms may allow for the team to add new features such as obstacle avoidance and swarming. These developmental platforms would be some of the first used and included the DJI Matrice 100 (see Figure 7, [21]) and the Intel Aero. Other systems were selected to be the primary systems used throughout the project and included the senseFly Albris and AseTec Falcon 8+. Due to price, availability, and manufacturer rep demonstrations these platforms were not selected. The AseTec Falcon would soon be acquired and rebranded under Intel. The Matrice 100 showed that it was capable of performing many of the missions identified above in Figure 5 and would turn from being a developmental platform to the main system used throughout the project. With a relatively inexpensive price and a variety of payloads (including thermal and visual cameras, see Section 3.1.4 below. The DJI Matrice 100 became a powerful tool.

The team would use a total of 3 Matrice 100s on this project. The Matrice 100 used either TB47D or TB48D batteries. For this project the team used mostly the TB48D batteries as they provide a longer flight time of approximately 20 minutes. This sUAS has a maximum takeoff weight of 3600g. The vehicle also had a hovering accuracy of 0.5m vertically; 2.5m horizontally.

Figure 7: DJI Matrice 100 platform [21]

The team would revisit the selection of other systems later in the project as the market was always evolving and new systems may become available to meet many of the mission requirements. One of these systems would be the DJI Matrice 210 RTK. The Matrice 210 RTK had the ability of carrying dual cameras, useful for such application as traffic monitoring, and a camera mounted top, useful for such applications as bridge inspections.

The Matrice 210 RTK (see Figure 8, [21]) had several advantages over the Matrice 100. These came mostly from a built in RTK GPS, ADS-B receiver, dual battery system, multiple payload configurations (dual and top mounted), a FPV camera and IP43 weatherproofing. The Matrice 210 RTK used either TB50 or TB55 batteries. For this project the team would use mostly the TB55 batteries as they offered a maximum flight time of approximately 30 minutes. This sUAS has a maximum takeoff weight of ~6000g. The RTK allows the 210 RTK to have a hovering accuracy of 0.1m vertically; 0.1m horizontally with the RTK enabled.

Figure 8: DJI Matrice 210RTK platform [21]

Other systems would be used by the research team that were used by the UAS Center. These included an Intel Falcon 8+ and Flyability Elios (see Figure 9, [22]). The Flyabilitiy Elios was used mainly toward the end of the project and allowed for up close inspections of bridges. The Elios is a unique sUAS as it has a protective frame allowing for collision tolerance. This frame allowed the Elios to fly next to bridges to inspect cracks or other small features up close. The Elios is equipped with a HD video, thermal imagery, onboard lighting for inspection of dark environments.

Figure 9: Elios Flyability platform [22]

3.1.3 UAV Vehicle Platforms - Tethered

As mentioned previously the project was first to use a moored aerostat. The idea of using a tethered vehicle remained throughout this project to offer a longer endurance sUAS. Similar to the free flight systems, a market survey was conducted. This market was filled with many unknows and companies just starting out. As such the tethers ranged from complete systems to selling just the tether that would connect to an off-the-shelf sUAS. A few of the specifications that would be looked at for the tethers included:

- Weight of the total System
- Payload Capacity
- Max Flight Duration
- Maximum Altitude
- Power Required

There was also a wide variety of price for these systems ranging from $20,000 to $300,000. Many of the tethers originally selected were not used due to their price.

Ultimately, a HoverFly system paired with a Yuneec Typhoon H was selected. The system was lightweight, could fly at 150ft. In addition, by swapping out battery packs, it would allow the team to use the Typhoon H in a non-tethered configuration as well. The team would conduct a variety of tests prior to deploying the tether out in the field. One of the concerns of using a COTS sUAS with the tether was that COTS sUAS are typically not designed to fly for more than 30 minutes or so without landing. Motors, onboard electronics, and on board sensors are typically speced in assuming brief flight operations determined primarily by battery duration. An endurance test

would help determine what issues would be experienced in the field. This did cause issues during testing as the system only had a flight time of approximately one hour forty-five minutes before it would overheat and fail. The manufacture replaced the system, but this failure would be repeated again.

The team would abandon the use of the HoverFly system and looked for new options. A similar system would be purchased from Powerline (see Figure 10, [23]). This system would use a Matrice 100. This was ideal since the team was already familiar with the M100 platform. The Matrice 100 would need to be modified to be equipped with a power regulator and counterweight and so another system was purchased that would permanently have this power regulator attached. This Matrice 100 would still be operated in free flight as needed, but would see a reduction in flight time owing to the additional takeoff weight of the retrofitted tether components. The entire system was powered by a 1000W light weight, portable, gas powered generator.

Many flight tests were conducted in order to insure no issues would occur to this sUAS during operation. An endurance flight of 4 hours was the indication that the system would not have the issues experienced with the previous tether. The manufacture would also update the backup battery system used later in the project providing additional layers of safety.

During the initial testing it was noted that the vehicle would experience changes in altitude during longer flights. This altitude change was caused by the initial barometer calibration. Upon startup the vehicle would automatically calibrate the barometer so an accurate altitude above the ground would be given. As mentioned previously these commercial systems were not designed to fly more than 30 minutes. The longer flights would see significant changes in ambient pressure and/or temperature throughout the day and in turn the vehicle would report the incorrect altitudes. This change was most evident in morning flights that extended into the afternoon as the vehicle would descend due to pressure/temperature changes. As long as the vehicle was not operating poorly the team would continue to use the tether. To verify the altitude during these longer flights a marker would be added to the tether cable so the operator can visually see if the sUAS is at the correct altitude.

This tether system would first be equipped with the Matrice 100's standard payloads of either a Zenmuse X5 or Zenmuse Z3. A modification would be added to allow a Zenmuse Z30, 30X optical zoom, to the platform. This would prove useful for traffic monitoring where the tether would be used most.

Figure 10: Powerline/M100 tethered platform

3.1.4 Cameras/Payloads

The team also surveyed various sensor payloads that could equip the potential sUAS used. Figure 11 shows some of the categories that we looked into. These payloads ranged from visual cameras, thermal cameras, chemical sensors, communications equipment, onboard computers, 3D Lidar systems, etc. Many vehicles on the market have their own payloads that can be used and many of the third party payloads may be difficult to integrate, operate, and control. The team used this survey as more of a guide to see what payload options were available, but most of the platforms used throughout the project had their own specific cameras that could be used.

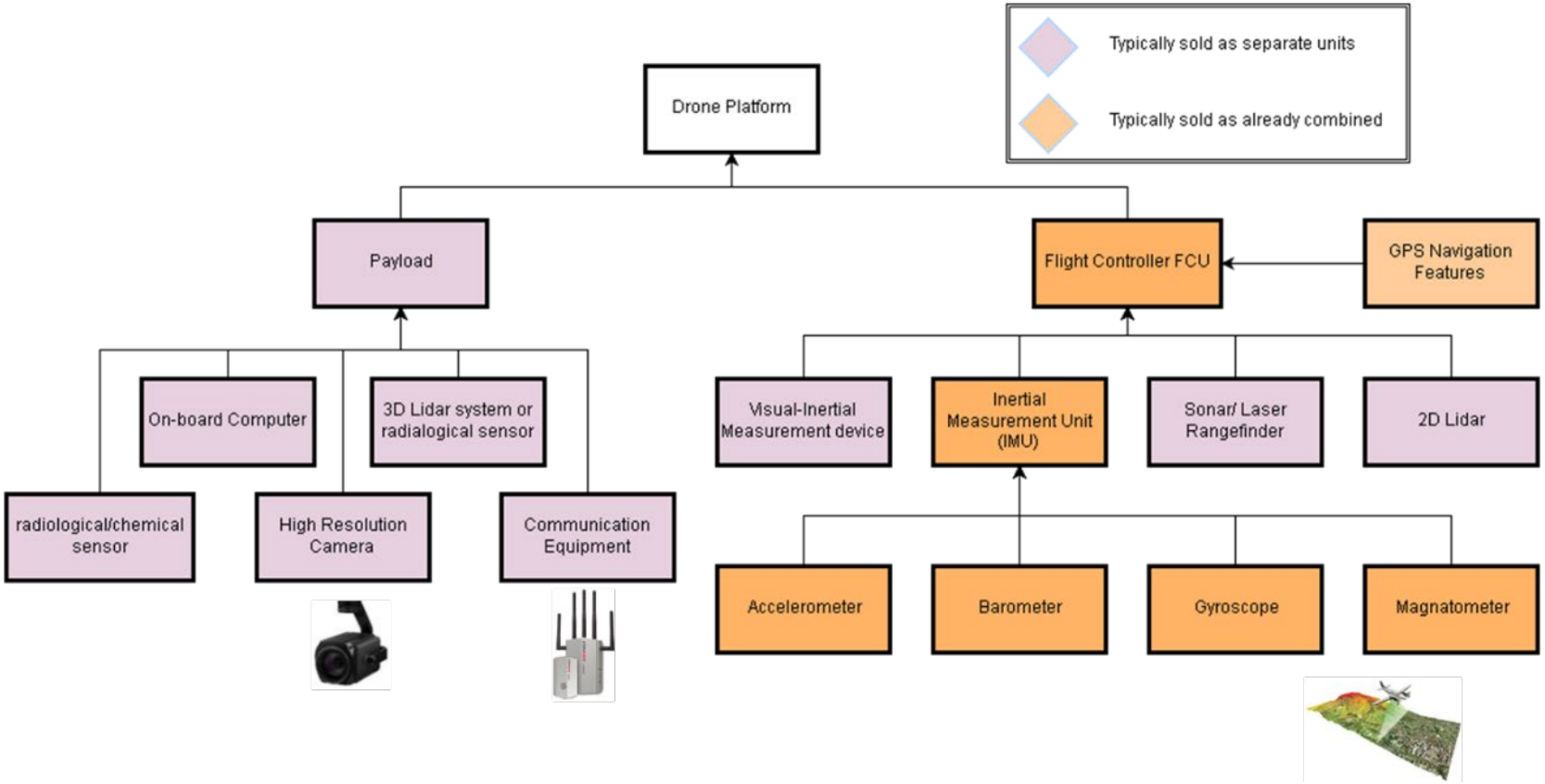


Figure 11: sUAS Sensor platform categories

The main sUAS that were used throughout the project were the Matrice 100 and the Matrice 210 RTK. These sUAS used nearly identical camera systems interchangeably and this was one of the advantages of selecting the DJI Matrice platforms over the Intel Falcon 8+. Some of the cameras used included [21]:

- Matrice 100:
  - Zenmuse X5
    - High-resolution (4K) camera with interchangeable lenses used for mapping and traffic monitoring
  - Zensmuse Z3
    - 3X Optical zoom camera used for mapping and facilities inspection
- Matrice 210
  - Zensmuse X5S
    - High-resolution (4K) camera with interchangeable lenses used for mapping and traffic monitoring
- Both
  - Zenmuse Z30
    - 30X Optical zoom camera used for bridge inspection and traffic monitoring
  - Zenmuse XTR
    - Thermal radiometric camera used for bridge and facilities inspection

The X5 camera is a 16MP camera with a 4/3" CMOS sensor. This camera has interchangeable lenses allowing the operator to select the best focal length for a given operation. This camera was mostly used for our traffic monitoring and mapping missions.

The Z3 is a 3X optical zoom camera. This 12MP camera was used mainly for mapping and facilities inspections. It is capable of 4K video. The focal length of this lens is between 22–77 mm. This camera was mostly used in the beginning of the project prior to the use of the Z30.

The X5S was a similar camera to the X5 but it is compatible with the Matrice 210. This camera had slightly better resolution at 20MP. It was also compatible with similar lenses to the X5. This camera was a primary camera for traffic monitoring, bridge inspections, and mapping.

The Z30 was the powerhouse of all our cameras. Capable of 30X optical zoom the camera proved its use for traffic monitoring and bridge inspections. The Z30 was used primarily on the Matrice 210 but also was equipped to the modified tethered Matrice 100. While the Z30 had the largest zoom it was only a 2MP camera so it was never used for mapping missions.

Finally, the last camera used often during this project was the XTR. This camera was a radiometric thermal camera that was used on both the Matrice 100 and the Matrice 210. The XTR is capable of 640x512 resolution at 30Hz. The camera is built off the FLIR Tau 640 camera. The radiometric camera allows for getting thermal data on every pixel of the camera. This camera was used heavily during bridge inspections as it was able to see delamination, for facilities inspection to see heat loss in buildings, and initial traffic monitoring to help identify vehicles.

3.1.5 Software: Flight control and planning and image processing

The use of ground control station (GCS) applications was crucial for the majority of operations during the project. These applications ran on the tablets connected to the radio controllers the pilot or camera operator was using. As with the flight hardware, this area has also undergone radical evolution and advancements since 2016. Many applications were looked into, but using the DJI GCS applications were the clear frontrunners based on the fact that the team chose to adopt many of

their systems and payloads. The three applications used most were DJI Go, DJI GS Pro, and DJI Pilot.

The DJI Go app was used on most general flights using the Matrice 100. It allowed the pilot and the camera operator to quickly get the vehicle in the air and allowed for switching between collecting images or video and zoom if the camera allowed. This would be the most common used app for traffic monitoring and bridge inspection while using the Matrice 100.

The DJI GS Pro app was similar to the Go app, but had the advantage to create waypoint flights for preprogrammed flight planning. The app also gave the use access to numerous vehicle and payload settings including altitude, camera angle, Ground Sampling Distance (GSD), overlap, etc. We will cover more on this application in the sections below.

The DJI Pilot app was used exclusively with the Matrice 210. This application allowed similar features of the Go app but included more features specifically for the Matrice 210.

In addition to software for flight control and planning, the team looked at several software packages for image processing. These packages typically allow the user who has flown a mission and collected a set of images to process and stitch these in order to develop composite 2D/orthomosaics and 3D/mesh/point cloud models. This area, termed photogrammetry, is currently also an open research topic although some areas are reasonably well developed. Successful application of such photogrammetry techniques requires precise control of both the camera and vehicle as well as the ability to obtain GPS signals so that images can be taken in a prescribed and regular manner across the entire area to be mapped. Some of the details of this process will be covered in the sections below on construction site monitoring and bridge/facility inspection. See also the documents [5,24,25,26,27]. In this section we wish to point out that like other aspects of UAV operations, such third party image processing packages have also undergone a significant evolution since 2016. At the start of this project Open Drone Maps and Pix4D were the primary market leaders in aerial mapping software. Subsequently, Context Capture came to market paired with the Intel Falcon 8+ by Topcon. Eventually, Context Capture was acquired by Bentley and rolled into their CAD software suite. All three packages were compared and contrasted. The Ohio UAS Center had already become familiar with Pix4D. Open Drone Maps had the advantage of being both open source and having been created in Ohio. In the end, Pix4D became the frontrunner and was adopted for use in this project. Further details of its use and capabilities will be outlined in Sections 3.3-3.5.

3.1.6 General UAS Flight Operations Considerations

As discussed above, the possible range of applications area for UAS is extremely large and growing on a daily basis. In order to identify application areas that might provide the best return on investment for this project, the research team, working with Ohio UAS Center personnel.

Many of the flights that were conducted during this research project required considerations to collect the data in very particular and specific manners. For example, traffic monitoring required an extensive set of tests to determine the best altitude and camera angle to allow for image processing. These considerations required that the operators follow a set standard operation for these flights.

The main guidelines for this project were that of 14 CFR Part 107. These rules from the FAA guided the team on their general flight considerations. The team created an SOP for all sUAS and missions used during the project [5].

The team created flight plans that would be used for the operations which included:

- Personnel
- Flight Operations
- Safety and Emergency Operations
- Flight Area Management
- Communications
- Data Reporting and Logging

The team typically used at minimum 2 person teams for all operations. This usually included a Pilot and a camera operator. This allowed the pilot to just focus on the flight while the camera operator could get the correct data. Additional personnel would be used such as visual observers or someone running the tether/generator.

Flight operations would include how the flight was performed. What app the team was using, what waypoint missions were to be flown and so on.

Safety was always a main consideration during every flight. Each test included what steps would need to be taken to ensure safe operations. Some of these tests would involve testing at bridges or along highways so the team would need to notify ODOT and the local police that we would be operating. This safely also included how accidents and incidents would be reported and who to contact prior to each operation.

Flight area management included checking NOTAMs and TFRs, using airspace waivers near airports, and just having a good understanding of potential obstacles the sUAS could encounter.

Communications was not just the internal communication between the crew members, but also external with local airport authorities, local police, ODOT, and locals who may be interested in out operations.

Finally, the data reporting and logging of data was important to verify the correct data was collected and shared to the appropriate person. For many of the mapping and bridge inspection missions hundreds of images would be taken and need to be sorted so they could be processing in a timely manner.

3.2 Traffic Monitoring

The use of UAVs as airborne platforms for traffic monitoring was identified early on in this project. They offer the advantage of providing an eye-in-the-sky that might augment, reduce or even eliminate roadway sensors which have maintenance and reliability issues. In addition, they increase safety for both ODOT employees and the traveling public in that they do not have to be installed requiring lane closure, etc. They can also be rapidly deployed on an as needed basis to a wide variety of locations. Finally, the range of cameras/lenses available coupled with the range of altitudes and the possible use of tethered vehicles allows for video and site pictures which can range from close in (e.g., lane-wise, etc.) to panoramic (e.g., taking and entire intersection, interchange, or stretch of roadway, etc.) and for data sets collected over extended periods of time (e.g., several hours).

UAVs do have some limitations, however. These include primarily: (1) flying at night requires FAA waivers; (2) weather conditions must be advantageous (e.g., there are operational limitations on wind, precipitation, etc.); (3) the video obtained must be post processed either manually by having humans watch and interpret (e.g., obtain traffic parameters such as counts, flow, headways, etc.) or by the application of AI/computer vision software. The latter software is not readily available commercially and AI/computer vision is still an open research topic which the research team did spend considerable time investigating as discussed in the sections below.

The results reported here borrow heavily from the SOPs developed by the research team for use by Ohio UAS Center pilots and personnel as part of this project [5,6,7] as well as the theses generated by the various graduate students involved in this research project [8,9]. The reader is referred to these documents for additional information and details.

3.2.1 General Considerations, Flight Planning and Vehicle/Sensor Package Considerations

As outlined in the SOPs [5,6,7] and discussed above in Section 3.1, vehicles such as the DJI Matrice 100, the Matrice 100 modified with the Powerline Tether System, and/or the Matrice 210 RTK are primary UAV platforms used for traffic monitoring. These platforms should be coupled with video cameras which are capable of providing a wide range of functionality such as optical zoom, high definition video recording capability based on the flight conditions and mission objectives. These include such cameras as the Zenmuse Z3, X5, X5s, and/or Z30.

When planning flights for traffic monitoring, it was helpful to conduct a pre-planning exercise. As an example, consider the Google Earth image given in Figure 12 which shows a location along I-75 in Cincinnati near the Mitchell Ave exit where some of our testing was conducted. In the particular case, obtaining video of the traffic flow near the ramp and ramp meter and along the main line was of interest. Several possible take-off/landing sites were selected based on access and Part 107

regulations. For example, sites A, C, and D have public access and good line of sight whereas site B is on private property and requires additional permissions. Sites A and C are relatively far from the ramp meter. Note that additional considerations were made of power lines (marked in red) and other potential obstructions. In the end, site D was selected for takeoff and landing and the UAV and camera were adjusted in flight in order to obtain a good site picture. The use of a tethered vehicle introduces still other considerations when operating near power lines or other obstructions. See SOPs [5,6,7] for further details on the aspects of flight planning considerations.

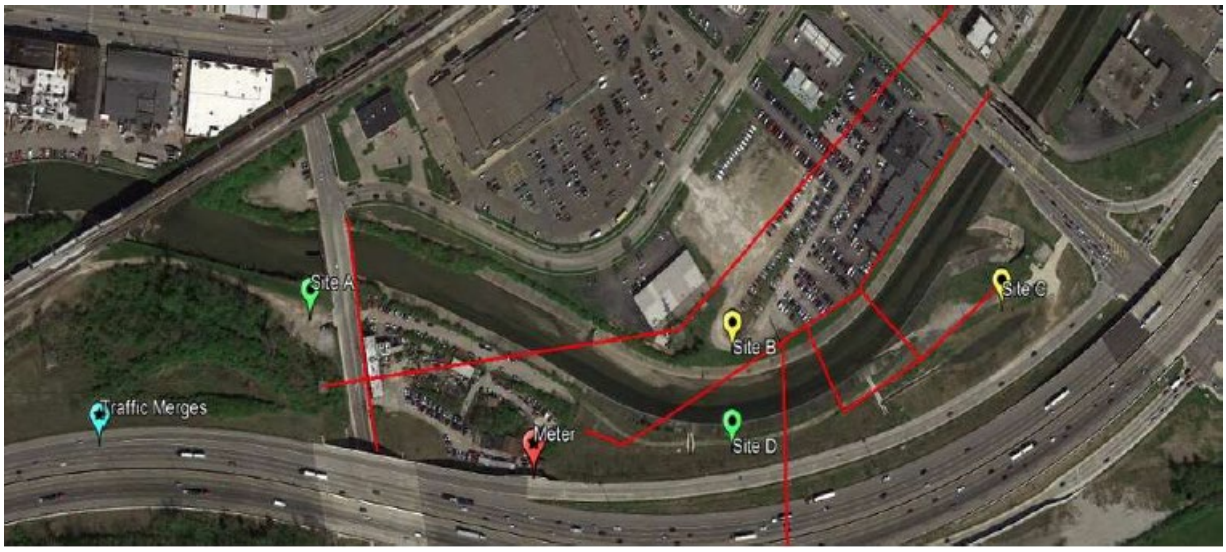


Figure 12: Examples of flight planning considerations

Figure 13: Examples of site pictures showing qualitative traffic features

Figure 13 provides examples of site pictures resulting from flights at site D. Such images and/or video may prove useful in order to obtain a qualitative understanding of traffic flow and patterns. They give a good overall/global idea of traffic characteristics as well as any issues which may be present. However, through our various activities, we found these types of images/video less useful in obtaining quantitative traffic parameters/metrics such a count, speed, flows, densities, headways, etc. For this latter purpose, it was found that site pictures such as those

shown in Figure 14 were easier to process and interpret, either manually by humans or by using AI/computer vision-based methods.

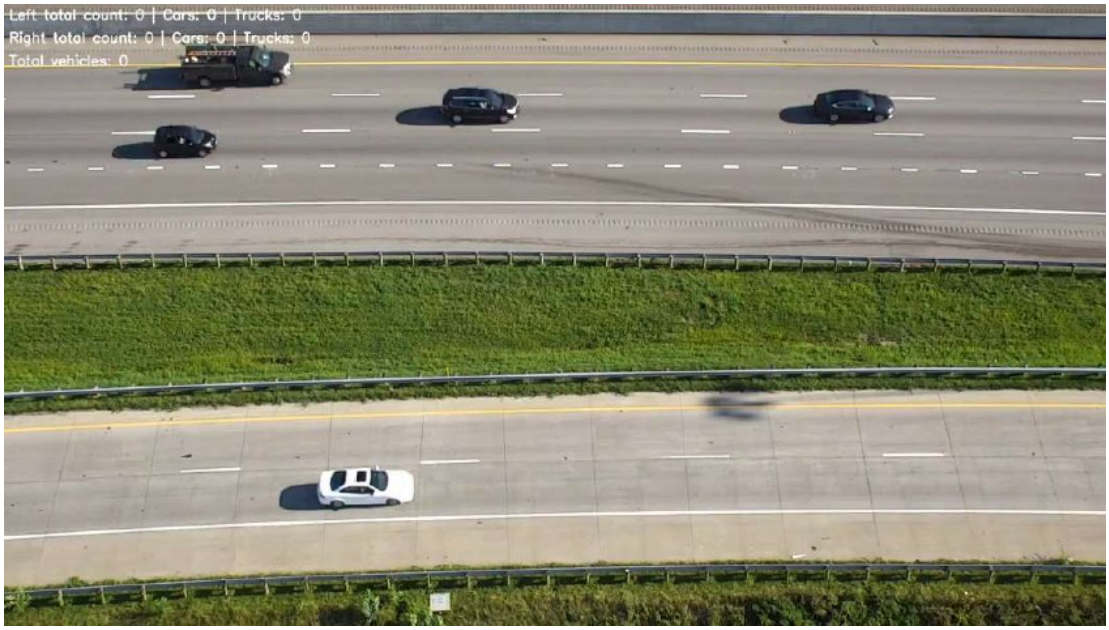


Figure 14: Examples of site picture better for quantitative traffic features

More specifically, a general UAV deployment schematic for traffic monitoring is shown in Figure 15. The UAV is flown next to the roadway to capture the view of the traffic flow. Once airborne, an untethered UAV's flight time generally varies between 15 to 20 minutes, but the tethered system can fly for extended periods of time.

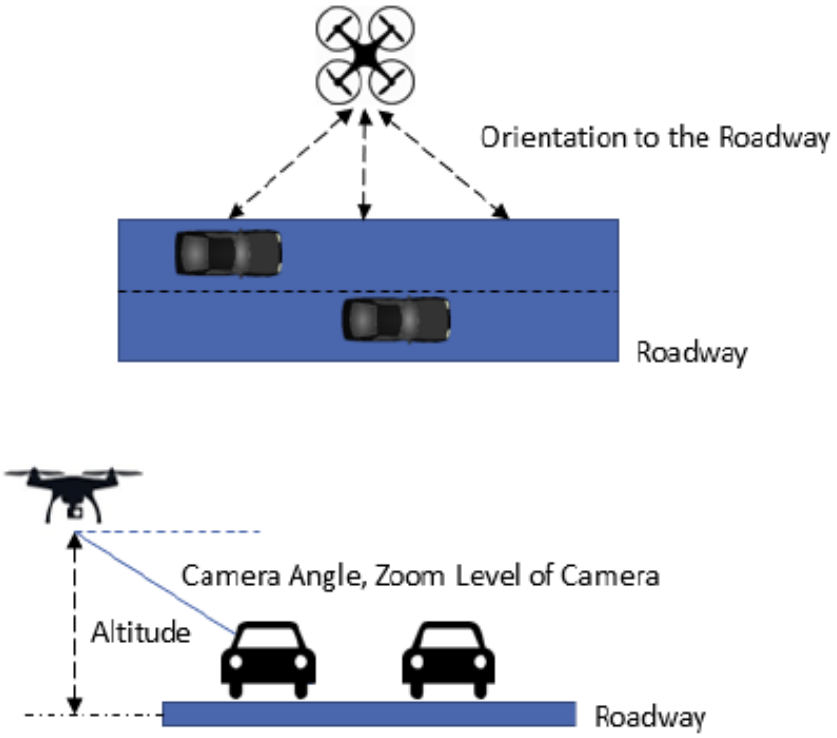


Figure 15: Typical UAV deployment for traffic monitoring

Several test flights were flown at various locations in order to further address the particulars of flight and camera parameters. Consider Figure 16 which summarizes the results of various tests of site pictures obtained at various angles relative to the roadway. As it shows, extracting estimates of distance or speed can become problematic using site pictures at an angle skewed to the roadway due to distortions caused by the depth of field. Further, at certain angles vehicles in one lane can block the views of vehicles in adjacent lanes complicating the process of running object detection routines, obtaining counts, etc. Additional tests were run in order to understand the impact of altitude, distance from the roadway, camera zoom, etc. [8]. Traffic was observed manually on site using a JAMAR [10]. In addition, UAV traffic video was processed manually and via various AI/Computer vision methods. Results were compared. See [6,8] and the data tables contained in Figures 17-18. Figure 17 shows the results of several experiments where the angle of the camera relative to the flow of traffic (see Figure 16) was examined. Figure 18 shows the results of several experiments where UAV altitude and distance from the road were examined. In the end, it was determined the best efficiency of processing was obtained with camera angles with respect to the edge of a road should be between +/-45 deg to 50 deg. and a distance from the road-to-altitude ratio in the range 1:1.15. Thus, if the UAV is flying at an altitude of 200 feet above ground level then the horizontal distance of the UAV from roadway should be around 175 feet. The zoom level of the camera could then be adjusted by the user to fine tune the site picture. Overall, it was found that such flights yielded video/images with relatively small spatial

distortion, vehicles in adjacent lanes did not block view of one another, and computer recognition algorithms such as YOLO [11,12] could easily be trained in order to obtain reasonable detection and tracking accuracies.

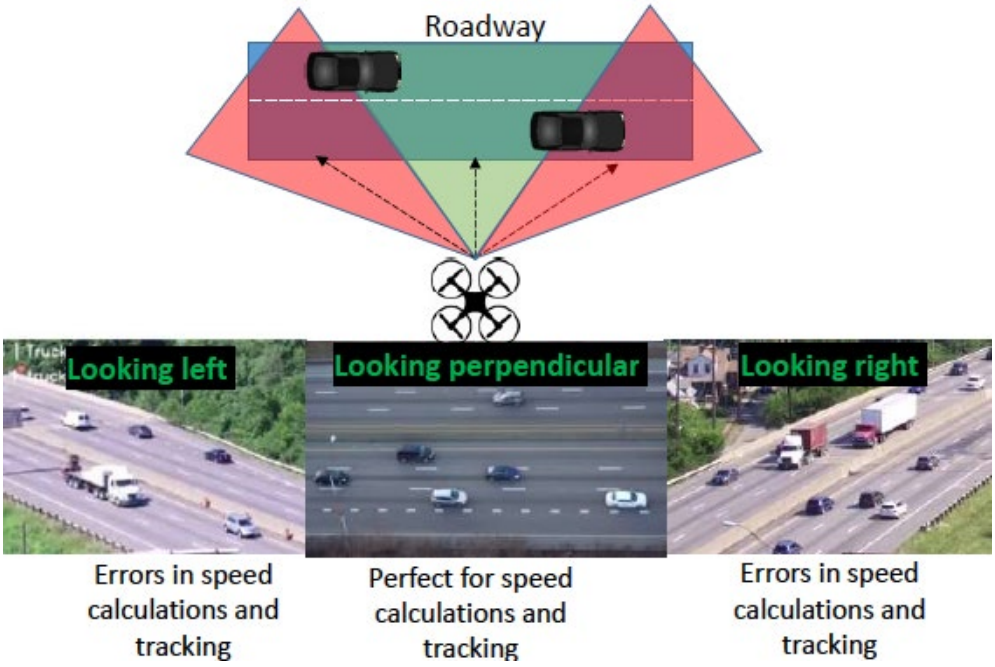
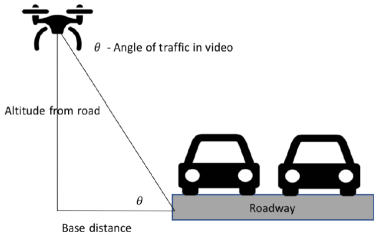


Figure 16: Details on UAV deployment for traffic monitoring

Zoom Level (mm)	Altitude (ft)	Camera angle	Distance from Road (ft)	Obstruction	Duration	Human Count		Machine Count		Error Rate(%)		Missed Detections		False alarms	
						Left	Right	Left	Right	Left	Right	Left	Right	Left	Right
22	-	90	175	Trees	01:10	96		18		-81.3		-	-	-	-
22	150	45	175	Trees	00:09	16	12	28	12	75.0	0.0	7	6	2	1
22	150	45	175	Trees and road signs	00:05	16	10	21	8	31.3	-20.0	1	4	1	0
22	150	45	175	Trees, road signs, bridge, banner	00:12	14	19	11	16	-21.4	-15.8	7	8	2	3
40	150	45	175	Trees, road signs, bridge, banner	00:17	26	21	26	26	0.0	23.8	8	7	2	5
40	150	45	175	Trees, road signs, banner	00:22	30	28	25	25	-16.7	-10.7	9	10	3	5
40	150	45	175	Trees, banner and road signs	00:42	44	49	55	60	25.0	22.4	20	11	10	7
22	150	90	175	Trees	00:10	18	9	26	13	44.4	44.4	1	3	2	2
30	150	90	175	Trees	00:12	11	18	15	21	36.4	16.7	4	5	2	5
22	150	90	175	Trees	00:12	14	11	21	16	50.0	45.5	2	3	5	10
40	150	45	175	Trees, road signs, bridge, banner	00:45	56	53	44	40	-21.4	-24.5	35	22	7	3
22	150	45	175	Trees, road signs, bridge, banner	00:21	38	39	31	18	-18.4	-53.8	23	22	12	13
40	150	45	175	Trees	08:11	608	510	35	53	-94.2	-89.6	Large	Large	Large	Large
50	150	45	175	Trees and road signs	10:01	890	591	52	65	-94.2	-89.0	Large	Large	Large	Large
22	150	45	175	Trees and bridge	01:09	50	70	45	21	-10.0	-70.0	27	42	Large	Large
40	150	45	175	Trees	04:17	361	272	108	77	-70.1	-71.7	Large	Large	Large	Large
29	150	45	370	None	01:25	29	32	44	46	51.7	43.8	22	3	15	16
14	150	45	175	Trees and bridge	01:01	61	71	15	23	-75.4	-67.6	Large	Large	Large	Large
29	150	90	370	None	02:03	34	44	39	54	14.7	22.7	0	0	5	10
25	250	90	170	None	03:26	306	263	304	264	-0.7	0.4	17	11	11	18
20	250	90	170	None	03:19	304	234	278	228	-8.6	-2.6	27	26	9	15
14	250	90	170	Trees and vehicles in background	03:11	270	214	258	235	-4.4	9.8	32	21	8	27
25	150	90	170	Trees	03:32	282	220	202	260	-28.4	18.2	104	26	5	39
20	150	90	170	Trees	03:13	250	235	207	232	-17.2	-1.3	58	42	6	26
14	150	90	170	Trees	03:07	261	197	154	212	-41.0	7.6	124	40	18	57

Figure 17: Traffic Monitoring flight experiments summary [6,8]



Flight Parameters				Custom Analysis Parameters				Statistical Analysis Parameters				
Site	Base (feet)	Altitude (feet)	Altitude above road(feet)	Overlap in vehicles in adjacent lanes	Missed Detection?	Angle - θ (degrees)	Is it a good zoom?	Human Count	Machine Count	Error(%)	Missed Detection	False Alarms
A	225	200	160	Yes	No	35.4	No. Zoom out needed, to cover 3 trucks in frame	X	X	X	X	X
A	225	240	200	Partial	No	41.6	No. Zoom out needed, to cover 3 trucks in frame	X	X	X	X	X
A	225	290	250	No	No	48	No. Zoom out needed, to cover 3 trucks in frame	192	185	-3.65	10.00	3.00
A	285	200	160	Yes	No	29.3	No. Zoom out needed, to cover 3 trucks in frame	X	X	X	X	X
A	285	240	200	Yes	No	35	No. Zoom out needed, to cover 3 trucks in frame	X	X	X	X	X
A	285	290	250	Partial	No	41.2	No. Zoom out needed, to cover 3 trucks in frame	X	X	X	X	X
A	370	200	160	Yes	No	23.3	No. Zoom out needed, to cover 3 trucks in frame	X	X	X	X	X
A	370	240	200	Yes	No	28.3	No. Zoom out needed, to cover 3 trucks in frame	X	X	X	X	X
A	370	290	250	Partial	No	34	No. Zoom out needed, to cover 3 trucks in frame	X	X	X	X	X
D	100	200	170	No	Yes - on ramp	59.5	Good	X	X	X	X	X
D	100	230	200	No	Yes - on ramp	63.4	Good	X	X	X	X	X
D	100	280	250	No	Yes - on ramp	68.1	Good	X	X	X	X	X
D	175	200	170	Partial	No	44.1	Good	X	X	X	X	X
D	175	230	200	No	No	48.8	Good	187	183	-2.14	8.00	6.00
D	175	280	250	No	Yes - on ramp	55	Good	X	X	X	X	X

Figure 18: Traffic Monitoring flight experiments summary [6,8]

Ultimately, while manual processing of UAV traffic video by humans was shown to be possible, it was found to be highly laborious and time consuming. Instead, it was determined that combining UAV video with AI/computer vision software for post processing was a better option and this avenue was explored as part of this project [6,7,8,9]. The strategy shown in Figure 19 was employed where software was used to implement object detection, classification and tracking and subsequent parameter extraction. There are no commercially available off the shelf traffic video processing packages. Further, the area of AI and computer vision is still largely an open research topic. As a result, full exploration of this area would be a research project unto itself. The research team was, however, able to test a number of deep learning and neural network concepts using vehicle and transportation training sets available online, augmented with training data sets generated using our own UAV video footage coupled with existing object detection and tracking algorithms. Technical details such as software packages, hardware considerations, choice of computer vision algorithms, etc. can be found in [8,9,13].

We were successful in training neural networks which could distinguish cars and trucks. Specifically, within the FHWA's hierarchy of vehicles (Figure 20, [14,15]), we were able to classify two groups: (1) passenger cars, class 2; and (2) trucks, classes 3 and 5-13. Figure 21 gives some examples of the types of situations that lead to object detection and tracking errors using AI/computer vision software. These types of error notwithstanding, as we will see in the sections that follow, results of high quality were obtained. Based on the preliminary results obtained in this area, reported the subsections that follow, the team feels that the proof of concept of UAV-based traffic monitoring was established and that remaining issues could be dealt with effectively to yield dedicated, automated traffic monitoring software that operates at/near real-time speeds and that produces highly accurate results. However, that would be the subject of a dedicated project onto itself.

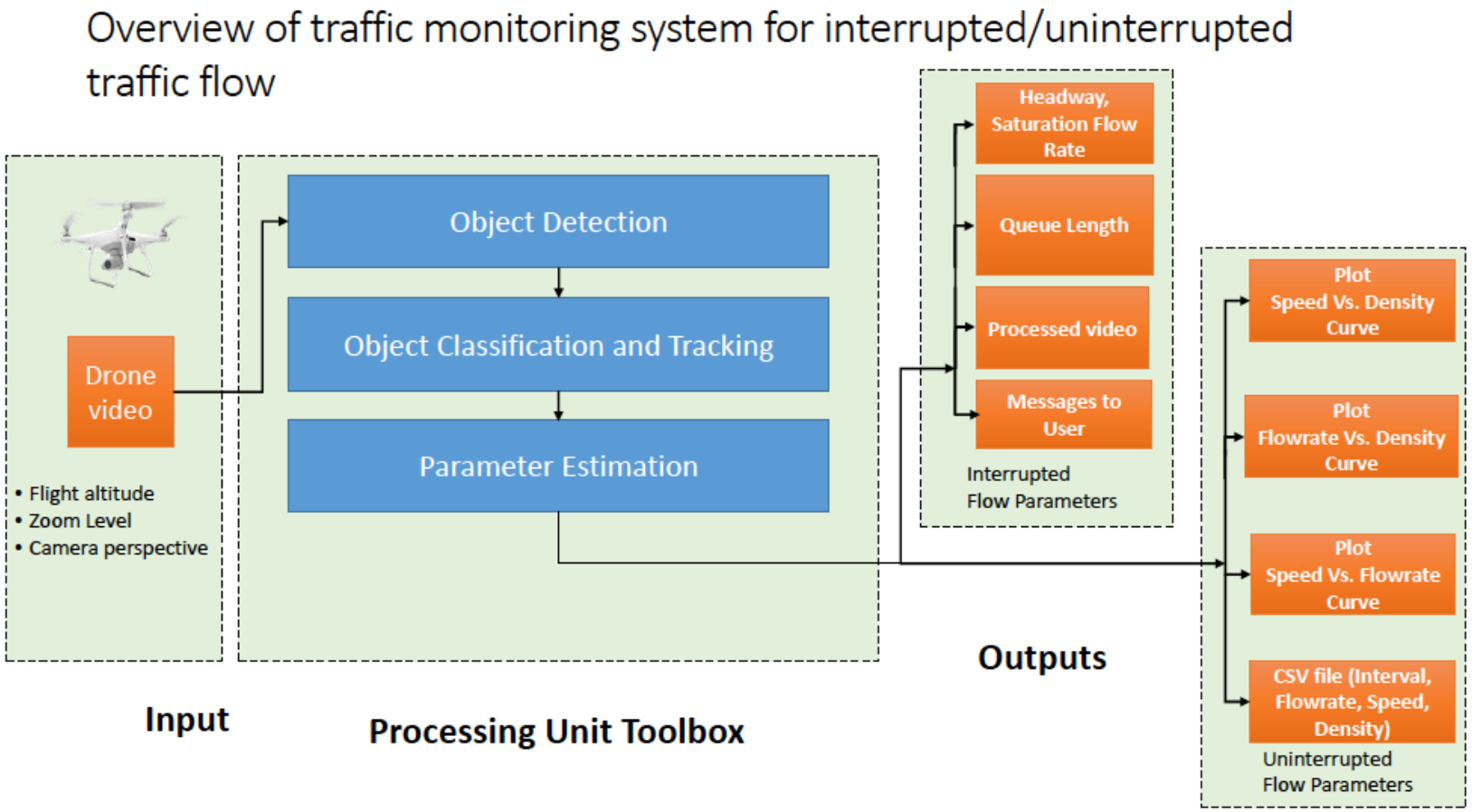


Figure 19: Software based traffic video processing

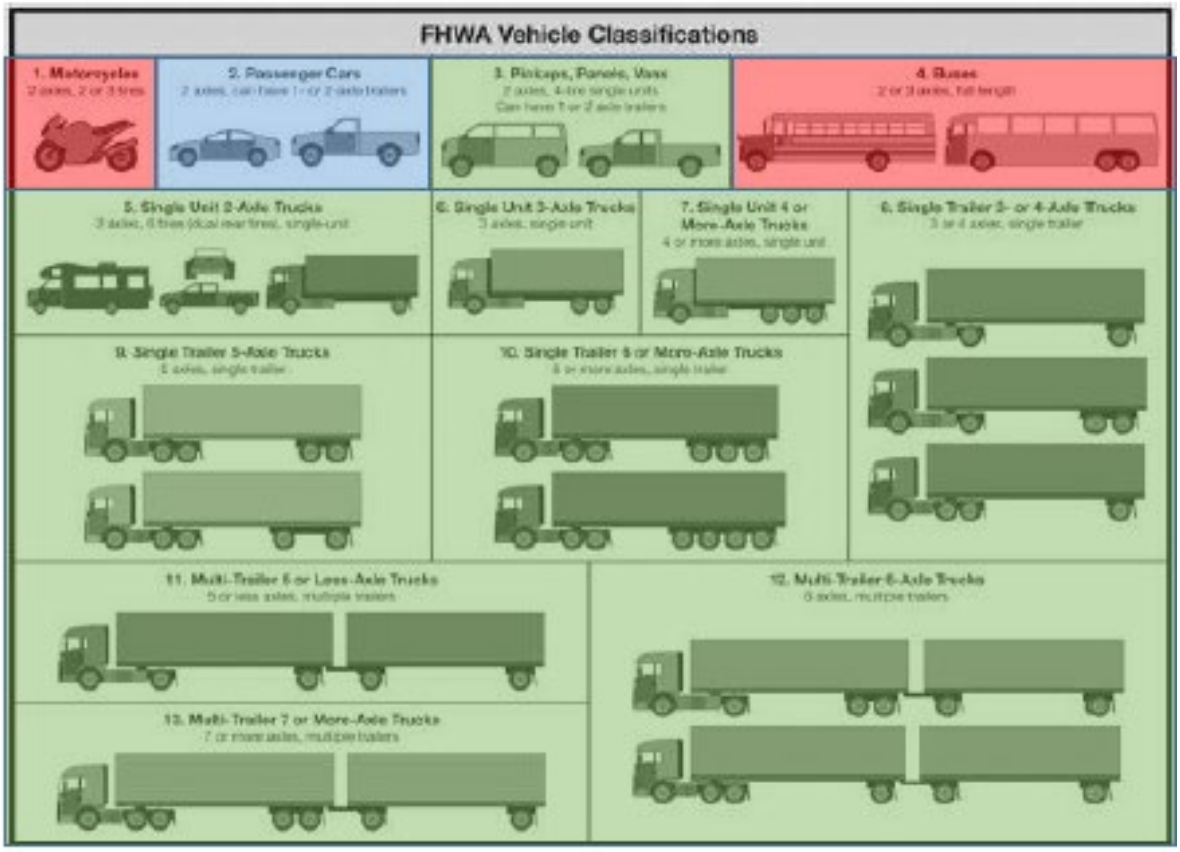


Figure 20: FHWA vehicle classification

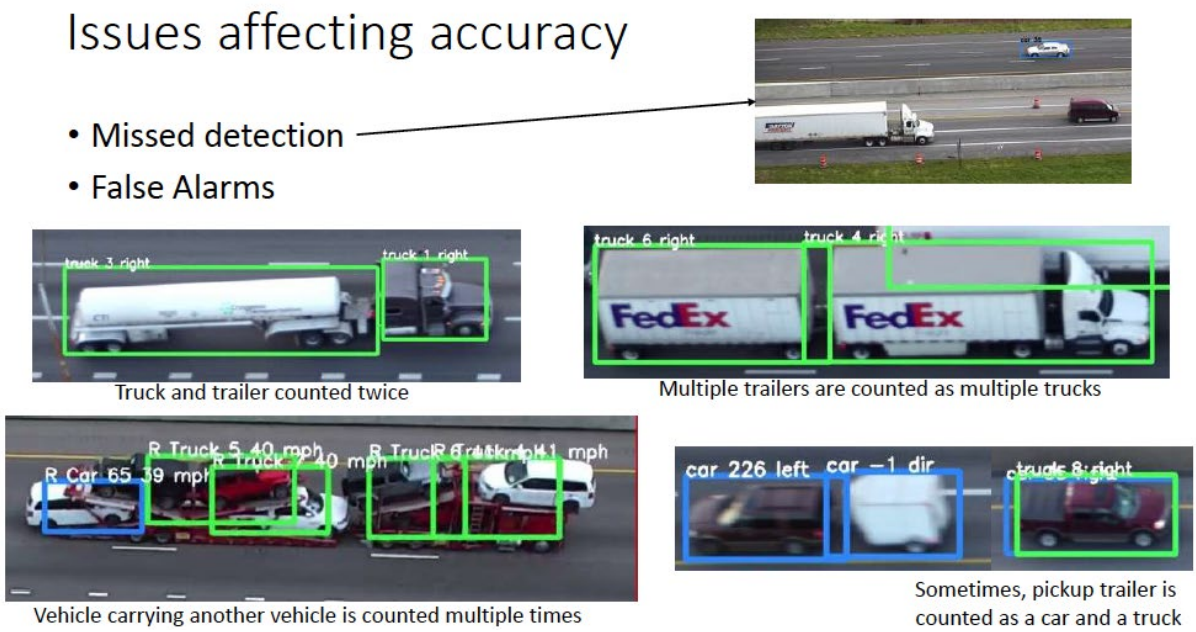


Figure 21: Some issues affecting AI/Computer vision based methods

3.2.2 Uninterrupted Flow

One type of roadway scenario that traffic engineers are interested in monitoring is that of uninterrupted flow [14,15]. That is, roadways, such as freeways and highways, where traffic is free to move continuously. These situations typically include roads where opposing directions are separated by barriers and there are two or more lanes of traffic in each direction. On such roadways efficient operation is characterized by a balance between capacity and demand and can be measured in terms of the variables: flow rate, speed, density, and counts, the first three of which are interrelated as shown in Figure 22.

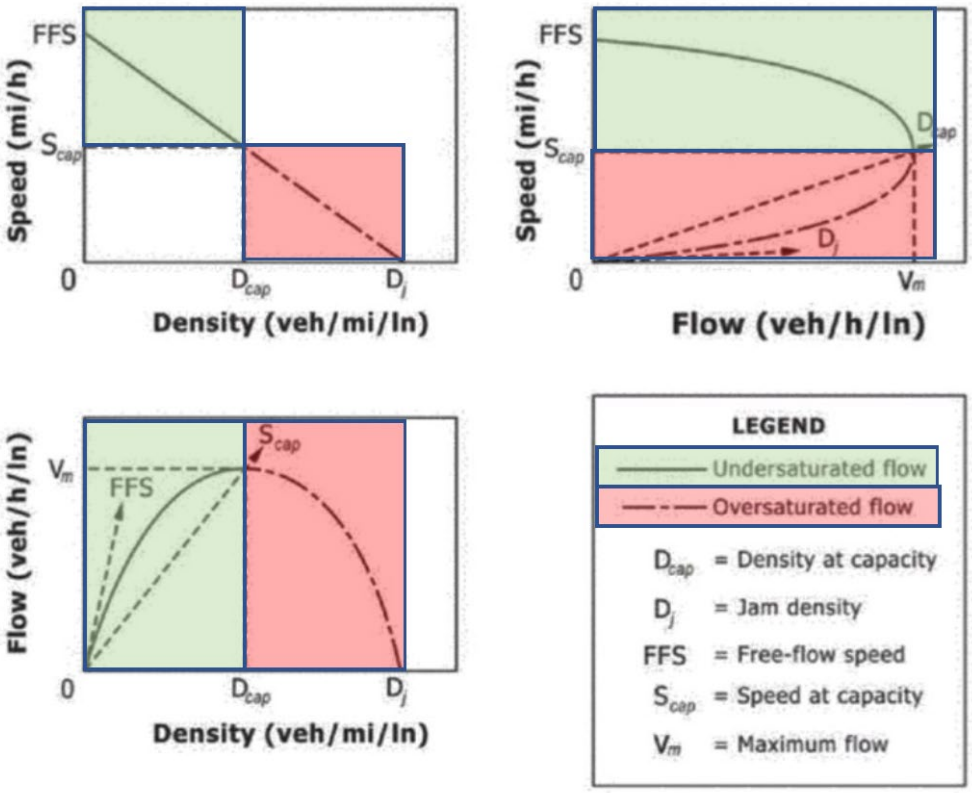


Figure 22: Relationship of flow rate, speed, and density [14,15]

In order to explore the use of UAVs in monitoring such traffic patterns, the research team worked with ODOT traffic personnel both at the Central Office and in Districts 1 and 8. Several sites were identified and approved for flight operations. These included several portions of I-75 from Findley south to Cincinnati, as well as portions of I-275 and the Ronald Reagan Cross County Highway in Cincinnati. These sites were flow as per the flight operations recommendation described in Section 3.2.1 and videos of various durations were recorded. These videos were then

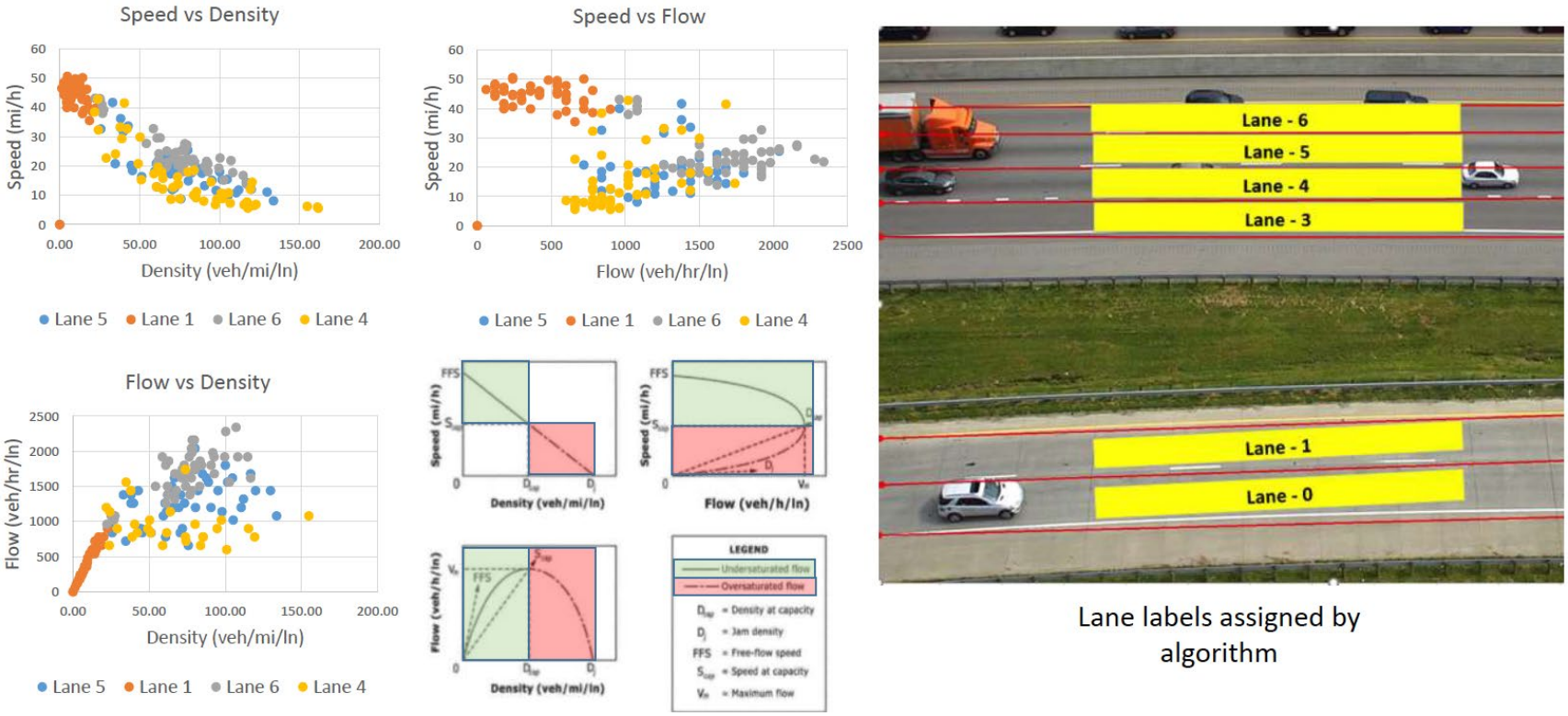
processed manually (by having a human watch the videos and conduct counts) as well as using a series of software scripts written and iteratively improved upon in order to help automate processing as described in Figure 19, see [6,8] for technical details. Manual and automated results were compared and used for further algorithm/software tuning. Figure 23 characterizes the sort of accuracies that were ultimately obtainable. As can be seen, an average precision of 89% was obtained. It is felt that with more research and study, particularly into training of the convolutional neural networks used of object detection and tracking, that greatly enhanced precisions could be achieved.

Based on these results, the software code was augmented to also estimate flow rate, density, volume and speed. The prototype code was developed to take the UAV video files and their respective subtitle files as input and output a csv file with the collected parameters and an output video file as shown in Figure 24. Figure 25 shows output plots similar to those from the FHWA HCM (see Figure 22). These contain estimates of flow rate, speed and density in relation to one another for various lanes of traffic along a section of I-75 in the Cincinnati area. These results, and others like them were found to be very useful be ODOT Traffic Engineering personnel.

Video ID	False Positive (False Alarms)	False Negatives (Missed Detections)	True Positives	Recall	Precision	F-1 Score
1	9	5	73	0.94	0.89	0.91
2	9	4	76	0.95	0.89	0.92
3	7	3	91	0.97	0.93	0.95
4	19	2	76	0.97	0.80	0.88
5	8	2	82	0.98	0.91	0.94
6	8	4	61	0.94	0.88	0.91
7	18	2	75	0.97	0.81	0.88
8	8	3	77	0.96	0.91	0.93
9	11	2	101	0.98	0.90	0.94
10	5	3	93	0.97	0.95	0.96
11	9	7	98	0.93	0.92	0.92
12	19	6	95	0.94	0.83	0.88
13	17	5	87	0.95	0.84	0.89
14	17	6	99	0.94	0.85	0.90
15	14	9	73	0.89	0.84	0.86
16	12	11	88	0.89	0.88	0.88
17	14	7	76	0.92	0.84	0.88
18	12	9	86	0.91	0.88	0.89
19	9	7	78	0.92	0.90	0.91
20	4	11	80	0.88	0.95	0.91
21	9	6	93	0.94	0.91	0.93
22	8	10	89	0.90	0.92	0.91
23	6	2	88	0.98	0.94	0.96
24	8	7	87	0.93	0.92	0.92
25	10	7	69	0.91	0.87	0.89
26	10	3	85	0.97	0.89	0.93
27	4	6	88	0.94	0.96	0.95
28	12	2	76	0.97	0.86	0.92
29	10	10	101	0.91	0.91	0.91
30	15	4	105	0.96	0.88	0.92
31	12	13	81	0.86	0.87	0.87
32	13	3	103	0.97	0.89	0.93
33	19	13	93	0.88	0.83	0.85
34	16	4	78	0.95	0.83	0.89
35	6	7	73	0.91	0.92	0.92
36	6	5	82	0.94	0.93	0.94
37	5	6	89	0.94	0.95	0.94
38	10	5	88	0.95	0.90	0.92
39	10	3	81	0.96	0.89	0.93
40	12	7	67	0.91	0.85	0.88
41	7	14	86	0.86	0.92	0.89
42	12	7	75	0.91	0.86	0.89
43	6	8	92	0.92	0.94	0.93
44	5	9	72	0.89	0.94	0.91
45	9	13	72	0.85	0.89	0.87
47	13	4	88	0.96	0.87	0.91
48	7	1	73	0.99	0.91	0.95
49	6	3	88	0.97	0.94	0.95
50	11	6	82	0.93	0.88	0.91
AGG value	506.0	296.0	4109.0	0.93	0.89	0.91

Figure 23: Characterization of traffic count accuracy [8]

Figure 24: Typical view of AI/Computer vision video output [8]



Lane labels assigned by algorithm

Figure 25: Characterization uninterrupted traffic flow parameters [8]

3.2.3 Interrupted Flow, Signalized Intersections

Another type of roadway scenario that traffic engineers are interested in monitoring is that of interrupted flow [14,15]. That is, roadways, such as city streets, where traffic is regulated by external means such as the use of traffic lights, stop signs, etc. On such roadways additional variables such as saturation headway, saturation flowrate, queue size, unmet demand, arrival demand, peak hour factor, etc. are of interest to traffic engineers in assessing operational characteristics.

As above in Section 3.2.2, the use of UAVs in monitoring such traffic patterns was also explored. Here again, the research team worked with ODOT traffic personnel both at the Central Office and in Districts 7 and 8. Several sites were identified and approved for flight operations. These included several interchanges, e.g., ones along I-675, and I-75 in the Dayton/Miamisburg area, US50 in the Fairfax/Mariemont area, and the Clifton Rd area near the UC campus as well as others. These sites were flow as per the flight operations recommendation described in Section 3.2.1 and videos of various durations were recorded. In some cases, in order to get a global view of the entire intersection, flight parameters had to be adjusted. Also, because of the multi-directional nature of traffic flow at intersections, the site picture could not always be at/near right angles to traffic flow.

The use of AI and computer vision software tools was also investigated for interrupted traffic flow such as signalized intersections. Prototype code was developed to process the UAV video building on the resulted obtained above for the uninterrupted case. Due to the nature and wide variety of interchanges, the fact that they contain stop and go traffic patterns, the presence of parked cars, etc.; the processing of such videos is significantly more complicated than for the uninterrupted flow case. These complexities also effected the object detection and tracking portions of the code. Correspondingly, the use of AI/Computer vision algorithms in this case typically required more user interactions, for example, in order to select a region-of-interest zone so that parked cars and other traffic could be excluded from normal lane traffic and so that counts and other parameters associated with those vehicles not part of the normal traffic would not corrupt/confuse processing, etc. See examples in Figure 26. Also, for some parameters such saturation headway and queuing, it was necessary to process the video in segments timed according to changes in traffic lights, etc.

Figure 26: Examples of additional user inputs needed for computer processing in interrupted traffic case [7]

See, for example, the UAV view given in Figure 27 which show traffic at an I-75 exit ramp in the Miamisburg area. Note that while the rules laid out in Section 3.2.1 could not be followed in detail, the altitude, camera angle, and zoom settings were selected in order that vehicles in various lanes and directions did not overlap or obscure one another. Note also that the presence of street signage, traffic lights, etc. over the roadway made it difficult to obtain completely unobscured views of the traffic. Note also that the increased UAV altitude yielded a site picture where the vehicles were viewed more from their rooves than their sides. This caused some difficulties for object detection as most vehicle detection neural nets are based on training sets which show the side, rather than the top, of the vehicle. Note also the

blue lines which are user defined entries necessary in order to be better able to track vehicles as they cross and enter/exit the intersection from the various directions. Refinement of such issues are examples of the types of things that would need to be addressed in further follow-on research in order to bring these methodologies along beyond the proof-of-concept stage and into mature, practical technologies.

Here, a subset of the videos were first processed manually by a human watching the videos and using a JAMAR device as would be the case for a normal field study [10]. These datasets created tuning sets which were used to develop and refine the software codes and adjust algorithm and processing parameters [7]. Ultimately, several software code routines were developed that were able to handle various interrupted traffic flow conditions fairly successfully. The discussion below gives several examples of the cases considered.

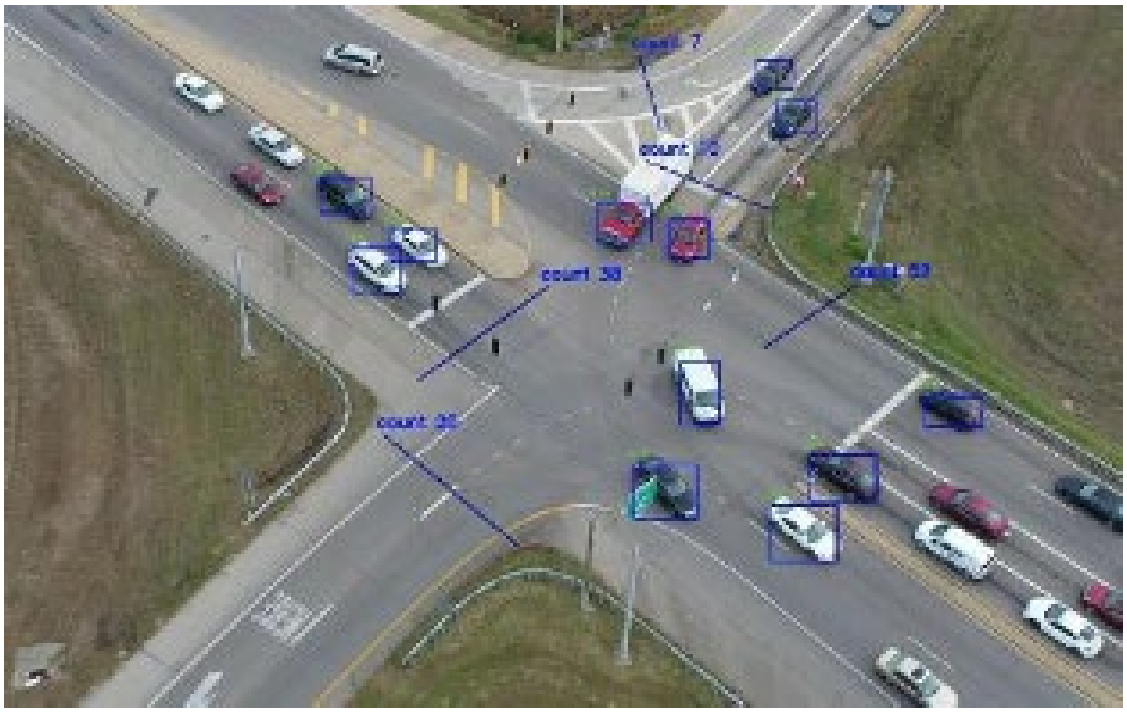


Figure 27: Characterization interrupted traffic flow parameters

Saturation Headway: For a signalized intersection with a queue waiting for the light to change, saturation headway is the difference in time between the passage of two consecutive vehicles through the intersection at steady sate flow. It is calculated, as shown in Figure 28 and is related directly to both the saturation flow rate and the intersection capacity [14,15]. As such, it is a variable of interest for traffic engineers.

A number experiments were conducted as part of this project in order to explore using UAV video in order to estimate saturation headway. Figure 29 shows one such intersection located at the corner of Martin Luther King Blvd and Clifton Ave near the UC campus. This intersection posed a good location for study as it was a busy

intersection, located near the lab, and had a park on one corner which facilitated flight operations of a wide variety.

A number of videos were recorded at this location. These included marking/recording of signal light changes in order to facilitate saturation headway calculations. Benchmark headway calculations were conducted manually in the lab by viewing the traffic videos while using a JAMAR device and its associated software which were able to output headways in response to operator clicks.

In addition, following the approach in Figure 19, various software codes and algorithms were developed to automate estimation of headways based on the UAV traffic videos by detecting and tracking moving vehicles in the field of view and capturing/tagging the video timestamps when the vehicles passed through the intersection. It was found that in order to obtain reasonable results two pre-processing steps were required: (1) turning lanes had to be masked out of the video by placing boxes of them so that artifacts such as vehicles in adjacent lanes and parked cars did not confound object detection and tracking routines, and (2) a line had to be placed in frame at the entrance of the intersection to register vehicle crossings as discussed above in Figure 26. The need to augment the video frame with these markings complicated flight operations in the sense that any drift in the UAV while on station above the intersection caused the markings to move in the field of view resulting the further detection and tracking errors. It is felt that further research leading to more advanced video processing codes could alleviate these issues. These issues notwithstanding, example results for the prototype codes developed are provided in Figure 30 along the JAMAR benchmarks for comparison. As can be seen, errors compared to JAMAR of less than 3% were obtainable, thus establishing a proof-of-concept.

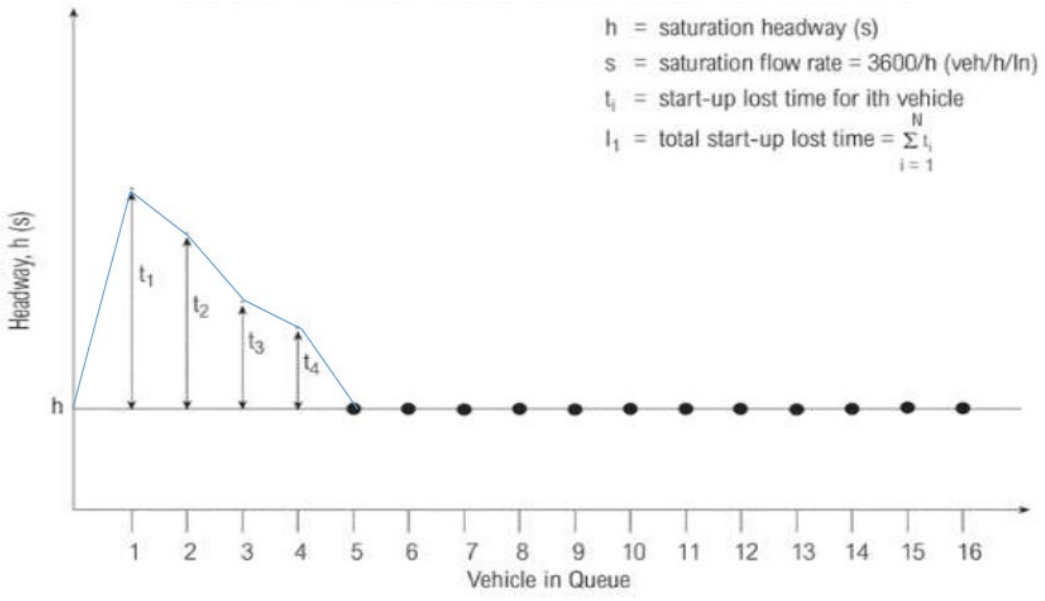


Figure 28: Diagram depicting theoretical saturation headway calculation [14,15]

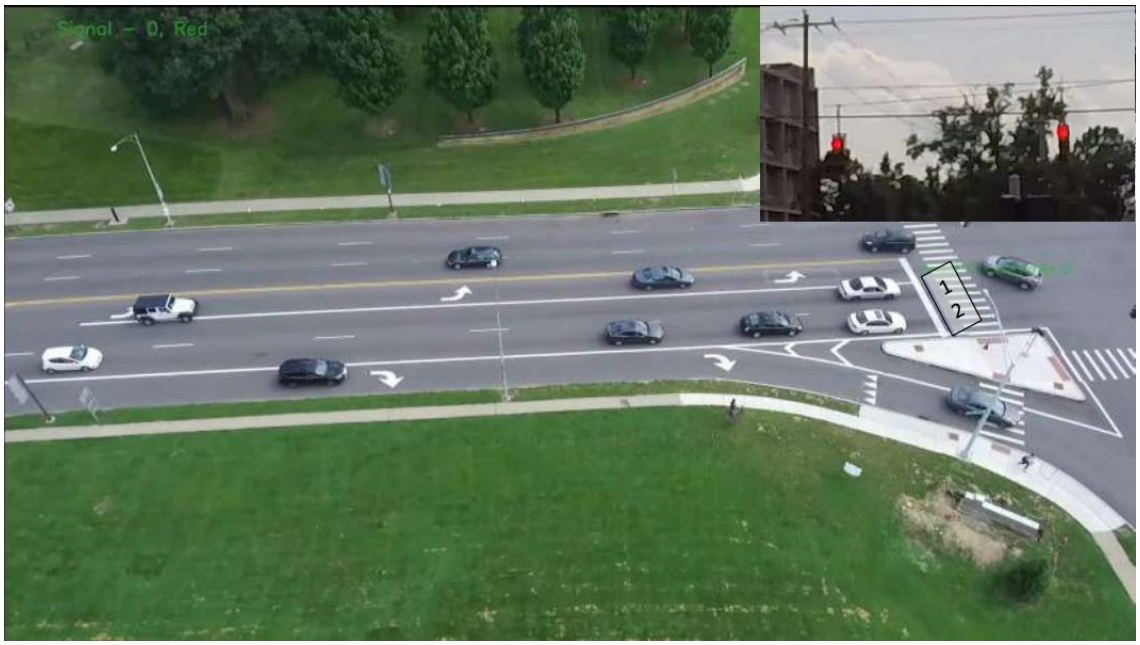
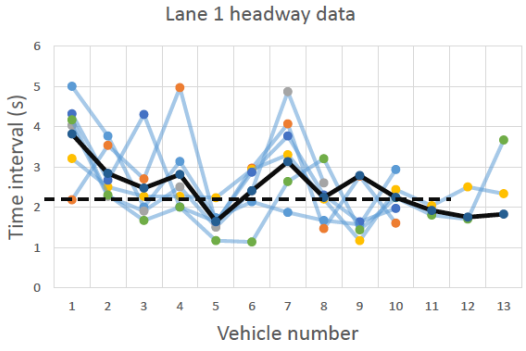


Figure 29: Example of UAV monitoring of signalized intersection

Test case	Angle from the road θ	Distance from road (ft) d	Altitude (ft) a	Number of green signals	Headway		Jamar Headway		Saturation Flow rate		Jamar Saturation Flow rate		Percent Error in Saturation Flow Rate	
					Lane 1	Lane 2	Lane 1	Lane 2	Lane 1	Lane 2	Lane 1	Lane 2	Lane 1	Lane 2
1	45	150	150	6	2.27	2.24	2.28	2.24	1581.5	1604.20	1575	1603.5	-0.41	-0.043
2	50	150	178	5	2.37	2.25	2.16	2.25	1513.2	1595.88	1661.7	1598.7	0.17	0.17
3	50	150	178	7	2.31	2.42	2.38	2.45	1557	1486.80	1508.4	1465.9	8.93	-1.47



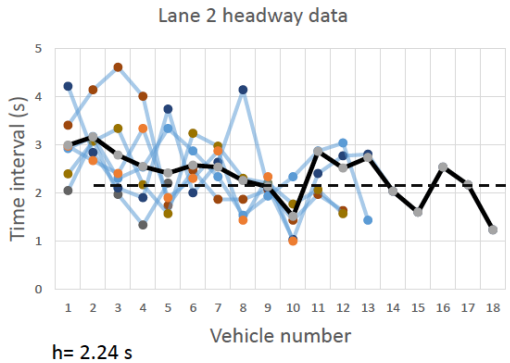


Figure 30: Saturation headway as computed from UAV video and using JAMAR device

Queueing Studies: For a signalized intersection with a queue waiting for the light to change, saturation headway (as discussed above) is a measure of the timing

for vehicles to pass through the intersection. Figure 31 below show another measure of performance for signalized intersections [14,15].

Looking at the graph on the left, as the light turns red a number of vehicles build up in the queue. Once the light turns green, the vehicles start leaving the intersection. If the duration of the green light is long enough, all the vehicles in the queue can make it through the intersection in a single light cycle. This is referred to as under-saturated flow. Comparing this with the graph on the right, the vehicles build up as before, but the green light duration is not long enough for the entire queue to drain out of the intersection and so some number of vehicles is left unserviced at the end of the light cycle. This is referred to as over-saturated flow. With time, this situation can lead to traffic jams and large driver wait times. As such, measurements of queue characteristics and another variable of interest for traffic engineers.

A number UAV experiments were conducted to study this aspect of traffic flow. Figure 32 shows one such intersection located at along US50 in the Fairfax, OH area where several flights were conducted, each recording video data. The tethered UAV was used here in order to be able to stay aloft for extended periods to capture long-duration video data.

As one can see, multiple lines converge down to two as US50 passes through Fairfax. The photo shows an extremely large queue that has built up at the intersection entering Fairfax. In this particular case, because the queue was forming was so long, it was not possible to fly and obtain a view of the entire queue at right angles to the traffic flow as described in Section 3.2.1 above.

Again, following the approach in Figure 19, various software codes and algorithms were tried in order to automate estimation of queueing data as shown in Figure 25, however, because of the highly skewed nature of the footage (see Figure 16), object detection and tracking, and hence counts, were difficult to obtain via AI/computer vision methods. However, it was still possible to process the data manually. In addition, the qualitative aspect of the traffic flow as captured in the videos were also of interest to traffic engineers. The UAV video recording was processed manually for various sequences/periods of light cycles. The tables in Figure 33 show the count data obtained for 11 light cycles during a particularly busy period. As the table shows there is significant unmet demand remaining in the queue at the end of every light cycle. This is typical of oversaturated flow. This data was subsequently used by both ODOT and contractor personnel involved in investigating the intersection to adjust signal light timing cycles to help reduce queue size and increase traffic flow in this region.

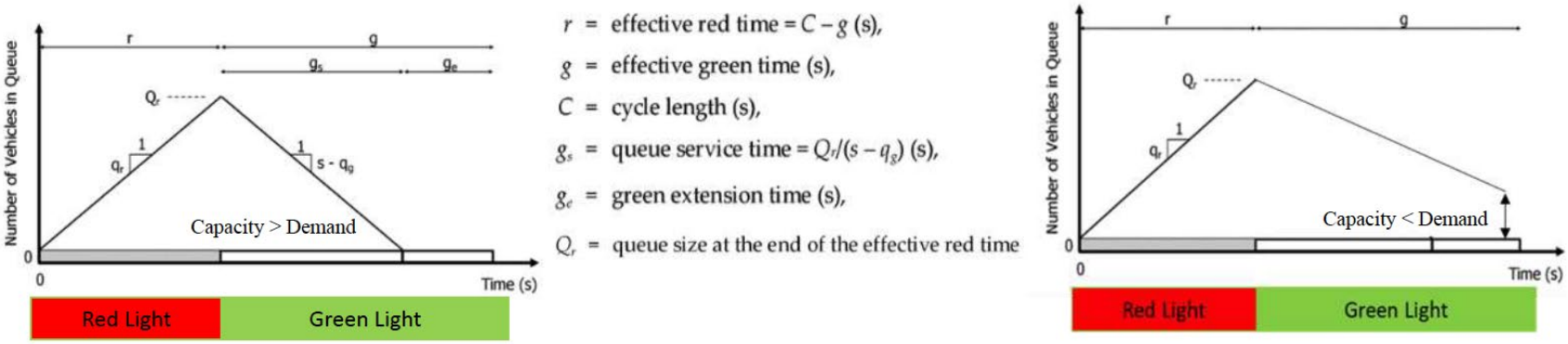


Figure 31: Diagram depicting queuing parameter calculation for under- and over saturated flows [18]

Figure 32: UAV image of built-up queues along US50 in Fairfax, OH area

Period	Stop Line Count	Unmet Demand	Arrival Demand
1	23	18	41
2	26	15	23
3	20	19	24
4	21	19	40
5	23	19	23
6	27	10	37
7	27	7	24
8	18	9	27
9	6	25	31
10	24	16	15
11	26	13	39

Figure 33: Table of queueing data obtained from Fairfax UAV video

A similar set of studies was conducted along I-75 at a series of on- and off-ramps in the Miamisburg, OH area, see Figures 27 and 34. In this case the intersection consisted of two on-ramps and two off-ramps for the north and south lanes of I-75.

This included a series of two traffic lights, one each at the northbound and southbound exchanges.

A system of three UAVs were flown at these intersections covering the periods of heavy traffic early in the morning and later in the afternoon. A tethered UAV was flow nearly continuously at one location, while a pair of untethered platforms was used at the other location. As the battery level reduced, one of the untethered UAVs was landed while the other one took off so as to obtain nearly continuous traffic coverage. Careful records were kept in case the footage from the various UAVs needed to be combined/synchronized later in the analysis.

The UAV video footage was interpreted in the lab. In this case AI/computer vision methods were able to yield better results, owing in large part to the improved site pictures at each signal location (e.g., see global intersection view in Figures 26, 27, 29, 34, etc.). Count data was obtained using software methods and passed on to ODOT traffic engineers. A snippet of the data is provided in the tables of Figure 35. The data indicates the total traffic counts for vehicles making it through the intersections at various periods during both the morning and the afternoon.

While the traffic flow is quite large as can be seen from the stop line count numbers, it is important to note that the unmet demand is zero in all cases meaning that no vehicles had to wait for more than one traffic light cycle to make it through the intersections. This is typical of under-saturated flow. Although the traffic flow in this area was found to be quite large, the data indicated that there was no need for further traffic light timing adjustments.

Figure 34: UAV image of signalized interchange along I-75 in Miamisburg, OH area

Video Title	Time Period (AM)	#Red Lights	Stop Line Count	Unmet Demand	Arrival Demand	Peak Hr Factor	Turning Count
DJI_0002	7:55 - 8:10	7	231	0	231	0.86	147
	8:10 - 8:25	8	233	0	233		188
DJI_0003	8:26 - 8:40	6	184	0	184		157
	8:40 - 8:56	8	264	0	264		164
DJI_0004	8:56 - 9:10	8	211	0	211	0.84	114
	9:10 - 9:25	10	239	0	239		103
DJI_0005	9:25 - 9:40	10	203	0	203		104
	9:40 - 9:50	6	147	0	147		75

Video	Time Period (PM)	#Red Lights	Stop Line Count	Unmet Demand	Arrival Demand	Turning Counts
DJI_0001, DJI_0002, DJI_0003	2:46 - 3:02	6	240	0	240	143
DJI_0004, DJI_0005	3:12 - 3:26	5	189	0	189	112
DJI_0007, DJI_0008, DJI_0009	3:37 - 3:54	6	226	0	226	158
DJI_0010, DJI_0011	4:03 - 4:19	5	216	0	216	118

Figure 35: Tables of traffic parameters data derived from Miamisburg UAV video

3.2.4 Interrupted flow, Roundabouts

Another type of roadway with an interrupted flow is a roundabout. Unlike regular intersections, roundabouts are non-signalized. A typical roundabout is where 3 or more approaching legs merge into a circulating stream surrounded by a central island. Traffic flow through a roundabout is dependent on vehicles in the approaches being able to enter gaps in the circulating stream. The Highway Capacity Manual [14] provides default guidelines for estimating the parameters necessary to evaluate a roundabout. The gap acceptance model is the preferred choice to evaluate the roundabout capacity. The parameters for estimating capacity through gap acceptance model are mean critical headway and mean follow-up headway. The mean critical headway is estimated by using maximum likelihood estimation method by R. Troutbeck [16,17] which requires two parameters: accepted headway and rejected headway. The mean follow-up headway is the mathematical mean of all the follow-up headways estimated.

Figure 36: Multilane roundabout in Dublin, OH

As in previous sections, use of UAVs was an integral part of the research project to estimate these parameters. For this project, the research team worked with ODOT traffic personnel and Kittelson & Associates. A multi-lane roundabout located a Dublin, Ohio on SR71 was chosen to conduct the study (see Figure 36). UAVs were chosen as the data collection tool as they provide wide field-of-view to cover the wide surface area of the roundabout as well as a global overhead view of the entire roundabout and its approaches. Two UAVs DJI Phantom 4 and DJI Matrice 210 were flown alternately to capture 1hr worth of video data between 3:00 PM and 5:00 PM one a typical late fall/early wither traffic day. This provided video data with both good afternoon lighting conditions as well as poor twilight lighting conditions, giving a varied scenario dataset for research purposes. Figures 36 and 37 provide the design and configuration patterns for this roundabout.

Leg	Approaching lanes	Circulating lanes
East leg	2 (1 By pass lane)	2
West leg	2 (1 By pass lane)	1
South leg	2 (1 By pass lane)	2
North leg	1 (1 By pass lane)	2

Figure 37: Dublin roundabout configuration characteristics

One way to collect parameters to estimate mean critical headway and mean follow-up headway is to use the click-based methodology [24]. In this method, the user looks at various spots of the roundabout and manually collects the timestamps of vehicles hitting that spot. Although, it is possible to use this method for collecting the required parameters for a single lane roundabout, its impractical to process several lanes for a multi-lane roundabout. In this project, we proposed a new way of collecting the required parameters. This method was termed as the gap-based method [9]. This method focuses on the time differences created by the vehicles in the circulating stream instead of collecting timestamps. Although the computational efforts for gap-based method are more, its more accurate at estimating the required parameters for a multi-lane roundabout.

In this project, use of AI and computer vision to develop a tool to collect the required parameters was also investigated. Figure 38 outlines the algorithm designed to collect accepted headway, rejected headway and follow-up headway for each lane of each leg of the roundabout. Details can be found in [9].

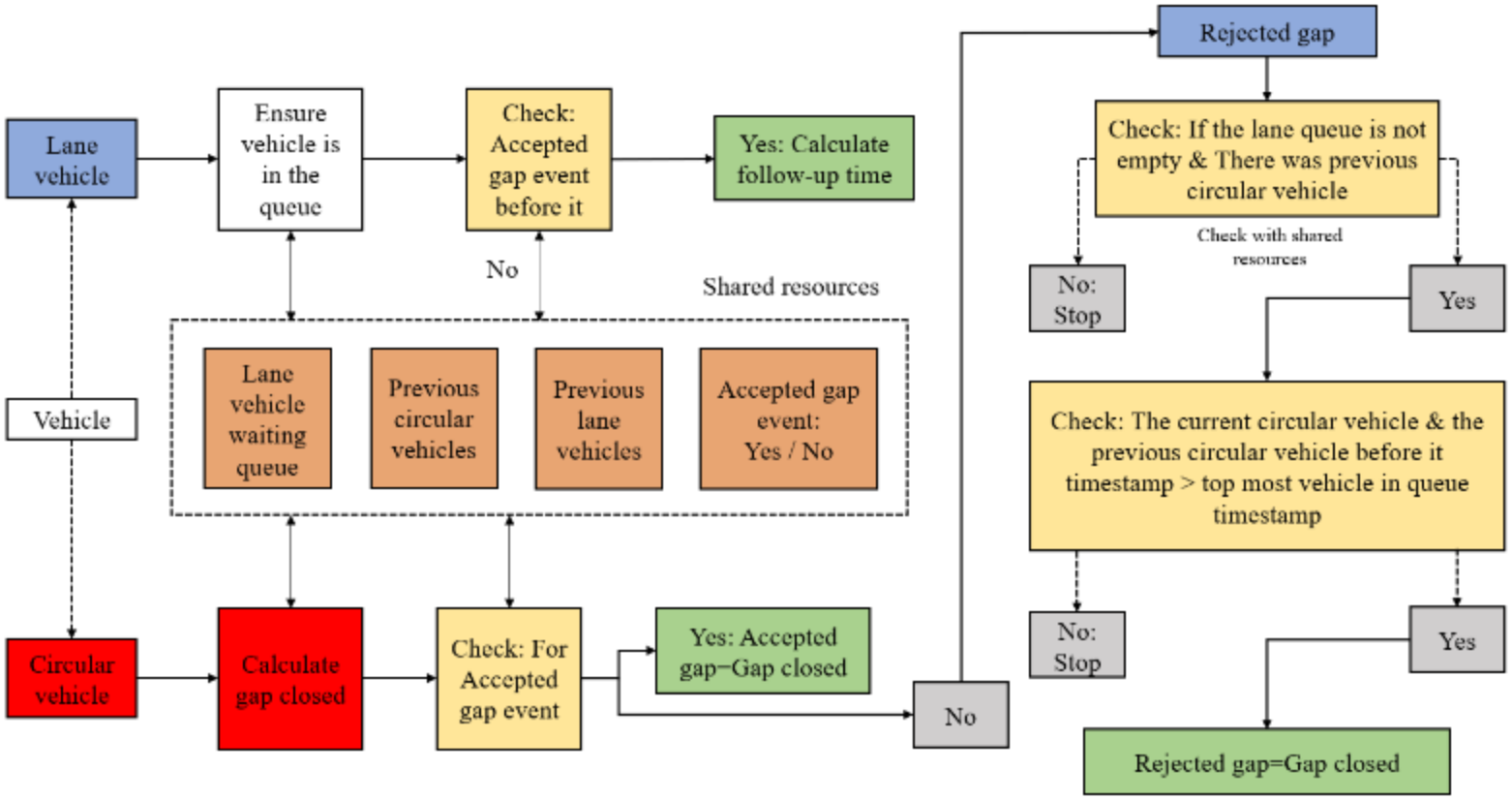
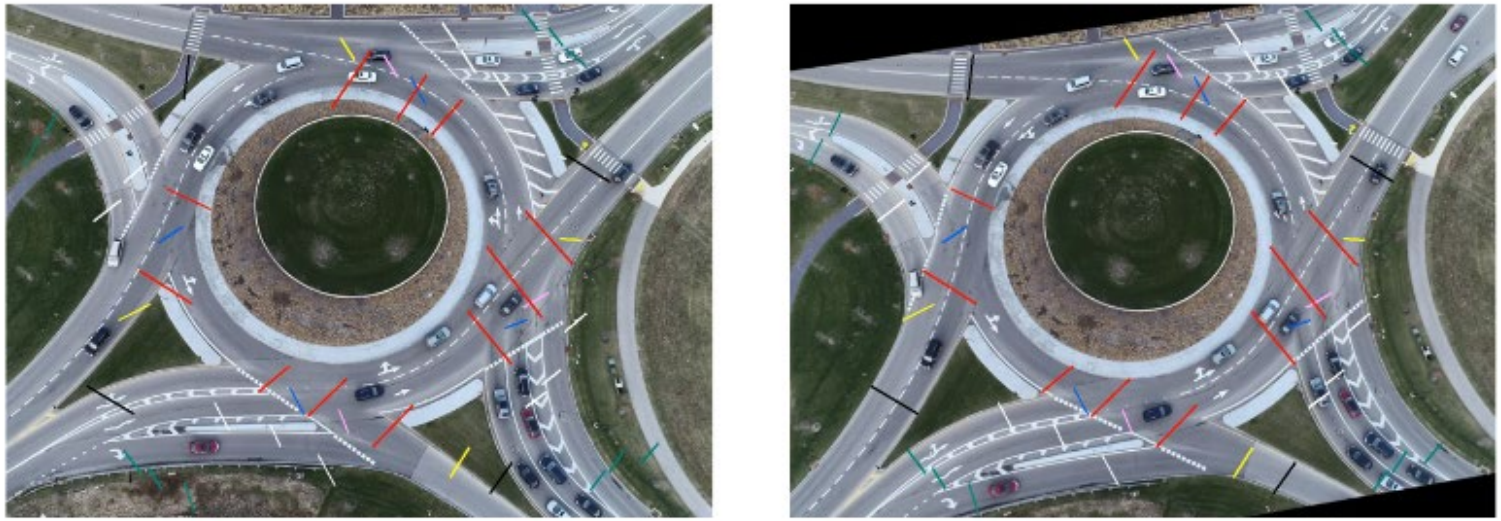


Figure 38: AI/Computer vision algorithm flow chart

In order to run the run the main algorithm, a pre-processing of the video data is needed to stabilize the video taken from UAVs and to mark the approaching lane and circulating lanes to collect the necessary parameters. The images in Figure 39 show the before and after results of image registration and drawing unit. Figure 40 shows an example of AI/Computer vision processing.



Video screenshot (without image registration)

Video screenshot (with image registration)

Figure 39: Image stabilization/registration to counteract UAV drift

At the core of the main unit is object detection and tracking. To detect vehicles from an overhead view, the detector uses a deep neural network trained on a custom dataset. RetinaNet [18,19] object detector is used to detect vehicles at each frame. Tracking is done using a custom designed tracker to track vehicles from one frame to another. A combination of detection and tracking allows to keep track of vehicles throughout the vehicles. More details related to this can be found in thesis [9] and paper [20] from the reference.

The collected parameters are compared in a two-step process. First, the parameters are compared with manually collected parameters and then the parameters are collected with the default values from the HCM 2010 and HCM 2016 manuals [14]. For confirmation against manual values, the difference between accepted headway, follow-up headway and largest rejected headway are checked. This step is critical to follow, to validate the results of software-defined technique to keep the markings of entering, circulating and exit lanes consistent with observers who manually process the video data. This iterative process helps to reconfigure the markers to maintain accuracy in data collection throughout the video. A tolerance of +/- 1 second is considered, if 50% or more points are within the tolerance level then software defined approach is showing good results and can be applied on all the entire video data. The graphs in Figure 41 below show comparisons for each lane of East leg for one video, this shows that the software defined approach is collecting parameters which closely match the manual values.

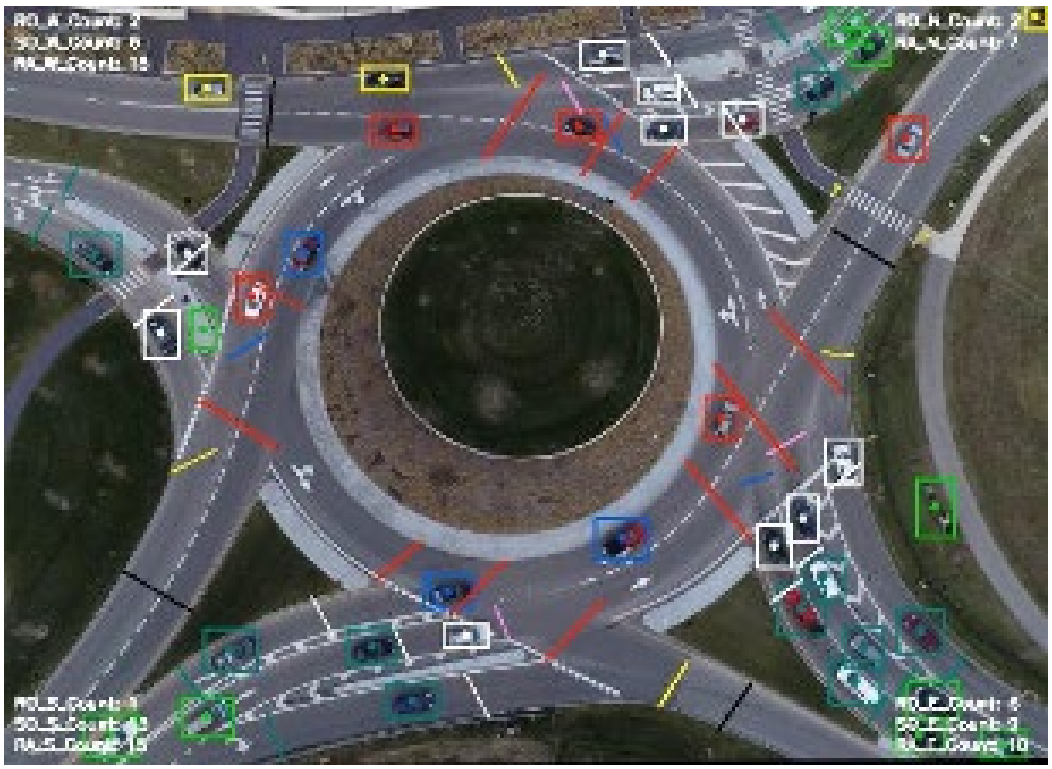
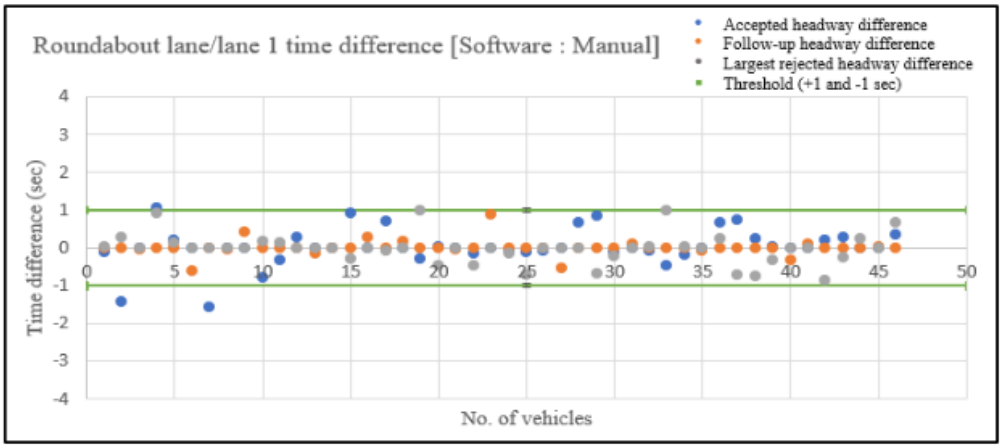
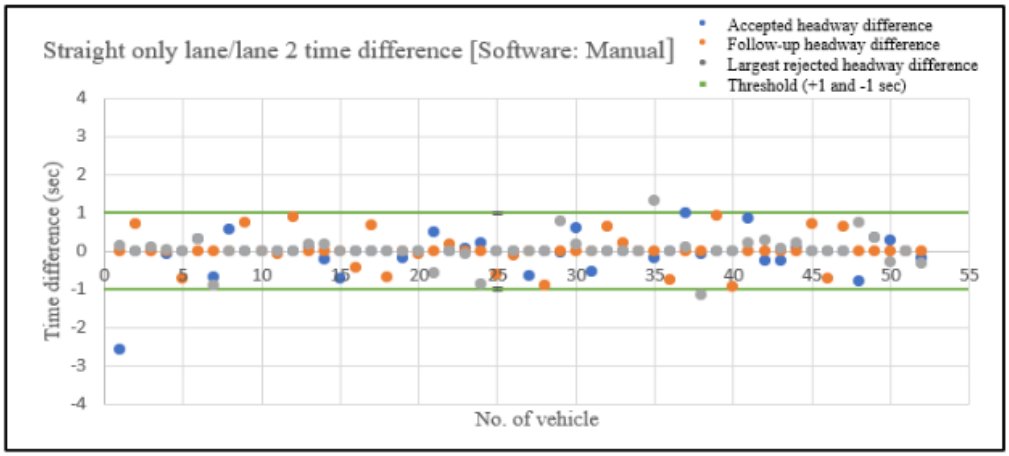


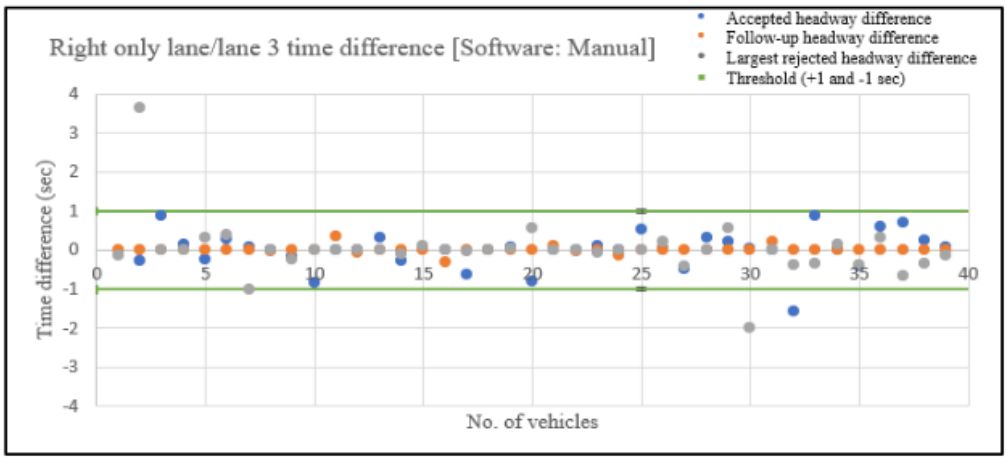
Figure 40: Image from computer processed UAV roundabout video at Dublin, OH



(a) Video 1: Lane 1



(b) Video 1: Lane 2



(c) Video 1: Lane 3

Figure 41: Lanewise accuracy results for AI/Computer vision Timing estimates

The collected parameters are then used to estimate the mean critical headway and the mean follow-up headway. Figure 42 shows the comparison of manually estimated values and values estimated by software defined method.

Video	Lane	Mean critical headway		Mean follow-up headway	
		Manual	Software	Manual	Software
Video 1	Roundabout\Lane 1	4.357	4.149	2.31	2.307
Video 1	Straight only\Lane 2	4.438	4.323	2.487	2.533
Video 1	Right only \Lane 3	3.041	2.787	2.77	2.785

Lane	Mean critical headway			Mean follow-up headway		
	Calculated	HCM 6	HCM 2010	Calculated	HCM 6	HCM 2010
Roundabout\Lane 1	4.085	4.65	4.293	2.476	2.667	3.186
Straight only\Lane 2	3.750	4.32	4.113	2.481	2.536	3.186
Right only\Lane 3	2.811	4.32	4.113	3.213	2.536	3.186

Figure 42: Accuracy results for AI/Computer vision headway estimates

After comparisons between the manually estimated values and the software defined values give positive results, the software is applied of all the collected video data. The results from this step are checked against the HCM default value. Below table shows the comparison of mean critical headway and mean follow-up headway for each lane of East leg for the one hour of collected video data against the HCM 2016 and HCM 2010 values.

These estimated values can be used to form a capacity model equation. The equation's below are taken from the HCM 2016 manual and shows how to form a capacity model equation with a configuration of 2 approach lanes with 2 circulating lane.

$$A = \frac{3600}{t_f}$$

Here t_f is the mean follow-up headway.

$$B = \frac{t_c - ({}^{t_f}/_{2})}{3600}$$

Here t_c is the mean critical headway.

The capacity model equation can be written as,

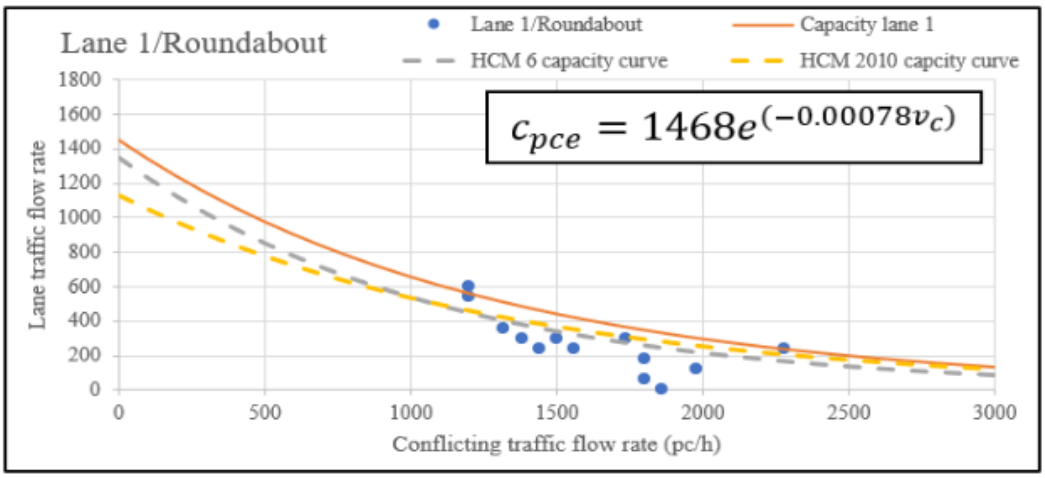
$$C_{pce} = Ae^{(-Bv_c)}$$

Lane	Parameter A			Parameter B		
	Calculated	HCM 6	HCM 2010	Calculated	HCM 6	HCM 2010
Roundabout\Lane 1	1454	1350	1130	0.000791	0.00085	0.00075
Straight only\Lane 2	1451	1420	1130	0.000697	0.00092	0.00070
Right only\Lane 3	1120	1420	1130	0.000334	0.00092	0.00070

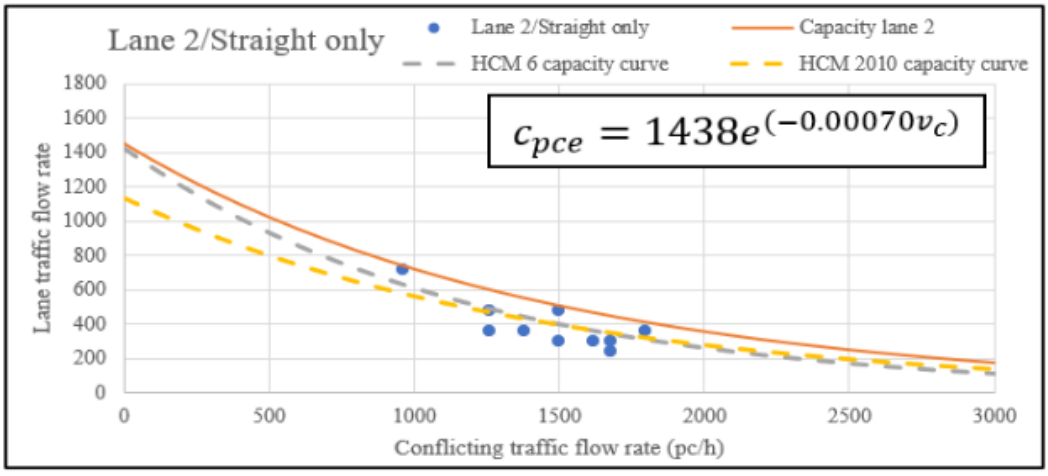
Figure 43: Estimated roundabout performance parameters compared to FHWA default values

Using these we can get the capacity model parameters for the East leg. Figure 43 shows the estimated capacity model parameters when compared with the default capacity model parameters for HCM 2016 and 2010.

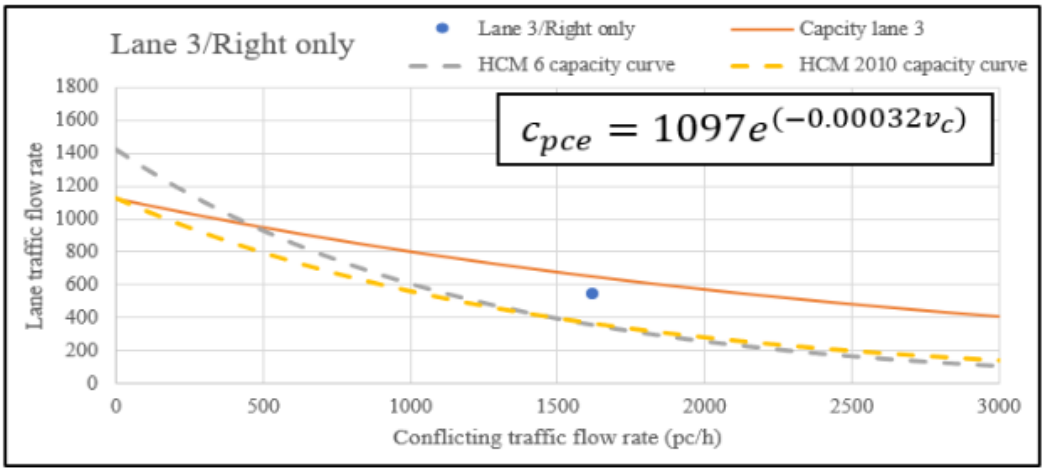
Capacity model for each leg can be checked by plotting the capacity model curve on an approaching flowrate vs circulating flowrate graph. The flowrate for this evaluation must collected queuing condition. Manually checking for queues can be a very arduous process, the software however is designed to check for queues at each lane and can be easily configured to check for minute interval flowrate which show 100% queuing. The graphs in Figure 44 show the comparison of flowrate for each lane of the East leg against the evaluated capacity model equation curve and the HCM 2016 and 2010 default capacity model equation curve. Based on the results from this project, the developed software was able to confirm that the parameters from the roundabout closely follow the default parameters from HCM 2016 manual and helped ODOT Traffic department in confirming their calibration values.



(a) Flow rate: Lane 1



(b) Flow rate: Lane 2



(c) Flow rate: Lane 3

Figure 44: Estimated roundabout performance parameters compared to FHWA default values

3.2.5 Multi-UAV Operations, Traffic Corridors

The traffic monitoring studies described above dealt primarily with a single intersection or section of highway. Traffic flows through an entire corridor, consisting of several intersections/interchanges, are also of interest to traffic engineers. As a final example of UAV-based traffic monitoring explored during the course of this project, we present the corridor as shown in Figure 45. This includes a stretch of SR741 just south of SR73 as it moves north-south through the Springboro, OH area. It consists of several intersections, some signalized and some not, as well as a sequence of five schools (elementary, middle, and high schools). Traffic flow through this corridor was one lane in each direction with extra turning lanes at selected intersections. As a result of the high density of schools along this stretch, there were considerable traffic backups both in the morning and in the afternoon as the schools began and ended their days.

Working with school officials, city officials and traffic engineers developed and implemented various mitigation strategies such as staggering start and end times and deploying loading and unloading procedures in the various school parking lots. However, traffic difficulties persisted.

The research team, working together with personnel Ohio UAS Center, developed a plan to simultaneously deploy six UAVs with a total of eight cameras spanning the entire corridor in order to obtain a continuous monitoring of the traffic flow patterns in the region on a typical day. The UAVs were flown as continuously as possible throughout the morning unloading/arrival and afternoon loading/departure time periods. In order to maintain safe operation and coordination of all flights, extensive logistical planning and recordkeeping was needed. All pilots were briefed on the plan, were in constant two-way communication throughout the day, and were careful to maintain timing records so that UAV footage from the various locations could be synchronized afterwards. In addition, lessons learned during the morning flight operations were folded into the afternoon flight operations.

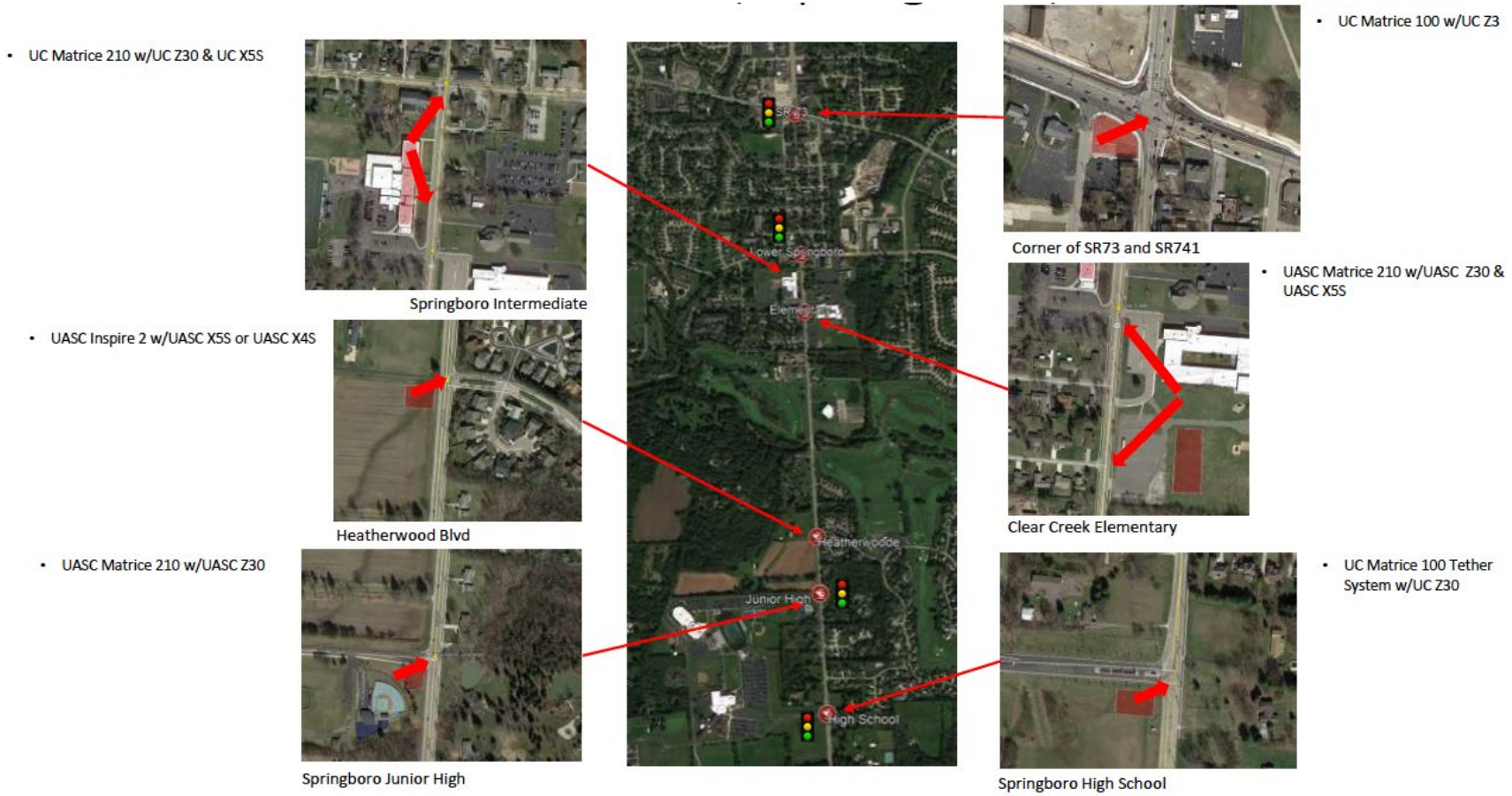


Figure 45: Overview of flight operations in the Springboro OH-741 corridor

Figure 45 gives an overview of the entire corridor along with locations for traffic lights and the locations from which the six UAVs were deployed. The fleet of UAVs included 2 DJI M100s, 1 DJI Inspire, 2 M210s and a third tethered DJI M100. Four of the UAVs carried single cameras and the two M210s were outfitted with dual gimbals which allowed them to carry 2 cameras aloft each pointing in different directions. This arrangement provided enough spacing for safe flight operations while at the same time offering complete video coverage of the corridor traffic.

A total of 218 videos were recorded between 6:45 AM and 5:00 PM totaling more than 46 hours of footage, and 0.5 TB in size. The data at each location was processed manually by humans in the lab to obtain queues and turning counts in each direction for each intersection observed. The methodologies outlined in the Ohio Traffic Forecasting Manual were used in interpreting the videos and obtaining traffic data [16].

An example of a typical site picture is shown in Figure 46 taken at the northern most end of the corridor where SR471 intersects SR73. Examples of turning count plots obtained for 15 minute windows are given in Figure 47. The data shown in Figure 32 is for the traffic flow at the southernmost end of the corridor located at the Springboro HS entrance. As can be seen from the graphs, the team was able to characterize the traffic count patterns for both the morning and the afternoon periods. The plots clearly show the large influx of traffic moving from main road (i.e., SR741) into the high school in the morning at the start of the school day and leaving the high school to get onto the main road (i.e., SR471) in the afternoon at the end of the school day.

Similar plots were obtained for all intersections monitored and the data was provided to both the City of Springboro and ODOT. Queuing data was also obtained at all intersections. At several intersections, the queues were so long that they extended beyond the site picture of a single UAV. In these cases, video from adjacent UAV cameras was synchronized enabling processors to understand the entire queue extent and behavior at each intersection. This technique allowed for queue counts to be developed which would not have otherwise been possible if only a single UAV had been deployed, lending further benefit to the use of a multi-UAV operation.

Figure 48 provides an overall summary of the study findings. As can be seen, the corridor traffic patterns were found to break down into two groupings: (1) The intersections to the south at the high school and junior high school, while heavily trafficked, generally flowed well and resulted in small cues. (2) However, the intersections to the north at the elementary and middle schools were typically much more congested and extensive queues developed that several of these locations.

In addition, it was found that these two regions interacted in the sense that the majority of the school traffic arrived from the suburbs in the north in the morning and departed to the north to return to the suburbs in the afternoon. As a result, the traffic leaving the high school and the junior high school to the south in the afternoon headed north and largely exacerbated the traffic patterns in the northern region at

the elementary and middle schools. Further it was observed that drivers were using the various side streets as a cut-throughs at various locations to work around the traffic backups. Thus, in addition to the quantitative data obtained as shown in Figure 47, the ability to view the flow of traffic qualitatively throughout the entire corridor provided a great benefit to traffic engineers who are able to use the global, corridor-wide information provided to identify and focus their attention at specific bottleneck points in the corridor and make detailed adjustments at these locations to further reduce traffic backups in the area.

In the end, the combined qualitative and quantitative data obtained from this study were provided to traffic engineers and Springboro City officials who used the information to both further adjust signal timing and make refinements to their drop-off and pick-up plans for each of the schools in the area.

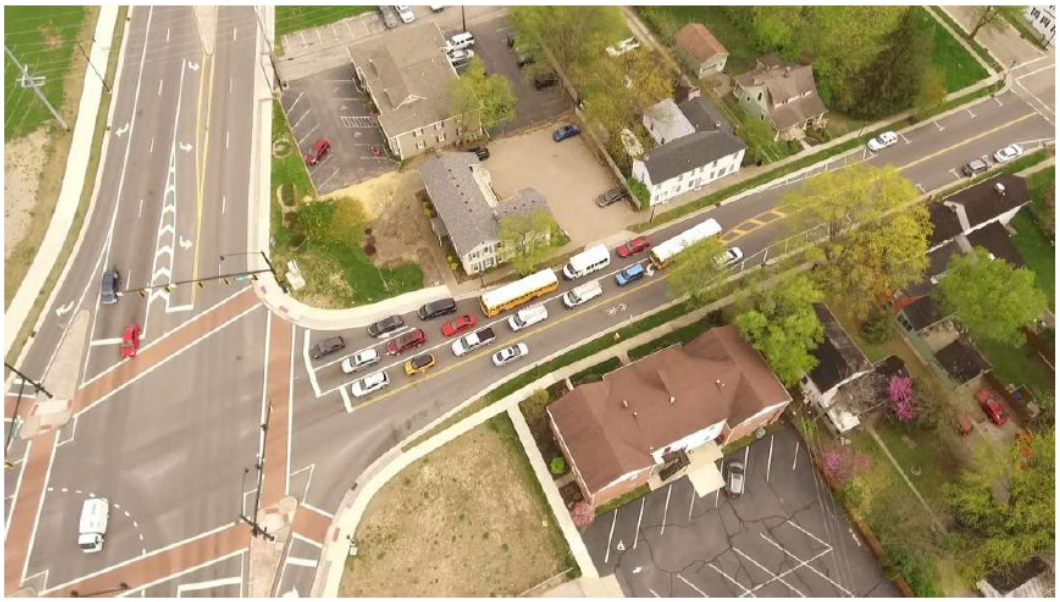


Figure 46: Site picture obtained by UAV at intersection of SR-714 and SR-73, northern terminus

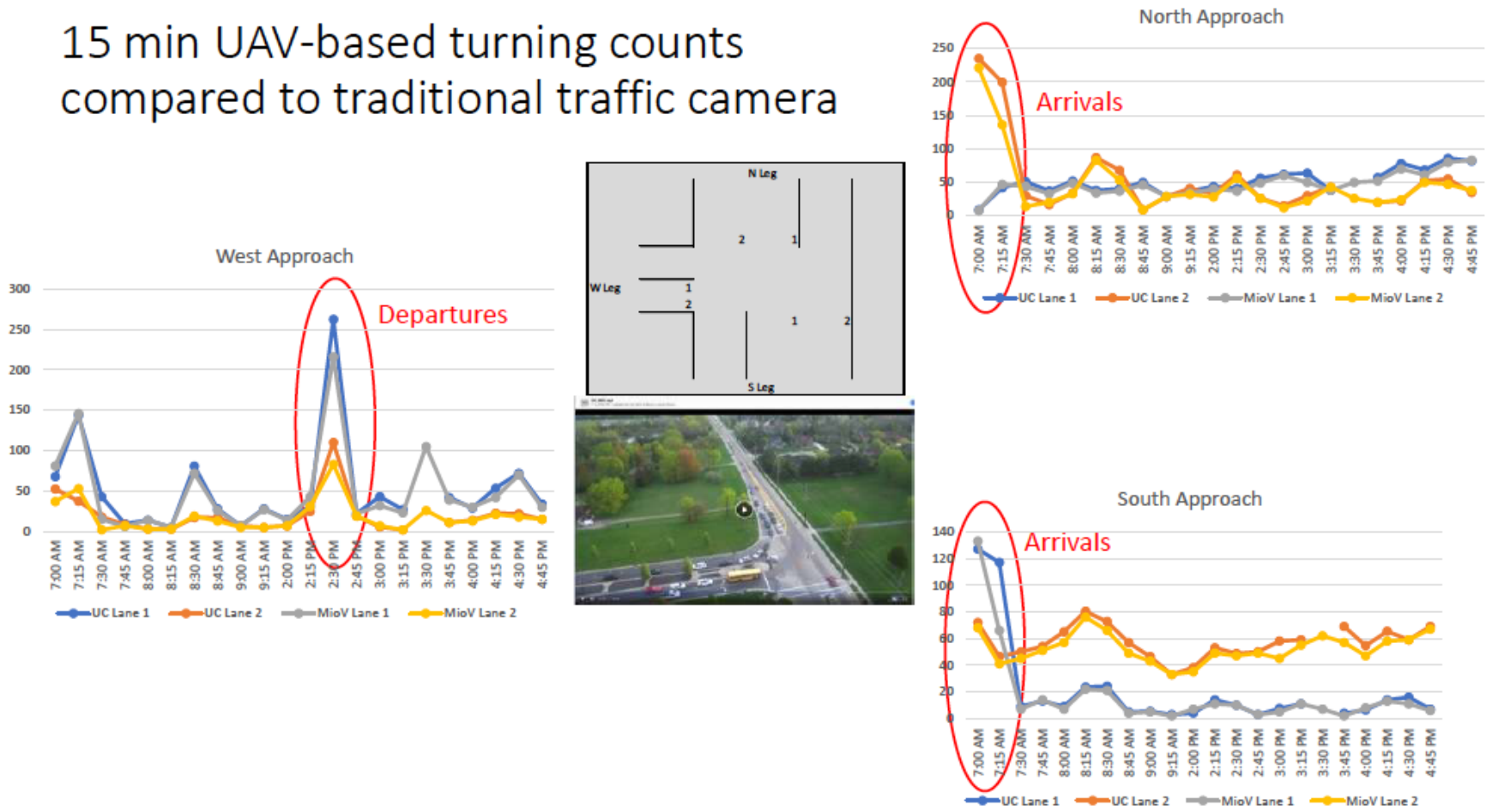


Figure 47: Example of turning count data obtained at southern terminus intersection at Springboro HS

Overall Observations and Results

Largely free moving traffic with minimal queues

Traffic view not easily observed, likely minimal queues

Significant traffic/queues observed, but did not last more than 2 light cycles

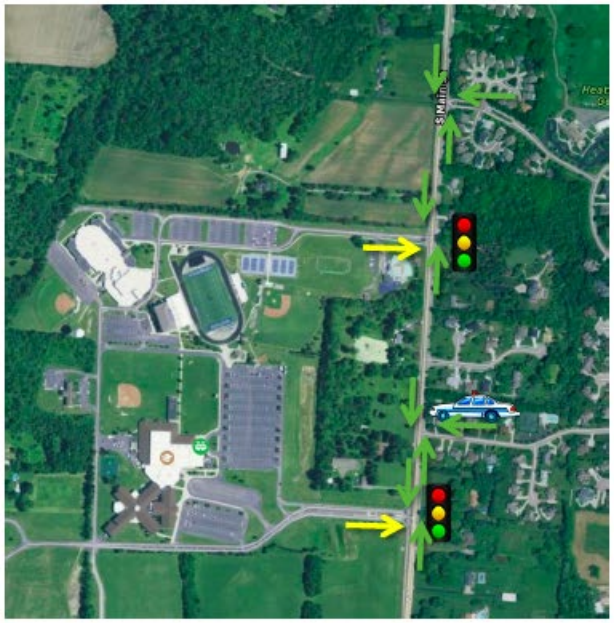


Figure 48: Overall findings of SR741 corridor traffic study

The main conclusions which result from the research efforts reported above on using UAVs for traffic monitoring can be summarized as follows:

1. UAVs proved useful in obtaining information in a wide variety of settings and at various levels of detail that proved useful to traffic monitoring and traffic engineers.
2. UAVs can be used to provide advantages for traffic monitoring like wider field of view, higher resolution and more flexibility in altitude and camera angle over traditional ATR/roadway sensors, cameras mounted on poles or manual processes (such as use of the Jamar).
3. With additional effort, effective AI/computer vision algorithms can be used to design software tools for automated traffic monitoring applications. Proof of concept has been established.
4. Tools developed for un-interrupted, interrupted, and roundabout traffic flow collect various parameters and provide good accuracy as compared to manually processing. Computed parameter's accuracy has also been validated against FHWA provided default parameters.
5. Many of these same algorithms developed for use with UAV footage could be adapted to video obtained from pole-mounted cameras permanently installed at intersections and along roadways (e.g., ODOT Milestone system).
6. Computer vision using deep learning can be leveraged to provide higher level parameter estimation like custom classification and capacity evaluation going beyond obtaining mere counts, etc.

3.3 Aerial Mapping and Construction Site Monitoring

The use of UAVs as airborne platforms for aerial mapping was identified early on in this project. The ability to augment these operations with either or both Ground Control Points (GCPs) and RTK GPS, afforded the possibility to include high resolution spatial information into the photogrammetry processing in order to obtain very accuracy 2D and 3D modeling even approaching survey quality. This required extensive mission planning in order to obtain image data sets with the necessary fidelity in terms of Ground sampling distance and image overlap. These latter qualities were available due to the use of flight planning software, such as DJI Go, that allowed for accurate preplanned flight control/operations.

In addition, the use of UAVs increased safety for both ODOT employees and the traveling public in that they could be operated in areas where it was unsafe for humans (e.g., active landslides, etc.) and/or could be conducted aerially above

ongoing/active construction sites without interrupting ground operations. They can also be rapidly deployed on an as needed basis to a wide variety of locations. Finally, the range of cameras/lenses available coupled with the range of altitudes (up to 400ft) provided site pictures and image data sets which can range from close in to panoramic and for data sets collected at multiple points in time so as to allow for tracking of operations, construction activity/progress, etc.

UAVs do have some limitations, however. These include primarily: (1) weather conditions must be advantageous (e.g., there are operational limitations on wind, precipitation, etc.); (2) coordination with surveyor is necessary to plan and mark a select number of GCPs in order to obtain the high level of accurate models, and (3) the images obtained must be post processed by the application of photogrammetry software (e.g., Pix4D) in order to obtain global 2D and 3D computer models. The latter software is readily available commercially and, as discussed below, the research team did spend considerable time investigating its use.

The results reported here borrow heavily from the SOPs developed by the research team for use by Ohio UAS Center pilots and personnel as part of this project [5,24] as well as the theses and papers generated by the various graduate students involved in this research project [25,27]. The reader is referred to these documents for additional information and details.

3.3.1 General Flight Planning and Vehicle/Sensor Package Considerations

As outlined in the SOPs [5,24] and discussed above in Section 3.1, vehicles such as the DJI Matrice 100, and/or the Matrice 210 RTK were the primary UAV platforms used for aerial mapping and construction site monitoring. Later in the project, DJI Phantom 4RTKs, obtained by the Ohio UAS Center, were also added to the fleet for aerial mapping applications. These platforms should be coupled with high resolution photographic cameras which are capable of providing a wide range of functionality such as optical zoom, image/exposure settings, etc. based on the flight conditions and mission objectives. The Z3 and XTR camera were both used but the higher resolution of the X5 and X5s allowed for more accurate models. Many of the experiments that were run showed ways to improve these models and were mostly associated with the application DJI GS Pro.

The GS Pro app allowed for easy planning of missions, but had a lot of settings that would need to be adjusted depending on the mission. Figure 49 shows some of the interface and some of the settings that could be changed.

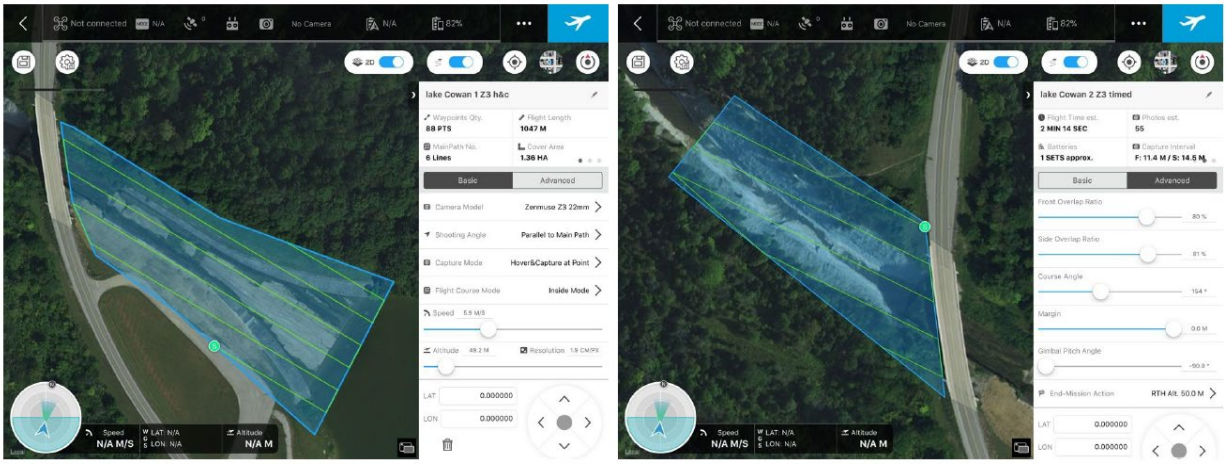


Figure 49: DJI GS Pro app

One of the most important settings was properly setting up the camera model. If the correct camera model was not inputted the model may have bad overlap which causes issues creating a model. When new lenses were added or the zoom changed on the camera the appropriate model would need to be updated. The importance of the overlap between images was experimentally shown to need to be approximately 70% or more. Changes in terrain would cause this overlap to change so appropriate measures would need to be considered prior to operations. The resolution or Ground Sampling Distance (GSD) would also need to be considered when planning missions. This GSD is the distance, measured in centimeters, per pixel. The smaller this value the more resolution detail the end model would have. The down side with having too small of a GSD is the time it would take to fly the mission and the flight altitude. If the operation was happening near many trees the mission would not be able to fly as low.

The GS Pro app also had limitation during the beginning of this project. Most notable is the limitation of waypoints. These waypoints were the GPS locations that the sUAS would follow. Many missions would be set up in a grid pattern and these waypoints would be the corners of these grid. If the mission was too large the number of waypoints would exceed 99 and would need to be split into multiple missions. These missions would need to be planned out to ensure each mission had the appropriate overlap and the same GSD.

The capture mode would see the biggest impact on the number of waypoints. If the mission was set up as “Capture at Equal Distance” the vehicle would fly at a constant speed and trigger the camera while flying. This did cause blurred images if the vehicle speed was too fast. The cameras also had a buffer time and so if many images were trying to be taken too quickly not all the images would save. Trial missions were conducted that led to Figure 50. This image shows the possible GSD and Overlap at a given speed for the X5 camera. So if the team wanted to operate at faster speeds we would have to adjust the overlap and GSD accordingly.

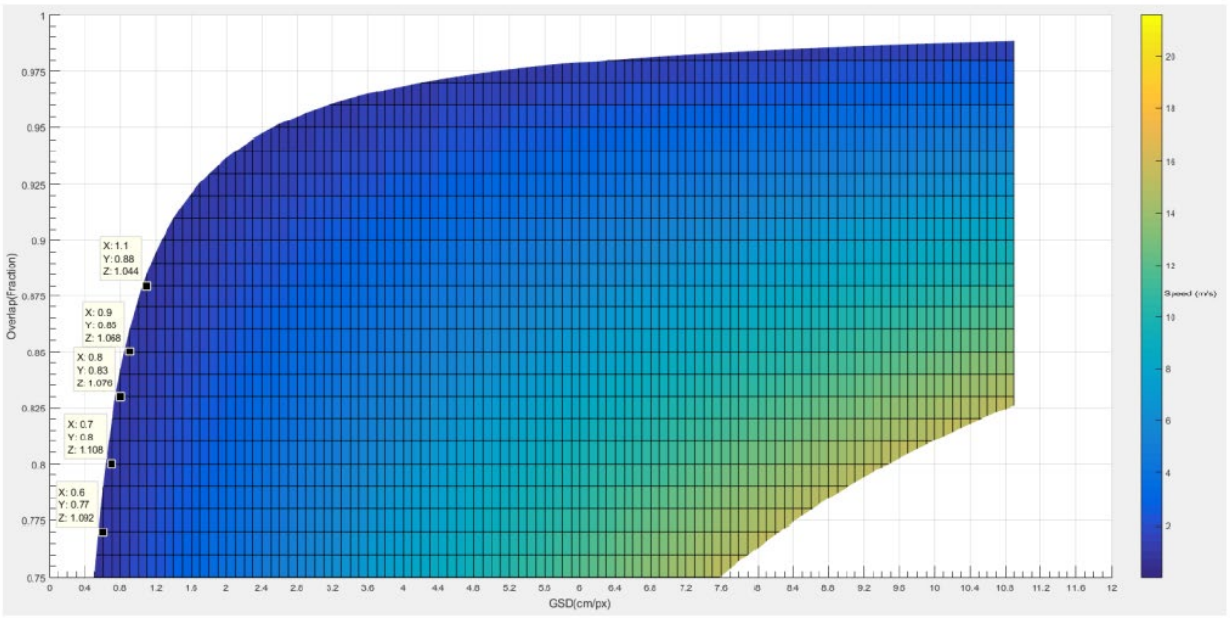


Figure 50: Map/Surface of GSD/Overlap/UAV speed interrelation

"Hover and Capture" was another capture mode used that helped create good models. This mode would cause the sUAS to hover and take an image before moving to the next location. This would cause each location the image to be taken to be a waypoint. As mentioned above the limitation of 99 waypoints would only allow for 99 images to be taken during each flight. This restriction of 99 waypoints would limit the use of "Hover and Capture" mode to smaller operations only. This waypoint limitation would be removed during the project, but the "Hover and Capture" mode still would take longer time to complete as the vehicle stopped constantly.

3.3.2 Use of RTK and GCPs

Improvements on creating models not only came down to camera performance but also the scaling and orientation of the images. Ground Control Points (GCPs) were used as a geo-referenced point during the mission. The GCPs would be placed prior to operations and have their locations recorded. The more accurate the more accurate the GCP location is the more accurate the project would be. Multiple GCPs would be distributed throughout the area of interest and be similar to one of the targets shown in Figure 51.

Figure 51: GCP markers

The GCPs also help the model work faster since there is a unique feature and can also be used as a manual tie point.

The GCPs would need to have their location known for it to help improve the model. To get the most accurate position RTK systems would be used. RTK or Real Time Kinematic is a type of GPS that utilized a base station that is known to help with corrections. The GCPs would be either measured by a survey crew or the team would use an RTK system out in the field such as the Matrice 210 RTK.

The RTK base station would be set up and left for several minutes why it got a more precise location. The "rover" would then be placed over the GCPs and recode their location. This process would be completed for each GCP.

3.3.3 2D and 3D Models and Model Accuracies

The accuracies of the outputs generated by Pix4D are calculated using the accuracy metrics shown in the equations below. Using the equations given below the error values for individual measurements (planimetric and pointwise) and the model were calculated.

$$\text{Error } (E_i) = d_i - \hat{d}_i$$

$$\text{Absolute Error } (AE_i) = \left|d_i - \hat{d}_i\right|$$

$$\text{Mean Error (ME)} = \frac{1}{N}\sum_{i=1}^{N} d_i - \hat{d}_i$$

$$\text{Mean Absolute Error (MAE)} = \frac{1}{N}\sum_{i=1}^{N}\left|d_i - \hat{d}_i\right|$$

$$\text{Root Mean Square error (RMSE)} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(d_i - \hat{d}_i)^2}$$

$$\text{Standard Deviation (SD)} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(E_i - \bar{E})^2}$$

The variables seen in Equation 8-14 are defined as follows:

- d_i: Actual measurement/output
- $\widehat{d_i}$: 3D point cloud measurement/output
- N: Total number of measurements
- E_i: Observed error
- $\overline{E}$: Mean of error values

The calculated error values are presented in a plot and the colors of the points in the plot are formatted based on the error values. When the measured error values are greater than 0.07 feet the points in the plots are colored red and when the error values are less than 0.07 feet the points are colored green. The 0.07 feet constraint value was set to ensure the recorded error values agree with the accepted 3D model error values required by ODOT surveyors [30]. Figure 52 shows ODOT's allowable error values for varying surveying classes in the horizontal and vertical planes [30].

Planimetric Accuracy Class	Recommended Use	Maximum Allowable RMSE (ft.)
Class I	Projects that require Class I planimetric features listed in Appendix A to be identified and mapped (i.e.: design engineering projects)	0.30
Class II	Projects that require Class II planimetric features listed in Appendix A to be identified and mapped (i.e.: planning studies)	1.00

DTM Accuracy Class	Recommended Use	Maximum Allowable Average Dz (feet)	Maximum Allowable RMSE (feet)
Class A	Paved areas	± 0.07	0.16
Class B	Vegetated areas outside of pavement that are maintained at a minimum biannual frequency (i.e.: farm fields, residential yards, roadside R/W, etcetera)	± 0.25	0.32
Class C	Vegetated areas that are not maintained	± 0.50	0.50
Class D	Areas where vertical accuracy is not critical or warranted (i.e.: planning engineering projects)	± 1.00	1.00

Figure 52: ODOT's Maximum Allowable Horizontal (Left) and Vertical (Right) Surveying Error Measurements [30]

3.3.4 Application Examples

SR266 Pre-Split Area Measurement, Stockport, OH: This section presents the case study conducted at the intersection of Point Lookout Road and the newly constructed SR266 in Stockport, OH located in ODOT District 10. The objective of this study was to measure the cross-sectional areas of pre-splits at the intersection using UAVs and 3D modeling techniques. The results were provided to ODOT surveyors and construction site personnel. The areas measured using the 3D model were compared to the values estimated by the ODOT plan sheet as well as a third-party surveyor.

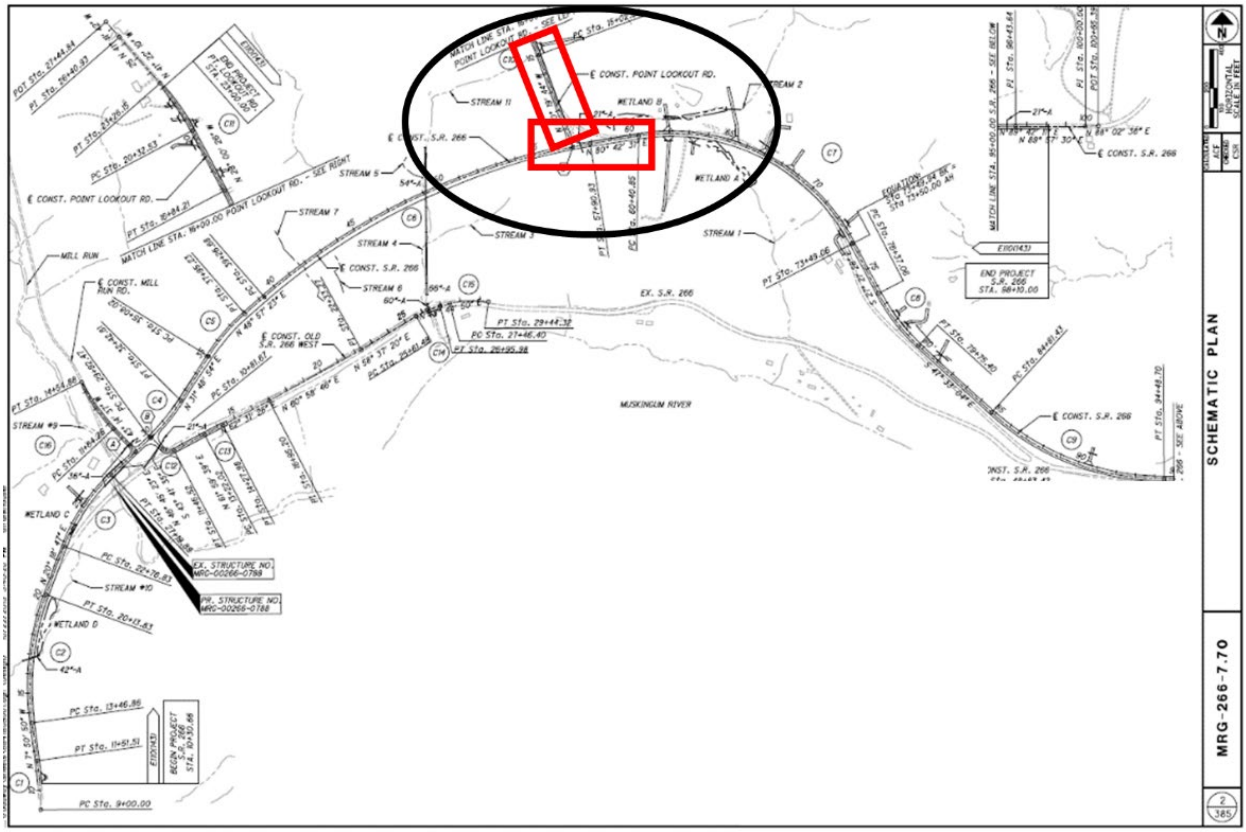


Figure 53: Plan Sheet of SR266 Construction Project with Area Mapped Circled

The plan sheets of the area mapped were analyzed prior to designing flight plans. Figure 53 shows the plan sheet of the project with the area to be mapped circled. Using the information present in the plan sheets a rough schematic of the area mapped was generated and shown in Figure 54. The rough schematic aided in dividing the area and estimating the rough positions for the GCPs. The area to be mapped was divided into five regions and 25 GCPs were laid out in the region. The area was divided to efficiently map the region using the sUAS. Using the elevation profiles of the region the flight plans designed considered the elevation changes and these were reflected in the flight heights and the image overlaps set for each mission.

Figure 55 shows the visual image of the area that was mapped and the takeoff points for the missions.

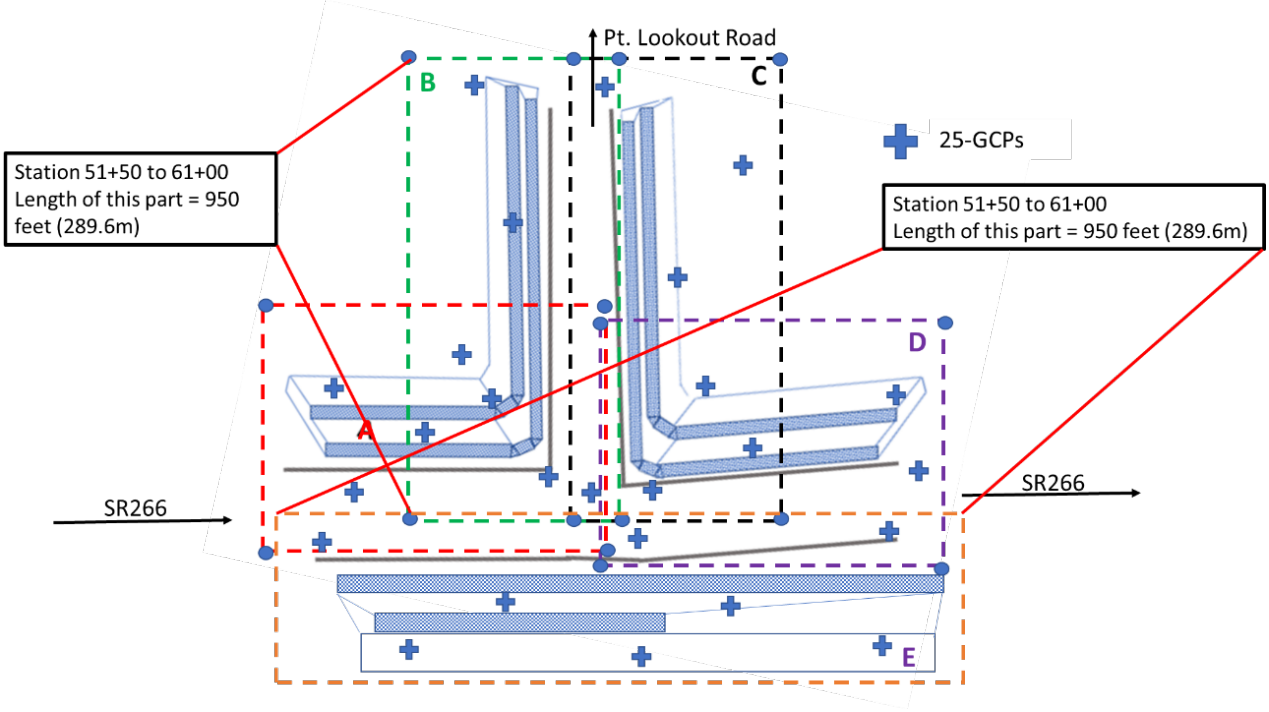


Figure 54: Rough Schematic of the Intersection Mapped

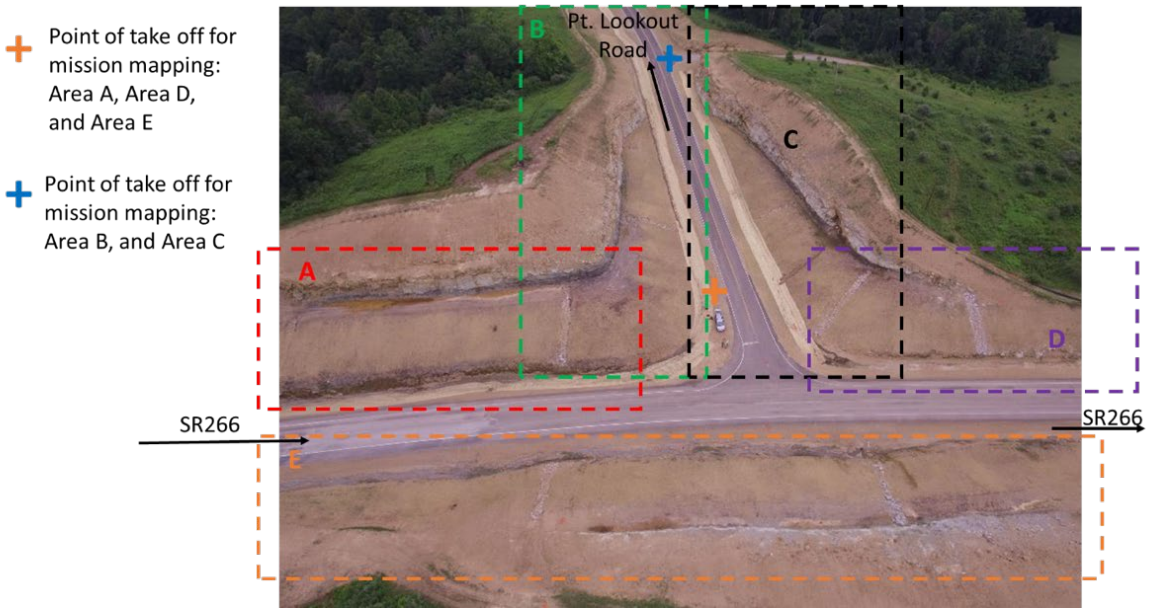


Figure 55: Visual Image of the Regions Mapped with sUAS Take Off Point Marked to Compensate for Elevation Changes

The images of the region were captured using DJI Matrice 100 sUAS and the DJI Z3 camera. The missions were planned on the DJI GS Pro flight planning application, where the mode of capture was set to equal timed mode and the images were captured in a grid pattern. Figure 56 shows the summary of the missions run at the site. The sUAS mapped a total area of 4.47 hectares. Figure 57 shows the screenshot of the flight plans for area E of the intersection in the DJI GS Pro app.

Area	Mission Type	GSD (cm/px)	Altitude (m)	FO/SO(%)	Flight Time	Recorded Photos
A	Parallel	1cm/px	25.5m	93%/93%	1hr 20min	1530
	Perpendicular	1cm/px	25.5m	93%/93%	1hr 30min	1693
B	Parallel	1cm/px	25.5m	93%/93%	1hr	1182
	Perpendicular	1cm/px	25.5m	93%/93%	1hr 10 min	1389
C	Parallel	1cm/px	25.5m	93%/93%	1hr 10 min	1547
	Perpendicular	1cm/px	25.5m	93%/93%	1hr 10 min	1694
D	Parallel	1.1cm/px	28m	93%/93%	30min	779
	Perpendicular	1.1cm/px	28m	93%/93%	30min	748
E	Parallel	1.8cm/px	46m	93%/93%	32min	832
	Perpendicular	1.8cm/px	46m	93%/93%	33min	979

Figure 56: SR266 flight operations mission characteristics

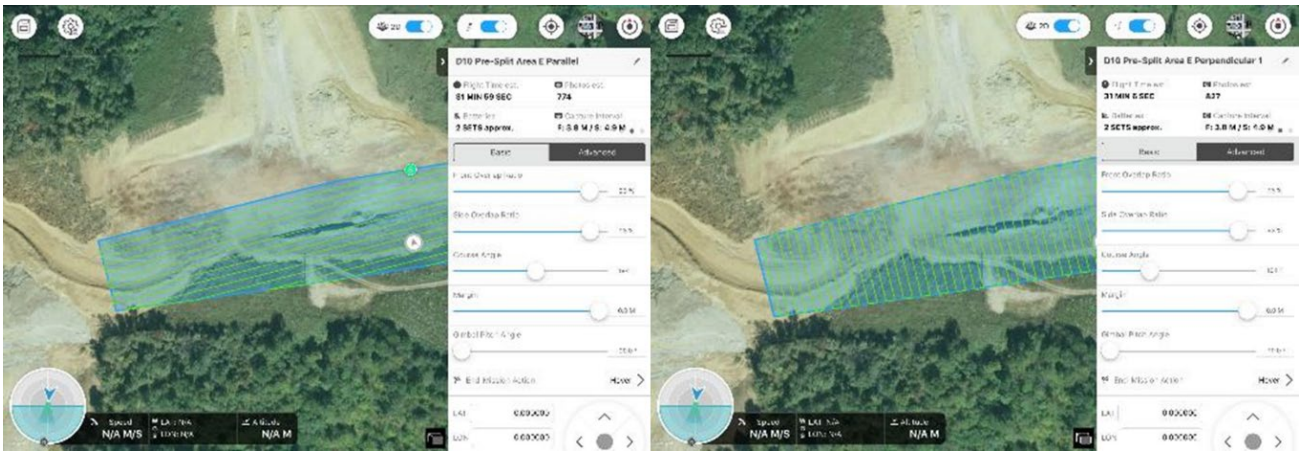


Figure 57: Screenshot of Area E Mission Plan on DJI GS Pro

The coordinates of the ground control points laid out in the field were recorded by a surveyor using survey-grade equipment. The coordinates were also recorded using the RTK system available on DJI M210 sUAS. The images captured were processed separately based on the area mapped, using Pix4D Mapper. 3D point clouds of the five areas were generated without ground control points. The areas of the pre-splits were measured using plane and surface markings, and the measured values

were compared to the values estimated using ODOT's plan sheets and the contractor's estimates. Figure 58 shows the 3D point clouds, of the respective areas, with the pre-splits marked by a surface and a plane using Pix4D Mapper.

The areas measured using the surface and plane markings are summarized in the table shown in Figure 59. The table shown in Figure 60 shows that the areas measured using the 3D point clouds lie between the estimates obtained using ODOT plan sheets and the areas estimated by the contractor, indicating the accuracy of the result obtained by UAV-based methods.

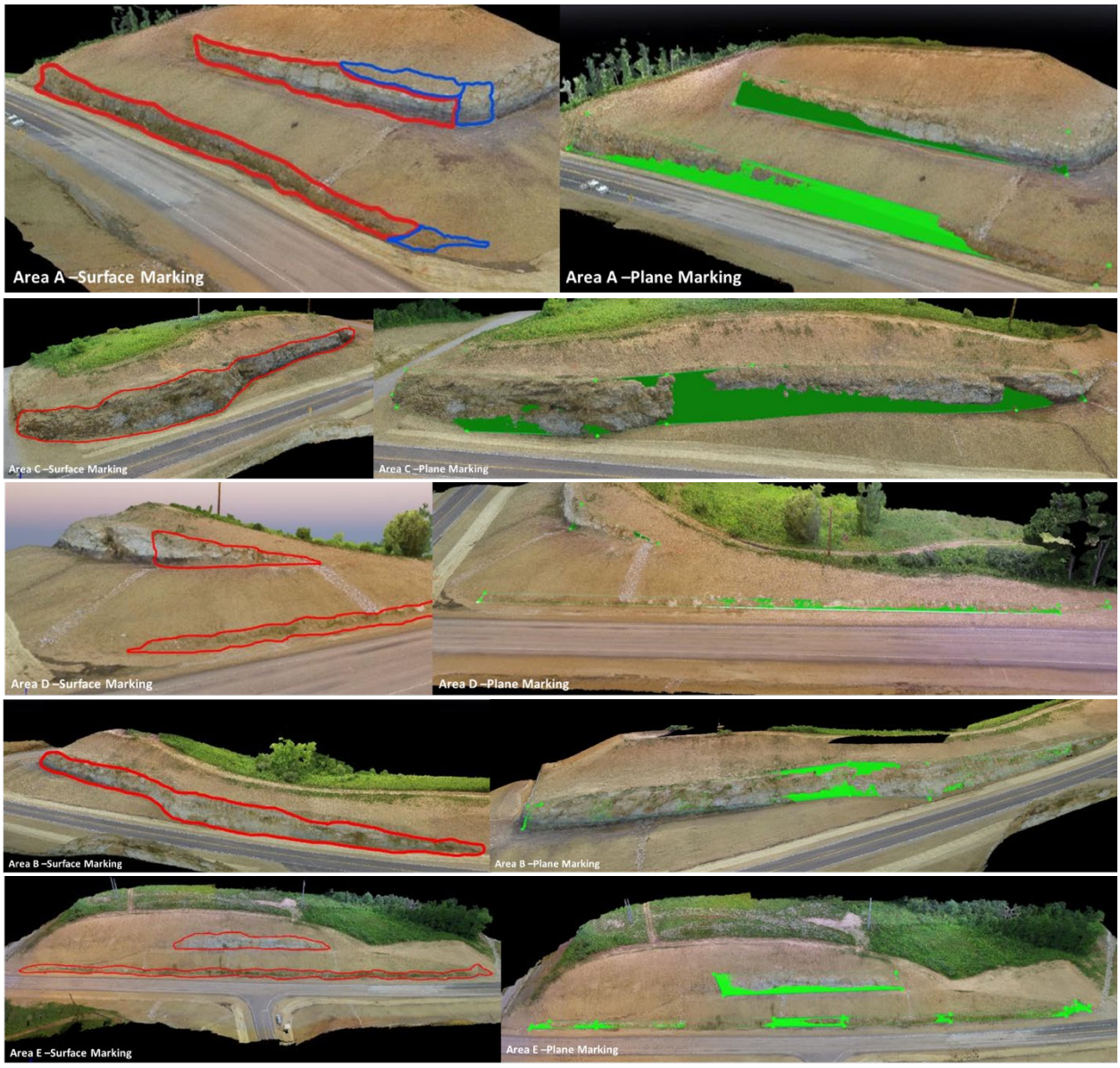


Figure 58: 3D Point Cloud of SR266 Areas with Pre-splits Marked; Left: Surface Marking, Right: Plane Marking

	w/o GCP (Surface Markings)		w/o GCP (Plane Markings)	
	sq. ft	sq. yd	sq. ft	sq. yd
Upper Cut	4,054.87	450.541	6,487.48	720.83
Lower Cut	6,124.74	680.527	9,302.27	1,033.59
Area A (STA. 53+00 to 57+00) -1	10,179.61	1,131.07	15,789.75	1,754.42
Area B (STA. 10+50 to 16+00) -1	13,061.57	1,451.28	10,088.67	1,120.96
Area C (STA. 10+50 to 16+00) -2	12,819.78	1,424.42	13,165.65	1,462.85
Upper Cut	675.78	75.09	740.73	82.3
Lower Cut	2,109.11	234.35	2,004.27	222.7
Area D (STA. 57+00 to 61+00) -1	2,784.89	309.43	2,745	305
Upper Cut	4,327.73	480.86	8,776.28	975.14
Lower Cut	7,847.81	871.98	10,244.34	1,138.26
Area E (STA. 53+00 to 61+00) -2	12,175.54	1,352.84	19,020.62	2,113.4

Figure 59: Areas Measured Using the Point Clouds Generated for the SR266 Case Study

Area as calculated by **ODOT (from project plan sheets)**	SR266 Pre-Split Total Area (Sta.53+00-62+00)	1,744 sy. yd	
	Point Lookout Road Total Area (Sta. 10+50-16+00)	3,044 sq. yd	
	Total Area	4,788 sq. yd	**Difference (surface-ODOT)**
Area as measured by **creating a surface Using 3D model w/o GCP**	SR266 Pre-Split Total Area (Sta.51+50-62+00)	2,793 sq. yd.	1049 sq.yd (+ve)
	Point Lookout Road Total Area (Sta. 10+50-16+00)	2,876 sq. yd.	168 sq. yd (-ve)
	Total Area	5,669 sq.yd.	**Difference (plane-ODOT)**
Area as measured by creating a plane **Using 3D model w/o GCP**	SR266 Pre-Split Total Area (Sta.51+50-62+00)	4,173 sq. yd.	2,429 sq.yd (+ve)
	Point Lookout Road Total Area (Sta. 10+50-16+00)	2,584 sq. yd.	460 sq. yd (-ve)
	Total Area	6,757 sq. yd.	
Area as measure by **Contractor (Approx.)**	Total Area (Sta.51+50-62+00 and Sta.10+50-16+00)	7,288 sq. yd. to 7,788 sq. yd.	

Figure 60: Comparison of Areas Measured Using 3D Point Clouds, ODOT Plan Sheets, and Contractor Estimate

I-75 Construction Site Tests, Toledo, OH: This section presents the case study conducted at a construction site located along I-75 in the Toledo, OH area within ODOT District 2. The primary objective of this study was to generate a 3D point cloud with an accuracy of 0.1' in the horizontal and vertical frame of reference. The construction site was also used to test the effect of GSDs, GCPs, sUAS, and image geotags on 3D point cloud accuracies.

The region mapped was an open field (future construction site) that lies between I-75 and a private property. The public property boundary lines were first located before the flight plans were designed. The images of the field were captured using DJI Matrice 100 and Matrice 210 RTK sUAS and DJI X5 15mm and X5s 15mm cameras. Objects (vinyl targets and metal ruler) of known measurements, were laid out in the field to test the accuracies of the outputs generated. Figure 61 shows the measurements of the objects laid out and their illustrations and images. Twelve GCPs were spray painted, using a stencil, on the ground and were distributed uniformly across the region mapped. The coordinates of the GCP centers were recorded by a surveyor using survey-grade equipment. The coordinates were referenced in the NAD83 2011 Ohio north horizontal coordinate system and in the NAVD88 Geoid 12A vertical coordinate system. Figure 62 shows the location of the GCPs on the region mapped and the boundary line that indicates the area accessible to the public. Figure 63 shows the images of the twelve GCPs marked on the field.

On-field Measurements

Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Name	S1 - S2	S2-S3	S3-S4	S4-S1	Long	Small	M1 - M2	M2-M3	M3-M4	M4-M1	M2 1-M2 2	M2 2-M2 3	M2 3-M2 4	M2 4-M2 1	L1-L2	L2-L3	L3-L4	L4-L1
Value (ft)	2.58	2.6	2.63	2.58	2	1.33	9.67	7.83	9.88	7.67	7.75	9.65	7.71	9.71	12.1	19.79	12.17	19.81
Object	Small Target				Metal Ruler		Medium Target 1				Medium Target 2				Large Target			

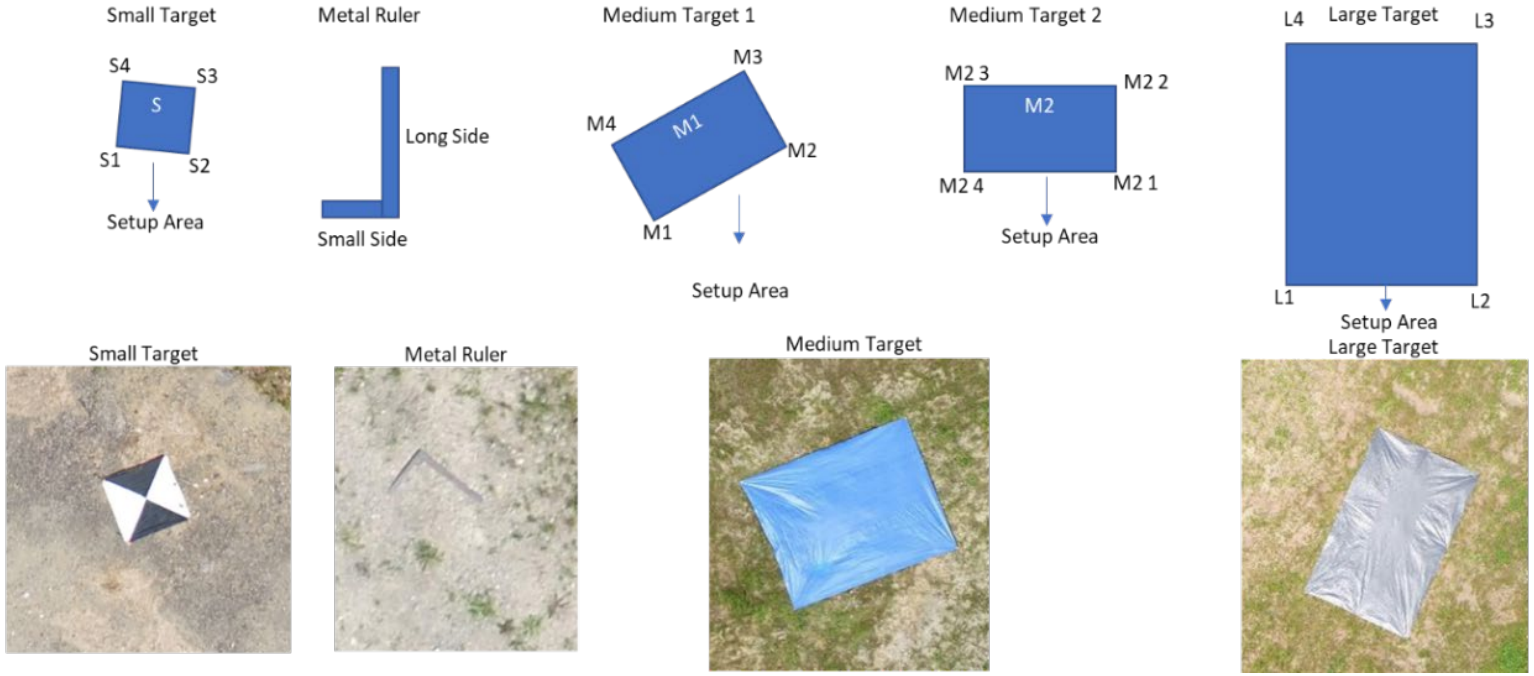


Figure 61: Measurement of Objects Laid Out in the Region Mapped for the i75 Construction Site

Figure 62: Images of the GCPs Marked on the Field for the I-75 Construction Site

Figure 63: Locations of the GCPs Marked on the Region Mapped for the I-75 Construction Site

		Mission 1	Mission 2	Mission 3
sUAS		Matrice 100	Matrice 100	Matrice 210 RTK
Camera		X5 15mm	X5 15mm	X5S 15mm
	Resolution	16MP	16MP	21MP
	Resolution (WxH)	4608px x 3456px	4608px x 3456px	5280px x 3956px
Flight Type		Grid Pattern	Grid Pattern	Grid Pattern
Ground Sampling Distance (GSD)		1cm/px	2cm/px	1cm/px
Flight Height (m)		38m (125ft)	80m (320ft)	45m
Total Flight Time		50min	15min	40min
Total Number of Batteries		4	2	4
Total Number of Photos		1280	337	930
Total Number of GCPs		12	12	12

Figure 64: Summary of the Missions Conducted for the I-75 Construction Site

Images were captured in a grid pattern, with 1cm/px and 2cm/px GSDs and with 75 % image overlaps, using the Matrice 100 sUAS and DJI X5 15mm camera. Images were also captured in a grid pattern, with 1cm/px GSD and 75% image overlap, using the Matrice 210 RTK sUAS and DJI X5s 15mm camera. Figure 64 shows a summary of all the missions conducted at the site. Figure 65 shows the locations of all the images captured for the study.

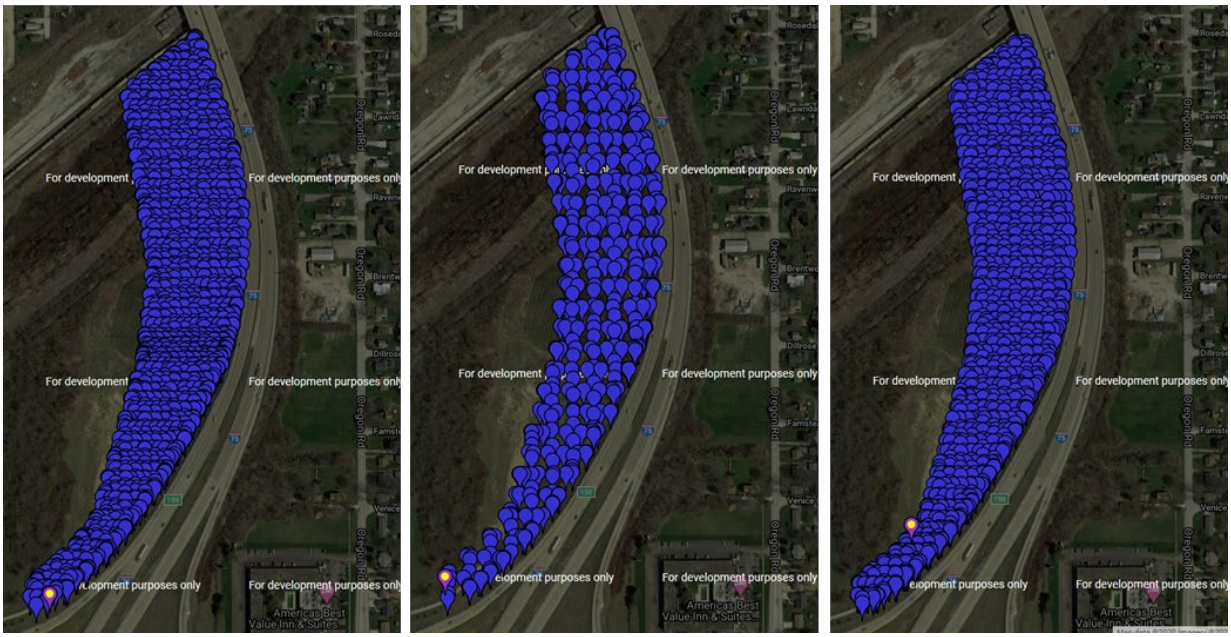
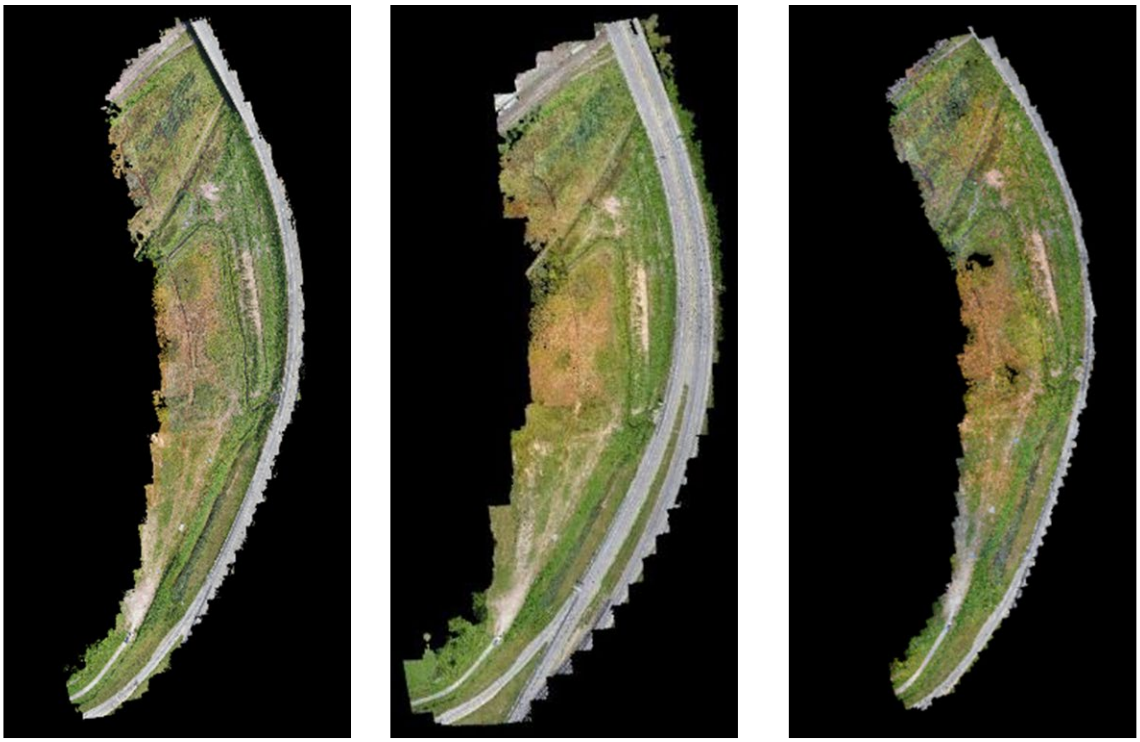


Figure 65: Location of the Images Captured for the I-75 Construction Site; Left: 1cm/px Dataset Using X5, center: 2cm/px Dataset Using X5, Right: 1cm/px Dataset Using X5s

The images captured at the I-75 construction site were processed using Pix4D Mapper in multiple ways using the default 3D model template. The images captured in each mission were processed separately without GCPs and with image geotags, with GCPs and with image geotags, and finally with GCPs and without image geotags. The measurements of the objects laid out in the field were measured using the point clouds generated in this case study. These measurements were analyzed separately to understand the effect of varying GSDs, GCPs, various sUAS and image geotags on 3D point cloud accuracies. Figure 66 shows the point clouds generated without GCPs using the combined (grid) dataset from each mission.



M100 & X5 (1cm/px GSD) M100 & X5 (2cm/px GSD) M210 & X5s (1cm/px GSD)

Figure 66: 3D Point Clouds Generated Using the Dataset Captured for the I-75 Construction Site; Left: 3D Point Cloud Processed Using X5 1cm/px Dataset, Center: 3D Point Cloud Processed Using X5 2cm/px Dataset, Right: 3D Point Cloud Processed Using X5s

The generated 3D point clouds were analyzed by measuring the objects laid out in the field using the polyline line tool and the computed values were compared to the values recorded on the field. The plots in Figure 67 show the individual absolute error measurements, and their mean absolute and root mean square errors of the 3D point clouds generated using the 1cm/px data, captured using the DJI Matrice 100 sUAS and DJI X5 15mm camera, without GCPs and with image geotags, with GCPs and images geotags and with GCPs and without image geotags.

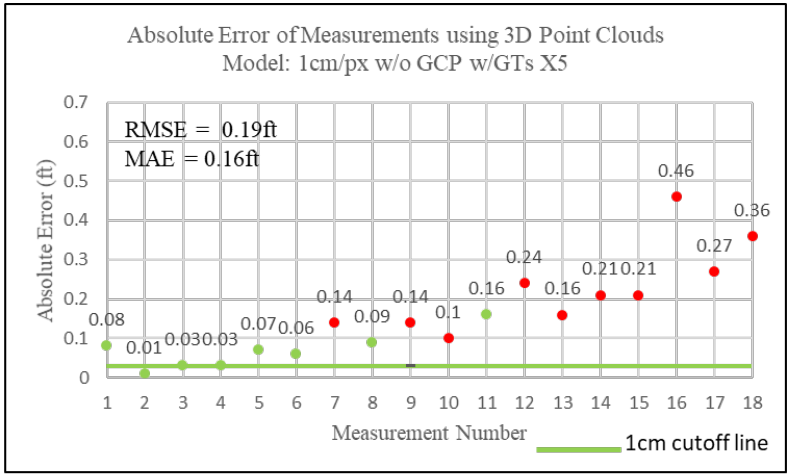
The plots in Figure 68 show the individual absolute error measurements, and their mean absolute and root mean square errors of the 3D point clouds generated using the 2cm/px data, captured using the DJI Matrice 100 sUAS and DJI X5 15mm camera, without GCPs and with image geotags, with GCPs and images geotags and with GCPs and without image geotags.

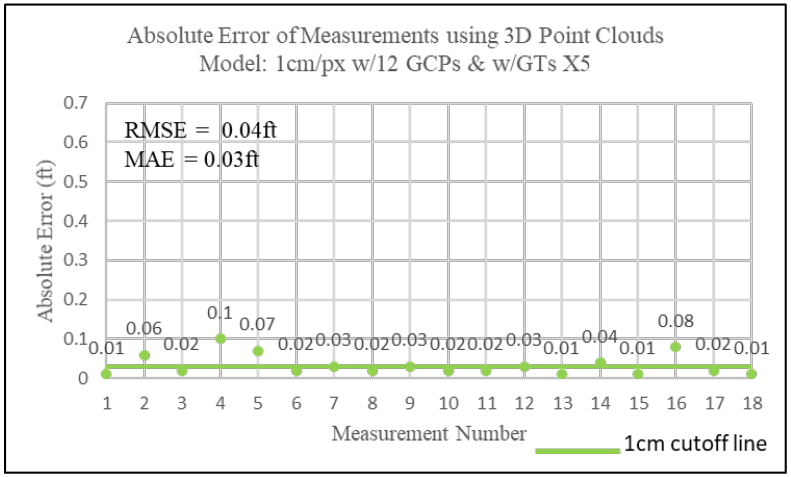
The plots in Figure 69 show the individual absolute error measurements, and their mean absolute and root mean square errors of the 3D point clouds generated using the 1cm/px data, captured using the DJI Matrice 210 RTK sUAS and DJI X5s 15mm camera, without GCPs and with image geotags, with GCPs and images geotags and with GCPs and without image geotags.

To study the effect of varying GSDs on the accuracy of 3D point clouds, the absolute error measurement plots in Figures 68 and 69 were compared. Comparing

the results compiled for the models generated without GCPs and with image geotags, one can clearly see that the mean and RMSE errors increase as the GSD value increases. The effect of using GCPs to calibrate 3D point clouds can be seen in Figure 70. The plots of the models generated with GCPs and image geotags clearly show smaller error values when compared to the plots for the models generated without GCPs. It can also be seen that the errors recorded using the model generated with the 1cm/px dataset and GCPs have errors within the ranges acceptable by ODOT surveyors [30]. Given the small error values recorded using the model generated using the 1cm/px dataset with GCPs and image geotags, the point cloud was edited to remove tall vegetations and noise.

The effect of image geotags can be since by comparing the error plots generated for the models with and without image geotags and with GCPs. The effect of image geotags can be seen clearly in the model generated using the 2cm/px dataset captured using the Matrice 100 sUAS and X5 camera. Removing the geotags helped improve the accuracy of the produced 3D point cloud. Whereas, removing the geotags from the images captured at a GSD of 1cm/px did not impact the calculated errors. The dataset captured using the DJI Matrice 210 RTK helps improve the accuracy of the generated 3D point cloud by a small factor. This is seen when the plots (plot of 3D point cloud generated using image geotags and without GCPs) in Figures 67 and 69 are compared.





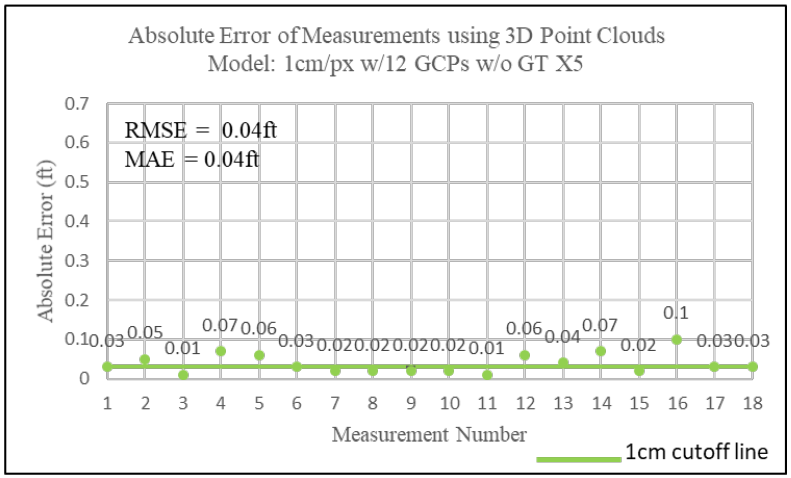
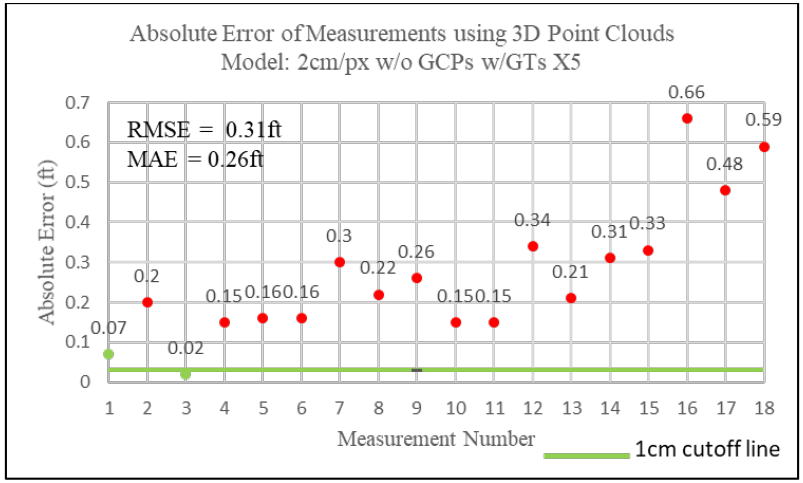
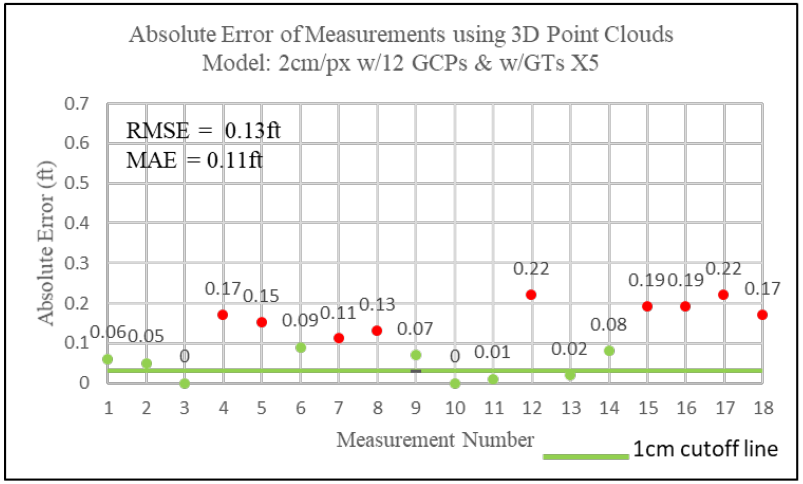


Figure 67: Absolute Error Measurement Plots of the 3D Point Clouds Generated Using the 1cm/px GSD Dataset Obtained Using DJI Matrice 100 and DJI X5 15mm; Top: 3D Point Cloud Generated Without GCPs and With Image Geotags, Bottom Left: 3D Point Cloud Generated With GCPs and with Image Geotags, Bottom Right: 3D Point Cloud Generated Without GCPs and with Image Geotags





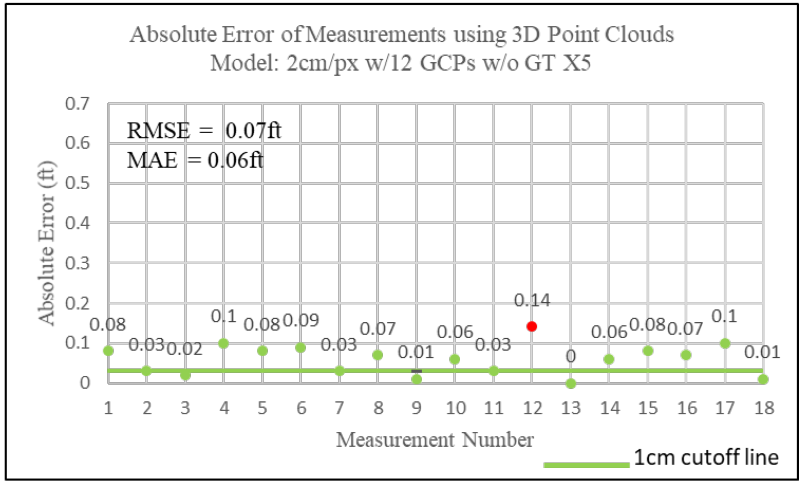
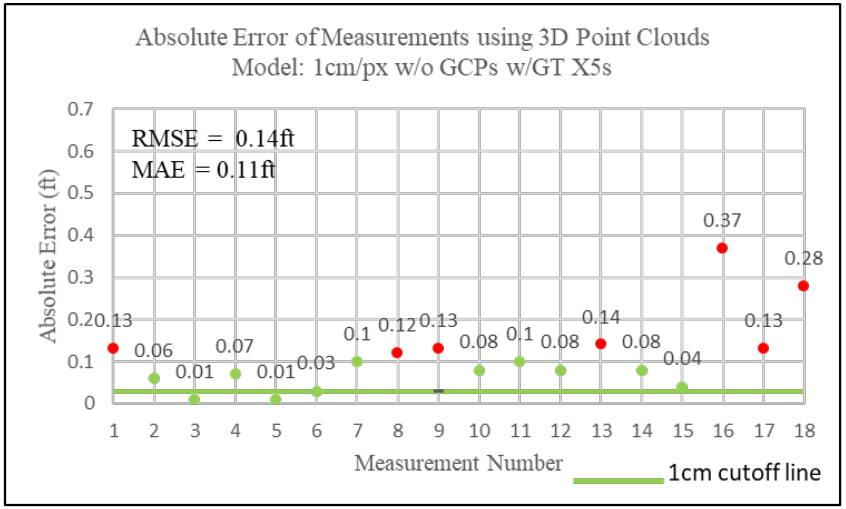
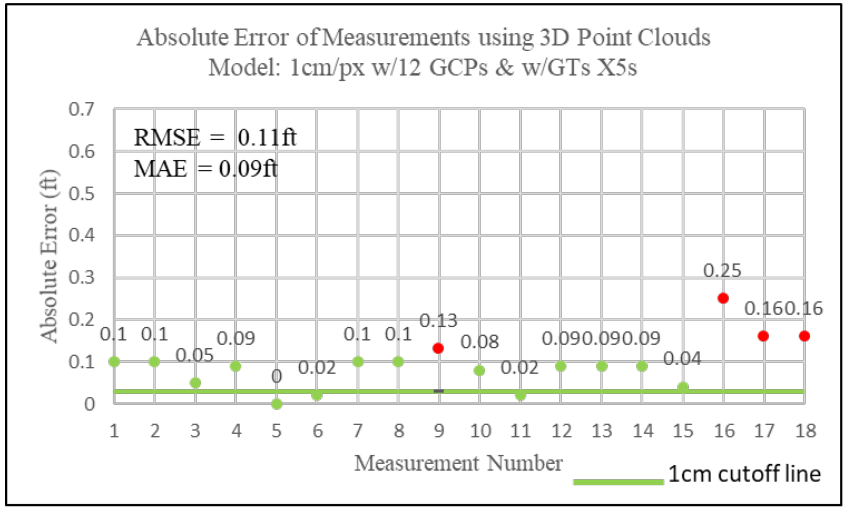


Figure 68: Absolute Error Measurement Plots of the 3D Point Clouds Generated Using the 2cm/px GSD Dataset Obtained Using DJI Matrice 100 and DJI X5 15mm; Top: 3D Point Cloud Generated Without GCPs and With Image Geotags, Bottom Left: 3D Point Cloud Generated With GCPs and with Image Geotags, Bottom Right: 3D Point Cloud Generated Without GCPs and with Image Geotags





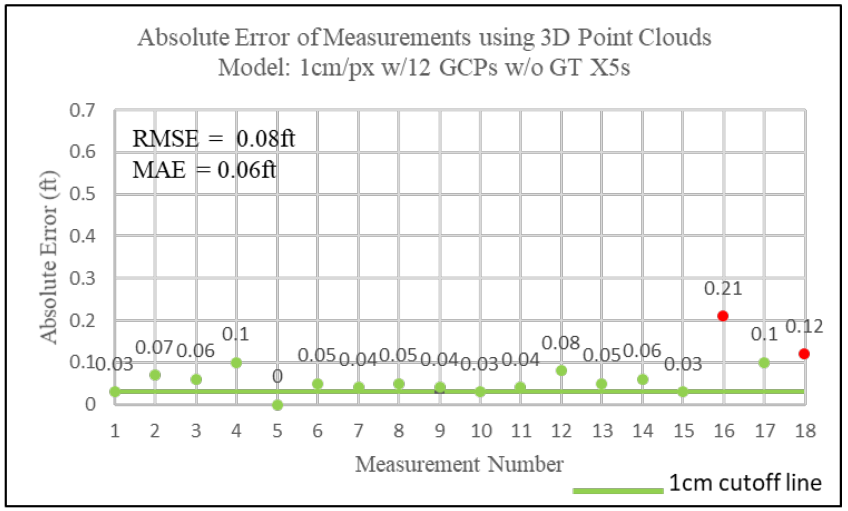


Figure 69: Absolute Error Measurement Plots of the 3D Point Clouds Generated Using the 1cm/px GSD Dataset Obtained Using DJI Matrice 210 and DJI X5s 15mm; Top: 3D Point Cloud Generated Without GCPs and With Image Geotags, Bottom Left: 3D Point Cloud Generated With GCPs and with Image Geotags, Bottom Right: 3D Point Cloud Generated Without GCPs and with Image Geotags

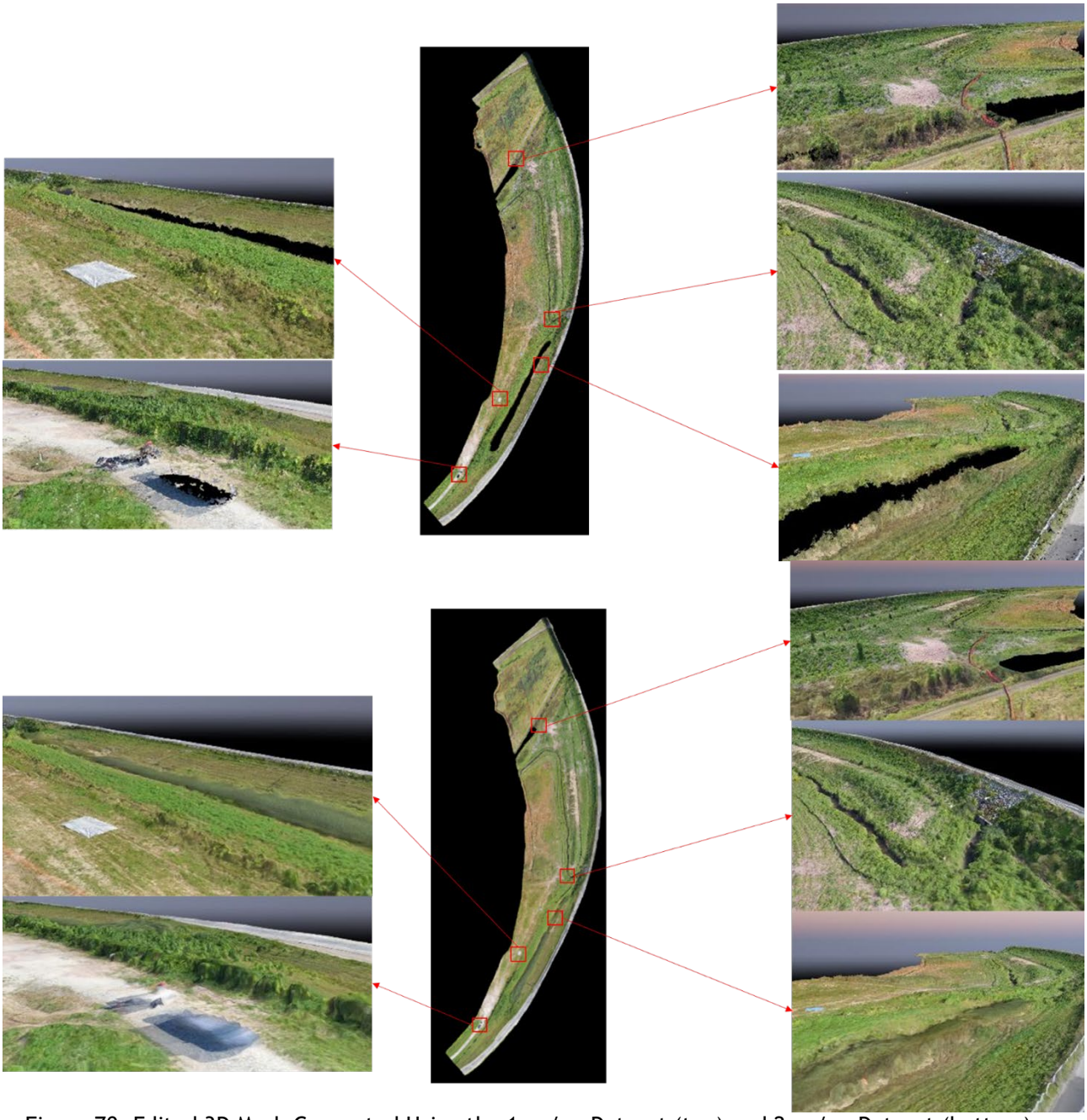


Figure 70: Edited 3D Mesh Generated Using the 1cm/px Dataset (top) and 2cm/px Dataset (bottom) with GCPs Captured for the I-75 Construction Site

Deer Creek Park Tests, Deer Creek, OH: This section presents the case study conducted at Deer Creek State Park in Deer Creek, OH. The primary objective of this study was to record the accuracies of the 3D point clouds generated using images captured with DJI Phantom 4 RTK sUAS. The sUAS has the capability to connect to Ohio's VRS system which helps improve the positioning of the vehicle in midair. The study documents the effect of a varying number of GCPs, GSDs, and pattern of image

capture on the accuracies of the 3D point clouds generated using the DJI Phantom 4 RTK dataset.

Deer Creek Park is a test site frequently used by ODOT surveyors to test their survey grade equipment. From their studies, certain regions within the park were identified to produce survey results of good quality. One such region is marked in yellow in the map shown in Figure 71. A region that lies inside the yellow boundary was selected to conduct the study (Figure 72). The selected region was divided into a grid of thirty cells to determine the location of vinyl targets, whose centers were used as GCPs and checkpoints. Figure 73 shows the proposed positions of the vinyl targets in the region mapped. Fifteen of the vinyl targets were selected to be used as GCPs and the remaining fifteen were used as checkpoints (Figure 74).

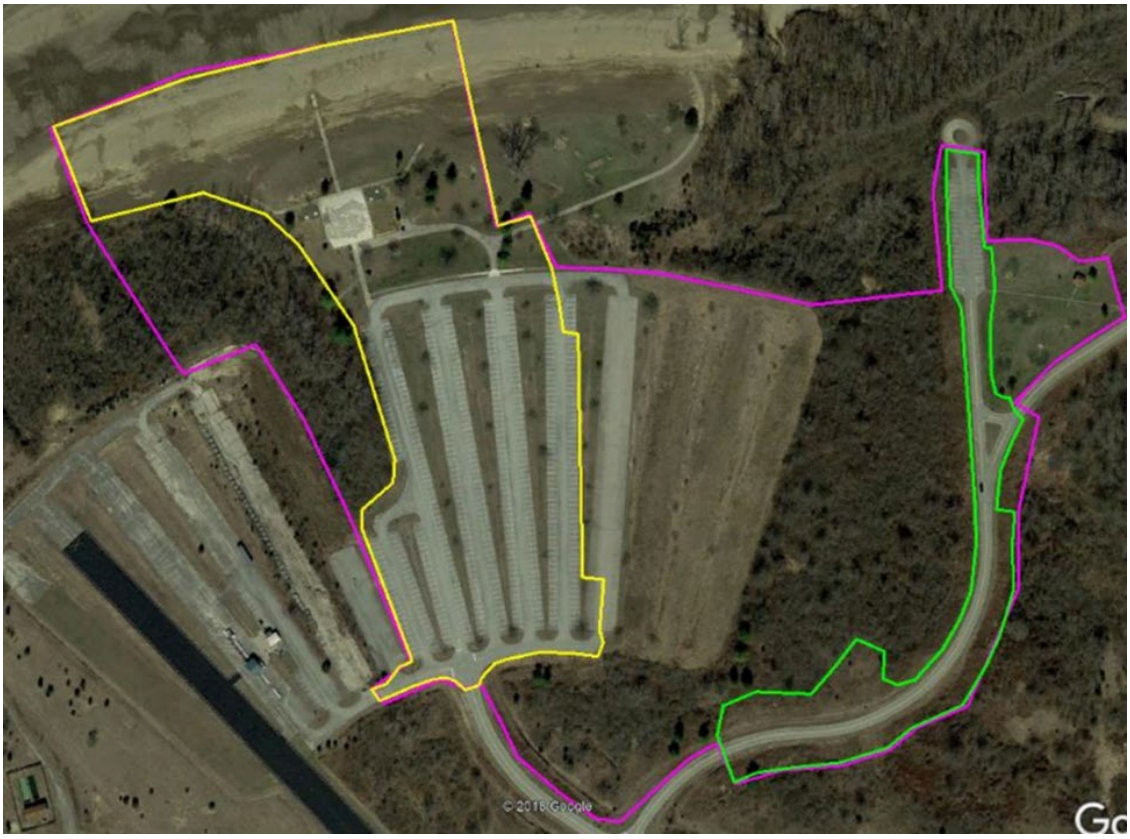


Figure 71: Deer Creek Park Regions Used by ODOT Surveyors to Test Their Surveying Equipment

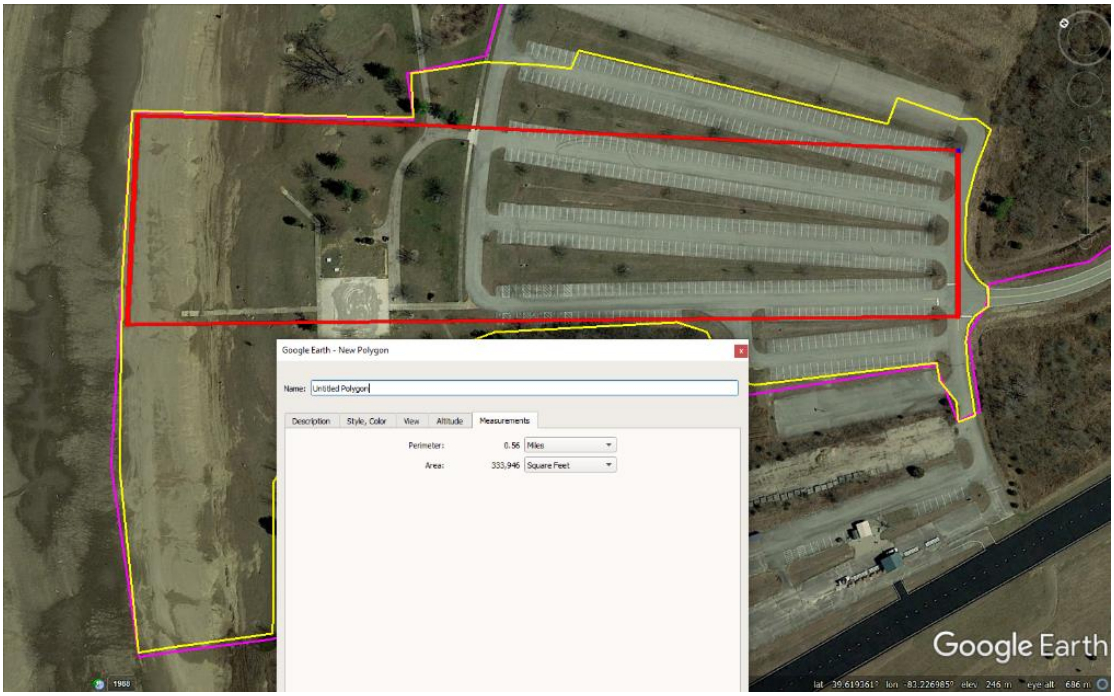


Figure 72: Region Selected Within the Yellow Boundary Line to Conduct the Study

Figure 73: Proposed Positions of the Vinyl Targets in the Mapped Region

Figure 74: Map of the Region Mapped with Vinyl Targets Designated as GCPs and Check Points

The positions of the targets that were going to be used as GCPs, were selected following Pix4D recommendations [30]. The selected GCPs were distributed uniformly on the mapped region. The coordinates of the target centers were recorded by an ODOT surveyor using survey-grade equipment. The coordinates were recorded in the NAD83 2011 Ohio South horizontal coordinate system and in the NAVD88 Geoid 12A vertical coordinate system. The coordinates were also recorded in the ellipsoid reference system (WGS84). Figure 75 shows the actual positions of the targets laid out and the targets selected as GCPs and checkpoints. Figure 76 shows the table with the coordinates of the target centers. On the day of the test thirty-one targets were laid out and analyzing the image dataset it was observed that target AT22 was not secured properly and was ignored for this study.

Figure 75: Actual Positions of the Vinyl Targets and Targets Selected as GCPs and Check Points

Point Name	Northing (ft)	Easting (ft)	Elevation (ft)		Latitude (degree)	Longitude (degree)	Ellipsoid Ht. (m)
AT01	589477.358	1764235.313	853.594		39d36'58.39647"	-83d13'30.35252"	745.721
AT02	589629.3	1764180.219	850.499		39d36'59.89377"	-83d13'31.07215"	742.629
AT03	589706.907	1764146.42	848.727		39d37'00.65810"	-83d13'31.51203"	740.858
AT04	589769.573	1764115.355	847.343		39d37'01.27497"	-83d13'31.91544"	739.476
AT05	589532.287	1764152.46	851.552		39d36'58.93276"	-83d13'31.41692"	743.683
AT06	589597.591	1764112.308	849.91		39d36'59.57499"	-83d13'31.93672"	742.043
AT07	589679.243	1764079.857	847.674		39d37'00.37940"	-83d13'32.35978"	739.808
AT08	589707.423	1764047.227	847.742		39d37'00.65532"	-83d13'32.77965"	739.878
AT09	589827.365	1764013.004	845.279		39d37'01.83801"	-83d13'33.22930"	737.417
AT10	589877.8	1763950.375	843.073		39d37'02.33149"	-83d13'34.03481"	735.213
AT11	589978.856	1763922.651	840.272		39d37'03.32805"	-83d13'34.39948"	732.413
AT12	590040.422	1763927.019	842.88		39d37'03.93687"	-83d13'34.34999"	735.022
AT13	590045.194	1763774.171	839.578		39d37'03.97187"	-83d13'36.30373"	731.727
AT14	590103.913	1763876.155	839.126		39d37'04.56032"	-83d13'35.00651"	731.27
AT15	590011.368	1763708.931	840.446		39d37'03.63238"	-83d13'37.13395"	732.598
AT16	589409.929	1764127.614	854.046		39d36'57.72150"	-83d13'31.72186"	746.178
AT17	589532.044	1764057.364	851.264		39d36'58.92281"	-83d13'32.63210"	743.399
AT18	589608.662	1764004.608	849.498		39d36'59.67585"	-83d13'33.31413"	741.636
AT19	589738.785	1763911.762	846.627		39d37'00.95451"	-83d13'34.51396"	738.769
AT20	589825.346	1763896.941	844.367		39d37'01.80884"	-83d13'34.71225"	736.509
AT21	589907.895	1763796.775	842.896		39d37'02.61672"	-83d13'36.00075"	735.043
AT22	590120.126	1763613.711	841.204		39d37'04.69968"	-83d13'38.36196"	733.36
AT23	590147.851	1763752.273	839.963		39d37'04.98471"	-83d13'36.59412"	732.113
AT24	590262.55	1763817.609	827.064		39d37'06.12351"	-83d13'35.77099"	719.21
AT25	590389.017	1763748.78	819.338		39d37'07.36794"	-83d13'36.66356"	711.488
AT26	590482.688	1763650.295	810.828		39d37'08.28587"	-83d13'37.93177"	702.983
AT27	590308.994	1763658.912	823.242		39d37'06.56990"	-83d13'37.80377"	715.396
AT28	590246.549	1763547.378	824.648		39d37'05.94387"	-83d13'39.22265"	716.807
AT29	590368.789	1763472.601	810.854		39d37'07.14603"	-83d13'40.19082"	703.016
AT30	590414.085	1763589.275	812.314		39d37'07.60299"	-83d13'38.70449"	704.471
AT31	589825.577	1764088.815	846.132		39d37'01.82636"	-83d13'32.26034"	738.266

Figure 76: Coordinates of the Vinyl Targets Placed in the Region of Study

Images of the region were captured using the DJI Phantom 4 RTK sUAS connected to the Ohio VRS system. Four missions were conducted where the first three missions captured images in a grid pattern with 80% image overlap and GSDs of 0.75cm/px, 1cm/px and 1.25cm/px. The fourth mission was a terrain awareness mission where the Phantom 4 sUAS changed its flight altitude with the terrain to obtain images with fixed GSD and image overlap values. To conduct the terrain awareness mission the digital terrain model of the area mapped was uploaded to the sUAS. The sUAS corrects its altitude stepwise at each pass if there is a terrain change. Figure 77 shows the location of the images captured for the Deer Creek study. Figure 78 shows the summary of all the missions conducted at Deer Creek Park.

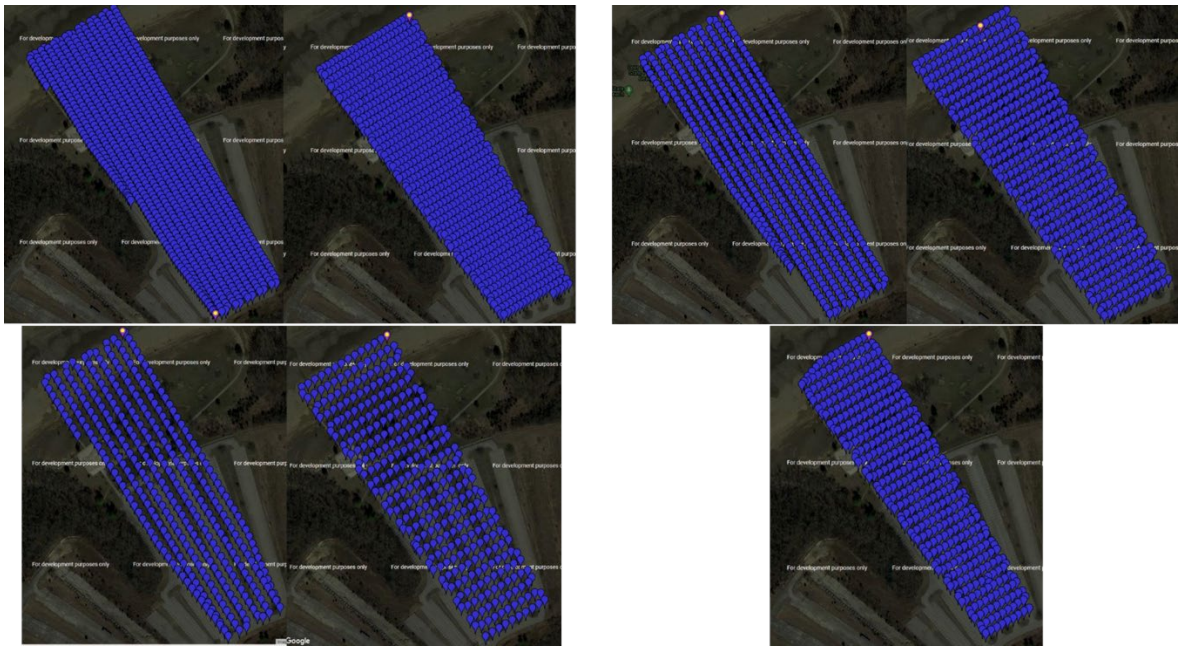


Figure 77: Locations of Images Captured for Deer Creek Study; Top Left: 0.75cm/px Dataset, Top Right: 1cm/px Dataset, Bottom Left: 2cm/px Dataset, Bottom Right: 1cm/px Terrain Awareness Dataset

		Mission 1	Mission 2	Mission 3	Mission 4
sUAS		DJI Phantom 4 RTK	DJI Phantom 4 RTK	DJI Phantom 4 RTK	DJI Phantom 4 RTK
Camera		Phantom 4	Phantom 4	Phantom 4	Phantom 4
	Resolution	20MP	20MP	20MP	20MP
	Resolution (WxH)	5472px x 3648px	5472px x 3648px	5472px x 3648px	5472px x 3648px
Flight Type		Grid Pattern	Grid Pattern	Grid Pattern	Perpendicular Terrain Awareness
Ground Sampling Distance (GSD)		0.75cm/px	1cm/px	1.25cm/px	1cm/px
Flight Height (m)		27m (88.6ft)	36m (118.1ft)	46m (151ft)	28m to 40m (91.9ft to 131.2ft)
FO/SO (%)		80%/80%	80%/80%	80%/80%	80%/80%
Total Number of Photos		1910	1068	691	435

Figure 78: Summary of the Missions Conducted for the Deer Creek Park Case Study

Figure 79: 3D Point Cloud Generated Using the 1cm/px Grid Dataset of the Region Mapped in Deer Creek Park

The images obtained in each mission were processed separately using Pix4D Mapper and the 3D point clouds generated were analyzed separately by comparing the computed point cloud coordinates of the checkpoint target centers with the coordinates recorded by the ODOT surveyor. The error values in pixels were also calculated by dividing the measured errors by the average GSD of the 3D point cloud. The point cloud was referenced in the NAD83 2011 Ohio south horizontal coordinate reference system and the vertical coordinates system was set to arbitrary on Pix4D Mapper. When the vertical coordinate system is set to arbitrary and the model is processed with GCPs, Pix4D Mapper uses the GCP's altitude to reference the model's altitude. When the vertical coordinate system is set to arbitrary and the model is processed without GCPs, Pix4D Mapper uses the image's vertical coordinate system to reference the model's altitude. The 0.75cm/px dataset (parallel and perpendicular) were processed together and separately without GCPs and with 14 GCPs. The 1cm/px dataset (parallel and perpendicular) were processed together and separately without GCPs and with 14, 12, 11, 9, 7, 5, and 3 GCPs. The 1cm/px terrain awareness dataset was processed without GCPs and with 14 GCPs. The 1.25cm/px dataset (parallel and perpendicular) were processed together and separately without GCPs and with 14 GCPs. Figure 79 shows the 3D point cloud generated using the 1cm/px grid dataset and 14 GCPs. Figures 80 to 85 show the map of the targets used as GCPs for the 12, 11, 9, 7, 5, and 3 GCP models respectively.

Figure 80: Positions of the Targets as GCPs for the 12 GCPs 3D point Cloud

Figure 81: Positions of the Targets as GCPs for the 11 GCPs 3D point Cloud

Figure 82: Positions of the Targets as GCPs for the 9 GCPs 3D point Cloud

Figure 83: Positions of the Targets as GCPs for the 7 GCPs 3D point Cloud

Figure 84: Positions of the Targets as GCPs for the 5 GCPs 3D point Cloud

Figure 85: Positions of the Targets as GCPs for the 3 GCPs 3D point Cloud

The RMSE values obtained for each of the models were compared to understand the effect of the number of GCPs, GSD values, and pattern of flight on the accuracies of 3D point clouds. Analyzing the error values obtained for each checkpoint in the models processed it was observed that checkpoint AT20 consistently had higher error values and was thus not included to calculate the RMSE values for the models. Figure 86 shows the RMSE plots of the easting, northing and altitude values for the 3D point clouds generated using a 1cm/px grid dataset with 0, 3, 5, 7, 9, 11, 12, and 14 GCPs. The plots in Figure 86 shows, an accurate 3D point cloud, with errors close to 0.02ft, can be generated with a minimum of 3 GCPs (distributed uniformly) using images captured with the Phantom 4 RTK sUAS connected to the VRS.

Figure 87 shows the RMSE plots of the easting, northing and altitude values for the 3D point clouds generated using 0.75cm/px, 1cm/px, and 1.25cm/px grid dataset without GCPs. The plots in Figure 87 show that similar error values in easting, northing and altitude were obtained using the three different GSD datasets. The similar errors observed could be due to the very similar GSD values of the images used to generate the 3D point clouds. It was also observed that the error values in the northing direction were close to 0.02 ft without any GCP calibration.

Figure 88 shows the RMSE plots of the easting, northing and altitude values for the 3D point clouds generated using 0.75cm/px, 1cm/px, and 1.25cm/px grid dataset with 14 GCPs. The plots in Figure 88 show that adding 14 GCPs helped in reducing the errors seen in the plots in Figure 107. The error values in the easting, northing, and altitude directions were close to 0.02 ft.

Figure 89 shows the RMSE plots of the easting, northing and altitude values for the 3D point clouds generated using 0.75cm/px, 1cm/px, and 1.25cm/px grid, parallel, and perpendicular dataset without GCPs. The plots in Figure 89 show that for most of the cases generating 3D point clouds using the combined dataset generates similar or smaller error values when compared to errors computed using the models generated separately (parallel and perpendicular dataset). It was also observed that the error values in the northing direction were still close to 0.02ft for the models processed using the combined datasets and without any GCP calibration.

Figure 90 shows the RMSE plots of the easting, northing and altitude values for the 3D point clouds generated using 0.75cm/px, 1cm/px, and 1.25cm/px grid, parallel, and perpendicular dataset with 14 GCPs. The plots in Figure 90 show that adding 14 GCPs helped in reducing the errors irrespective of the dataset used. Calibrating the models with 14 GCPs bought the errors close 0.02ft. The pattern of image capture did not have any impact on the errors calculated for the easting and northing values but helped reduce the errors calculated for the altitude values.

Figure 91 shows the RMSE plots of the easting, northing and altitude values for the 3D point clouds generated using 1cm/px perpendicular and 1cm/px terrain awareness datasets with 14 GCPs and without GCPs. The plots in Figure 91 show that the 3D point cloud generated, without GCPs, using the 1cm/px terrain awareness data has relatively better accuracies in the northing and altitude directions compared to

the 3D point cloud generated using the 1cm/px non-terrain awareness dataset. However, these errors reduce to values close 0.02ft when the models are calibrated using 14 GCPs.

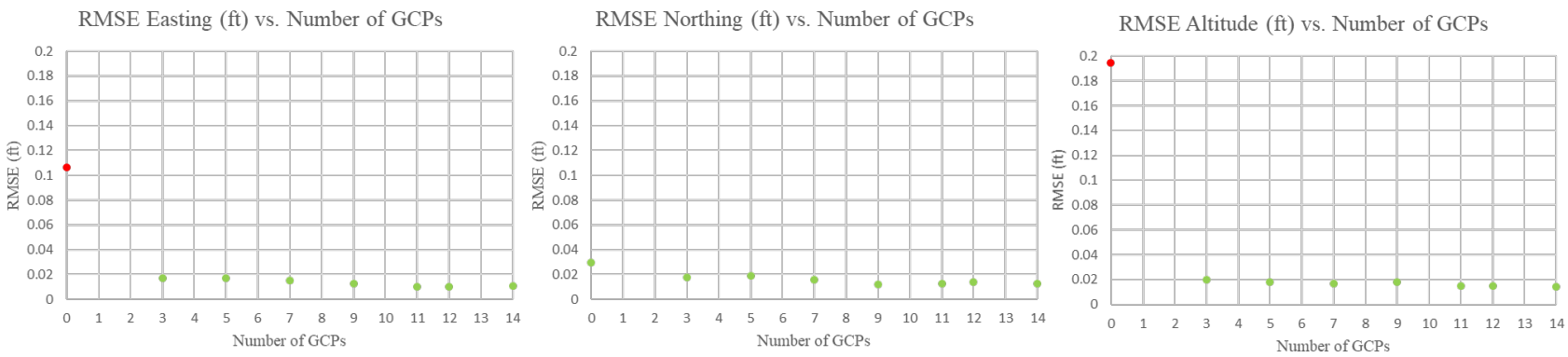


Figure 86: RMSE Plots of Easting, Northing, and Altitude of the 3D Point Clouds Generated Using 1cm/px Grid Dataset with 0, 3, 5, 7, 9, 11, 12, and 14 GCPs

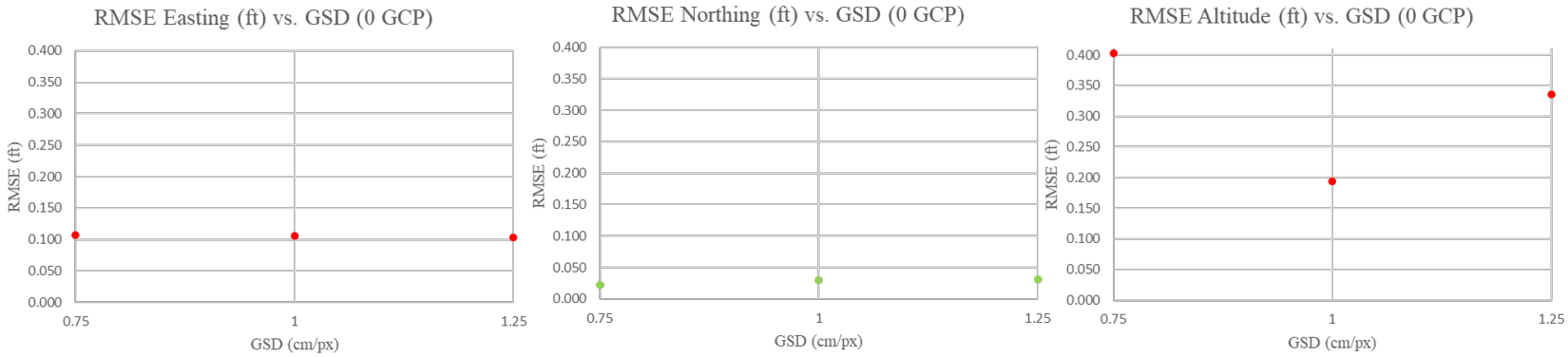


Figure 87: RMSE Plots of Easting, Northing, and Altitude of the 3D Point Clouds Generated Using 0.75cm/px, 1cm/px, and 1.25cm/px Grid Dataset Without GCPs

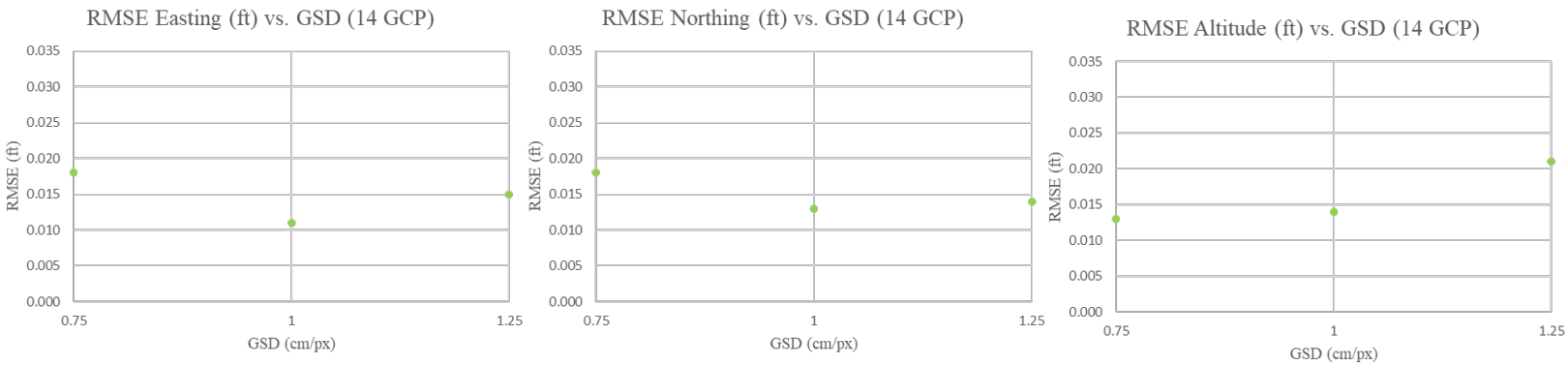


Figure 88: RMSE Plots of Easting, Northing, and Altitude of the 3D Point Clouds Generated Using 0.75cm/px, 1cm/px, and 1.25cm/px Grid Dataset 14 GCPs

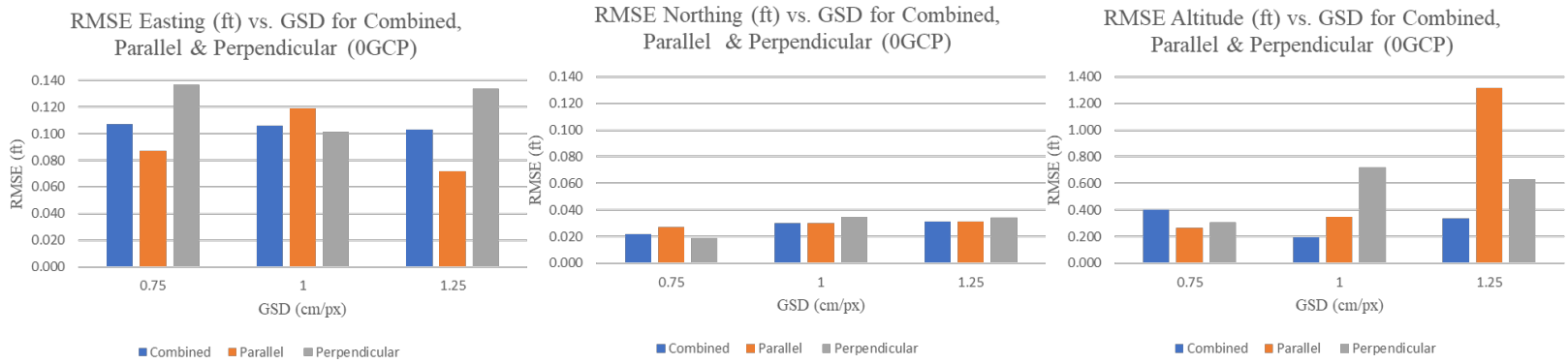


Figure 89: RMSE Plots of Easting, Northing, and Altitude of the 3D Point Clouds Generated Using 0.75cm/px, 1cm/px and 1.25cm/px Grid, Parallel and Perpendicular Datasets Without GCPs

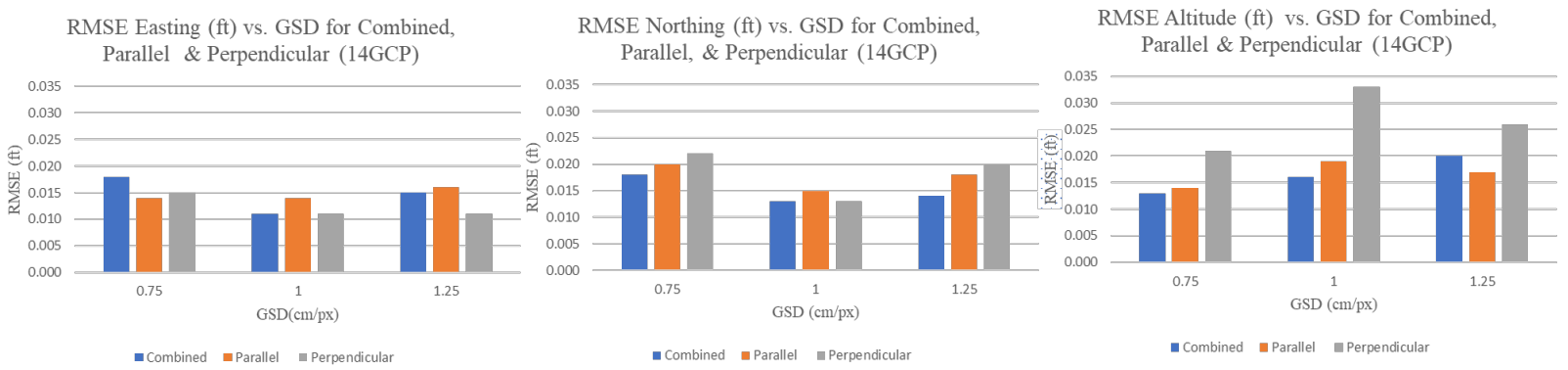


Figure 90: RMSE Plots of Easting, Northing, and Altitude of the 3D Point Clouds Generated Using 0.75cm/px, 1cm/px and 1.25cm/px Grid, Parallel and Perpendicular Datasets 14 GCPs

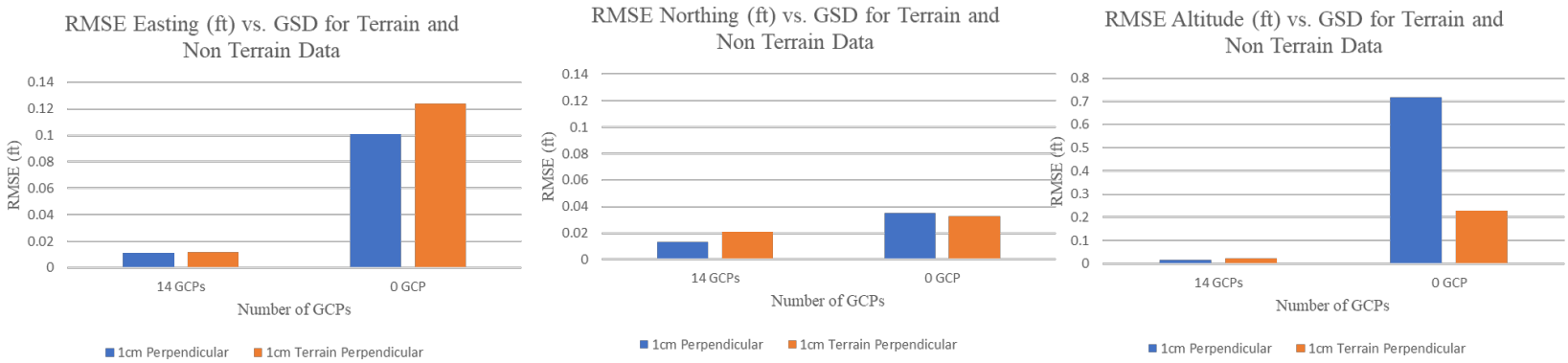


Figure 91: RMSE Plots of Easting, Northing, and Altitude of the 3D Point Clouds Generated Using 1cm/px Perpendicular and 1cm/px Terrain Awareness Datasets With 14 GCPS and Without GCPs

SR72 Highway Mapping, Cedarville, OH: This section presents the case study conducted at state SR72 in Cedarville, OH, ODOT District 6. The primary objective of this study was to record the accuracies of the 3D point clouds generated using images captured with DJI Phantom 4 RTK sUAS and the effect of using ground control points. The sUAS can connect to Ohio's VRS system which helps improve the positioning of the vehicle in midair. The study documents the effect of the quality of GCPs, laid out, on the accuracies of the 3D point clouds generated using the DJI Phantom 4 RTK dataset.

The SR72 that runs through Cedarville, Ohio was mapped to build its 3D point cloud and help ODOT personnel survey the state route. Mag-nails with small strips of reflective tape were drilled into the road along the shoulder of the highway. The mag-nails laid out were used as ground control points and control/checkpoints to calibrate and check the accuracy of the 3D point clouds respectively. The mag-nails were drilled in a zig-zag pattern along the shoulder of the highway. Images of a five-mile stretch of the state route were captured as shown in Figure 92.

Ninety-three mag-nails were drilled, and their locations were recorded for this case study. The coordinates of the target centers were recorded by an ODOT surveyor using survey-grade equipment. The coordinates were recorded in the NAD83 2011 Ohio South horizontal coordinate system and the NAVD88 Geoid 12A vertical coordinate system. Figures 93 and 94 show the actual positions of the mag-nails laid out and their zig-zag pattern.

Figure 92: Locations of Images Captured for the SR72 Highway Mapping Case Study

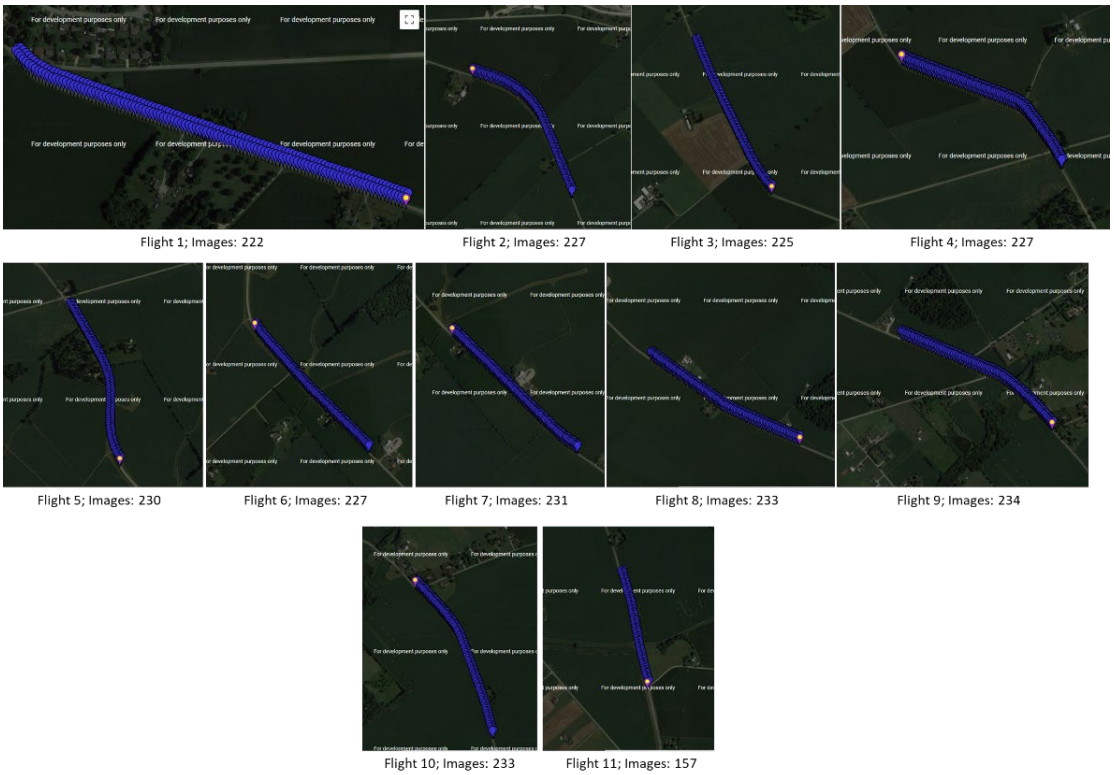


Figure 93: Location of Images Captured for each of the Eleven Flight Missions Conducted for the SR72 Case Study

Figure 94: Positions of the Mag-nails used as GCPs and Check Points

Images of the five-mile stretch of road were captured using the DJI Phantom 4 RTK sUAS connected to the Ohio VRS system. Eleven different flights were flown to capture the necessary images of the state route. A total of 2,446 images were captured with an approximate overlap percentage of 80% and an average GSD of

1cm/px. Figures 93 and 94 shows the screenshots of the images captured using for each of the eleven flights conducted at SR26.

The positions of the targets that were going to be used as GCPs, were selected following Pix4D recommendations [30]. The selected GCPs were distributed uniformly on the mapped region.

All the images captured were processed together using GCPs in two different ways. The first version was processed using 32 GCPs selected uniformly across the entire region. The remaining 61 points were used to analyze the point cloud accuracy by comparing the computed point cloud coordinates of the checkpoint target centers with the coordinates recorded by the ODOT surveyor. The point cloud was referenced in the NAD83 2011 Ohio south horizontal coordinate reference system and the vertical coordinates system was set to arbitrary on Pix4D Mapper. Figure 95 shows the pointwise difference computed for the checkpoints between the point cloud and surveyor. Despite using 32 GCPs to calibrate the point cloud, the desired accuracies were not obtained. To maintain the uniform selection of GCPs used to calibrate the model, mag-nails that were barely visible had to be selected for the calibration.

It was noticed that the centers of a few mag-nails were not visible in the images. This led to segregating the mag-nails into three different quality types: Good, Maybe and Bad. Figure 96 shows the table of mag-nails that were segregated and sample images of points that were good, maybe and bad. The mag-nails that were classified as good was used to process the second version of the model on Pix4D. The good mag-nails were further divided into GCPs and checkpoints as shown in Figure 96. The second version of the Pix4D project was then processed using 22 GCPs and the remaining 17 points were used as checkpoints. The point cloud was referenced in the NAD83 2011 Ohio south horizontal coordinate reference system and the vertical coordinates system was set to arbitrary on Pix4D Mapper. The good mag-nails that were used as GCPs were laid out uniformly as possible as shown in Figure 97.

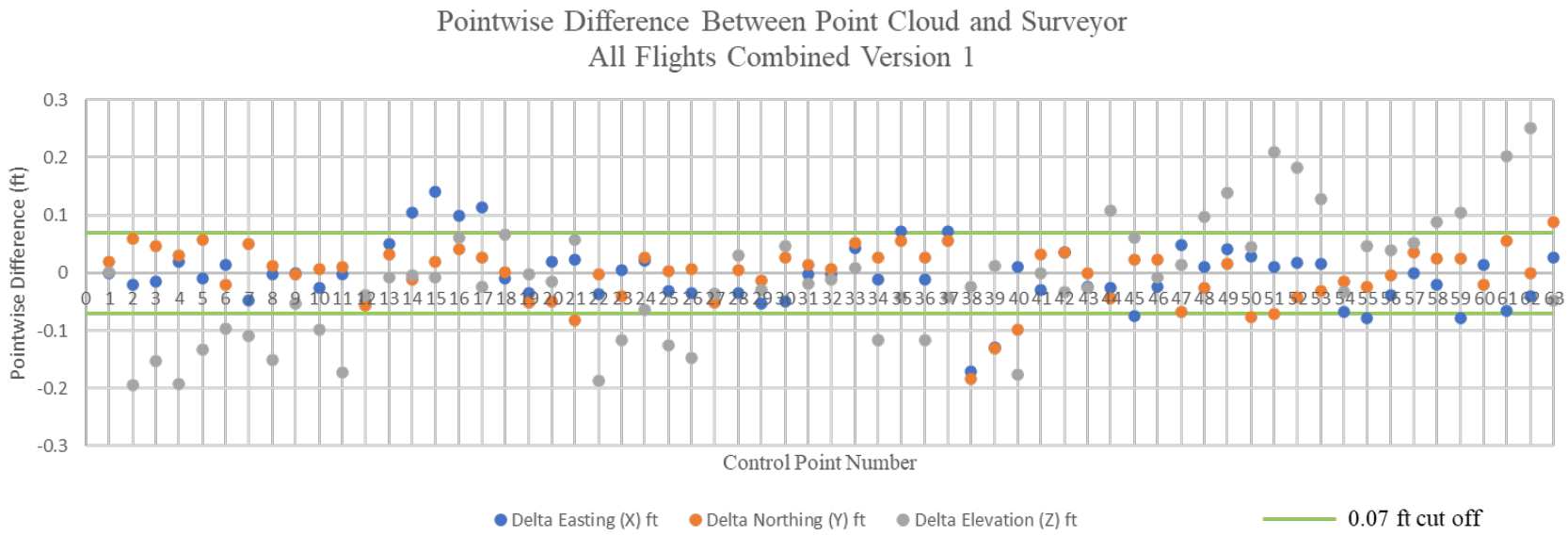


Figure 95: Pointwise Error Plot Calculated for Each of the Check Points used for Version 1 of the Point Cloud Processed using the SR72 Dataset

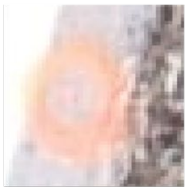
Number	Good Points	Maybe Points	Bad Points
1	VA2	VA1	VA10
2	VA3	VA4	VA12
3	VA9	VA5	VA16
4	VA15	VA6	VA17
5	VA20	VA7	VA18
6	VA21	VA8	VA19
7	VA23	VA11	VA22
8	VA26	VA13	VA24
9	VA28	VA14	VA25
10	VA32	VA30	VA27
11	VA34	VA33	VA29
12	VA38	VA102	VA31
13	VA100	VA106	VA35
14	VA101	VA116	VA36
15	VA104	VA119	VA37
16	VA105	VA126	VA103
17	VA108	VA133	VA107
18	VA109	VA134	VA110
19	VA111		VA112
20	VA117		VA113
21	VA118		VA114
22	VA120		VA115
23	VA121		VA122
24	VA124		VA123
25	VA132		VA125
26	VA135		VA127
27	VA136		VA128
28	VA140		VA129
29	VA141		VA130
30	VA142		VA131
31	VA143		VA137
32	VA144		VA138
33	VA147		VA139
34	VA149		VA145
35	VA150		VA146
36	VA151		VA148
37	VA152		
38	VA153		
39	VA154		

"Good points" were used as GCPs and CPs for the new version

Number	GCPs	CPs
1	VA2	VA3
2	VA9	VA21
3	VA15	VA23
4	VA20	VA26
5	VA28	VA34
6	VA32	VA100
7	VA38	VA109
8	VA101	VA111
9	VA104	VA117
10	VA105	VA121
11	VA108	VA124
12	VA118	VA136
13	VA120	VA140
14	VA132	VA142
15	VA135	VA151
16	VA141	VA152
17	VA143	VA153
18	VA144	
19	VA147	
20	VA149	
21	VA150	
22	VA154	

GCP: VA32 (Ref: Flight 8)
IMG: 006 (Good Mag-nail)

GCP: VA1 (Ref: Flight 11)
IMG: 005 (Maybe Mag-nail)



GCP: VA12 (Ref: Flight 10)
IMG: 061 (Bad Mag-nail)

Figure 96: Table Showing the Classification of Mag-nails as Good, Maybe and Bad; Center: Table Showing the Good Mag-nails Classified as GCPs or Check Points; Right: Images of Mag-nails that were Classified as Good, Maybe and Bad

Figure 97: Location of the Good Mag-nails Used as GCPs to Process Version 2 of the Pix4D Project

Pointwise Difference Point Cloud				
Number	Point Name	Delta Easting (X) ft	Delta Northing (Y) ft	Delta Elevation (Z) ft
1	VA21	0.008	0.0132	0.0675
2	VA26	-0.016	-0.0213	0.037
3	VA100	0.01	-0.0126	-0.0667
4	VA136	0.021	-0.0253	-0.0568
5	VA151	-0.013	0.0203	-0.056
6	VA152	-0.018	-0.0585	-0.0654
7	VA153	0.018	-0.0142	-0.0691
8	VA109	-0.046	-0.0357	-0.0346
9	VA142	-0.013	-0.048	0.0395
10	VA140	0.023	0.0364	-0.0592
11	VA117	0.029	-0.0089	-0.0565
12	VA121	0.016	-0.0103	0.0635
13	VA124	-0.041	-0.0403	0.0187
14	VA34	-0.008	0.0216	0.0341
15	VA23	0.019	-0.0444	0.069
16	VA111	0.049	-0.0314	0.0234
17	VA3	-0.006	0.0585	0.0662

	Delta Easting (ft)	Delta Northing (ft)	Delta Elevation (ft)
Mean of Absolute Error	0.021	0.029	0.052
Std. Deviation	0.025	0.032	0.056
Variance	0.001	0.001	0.003
RMSE	0.024	0.033	0.054

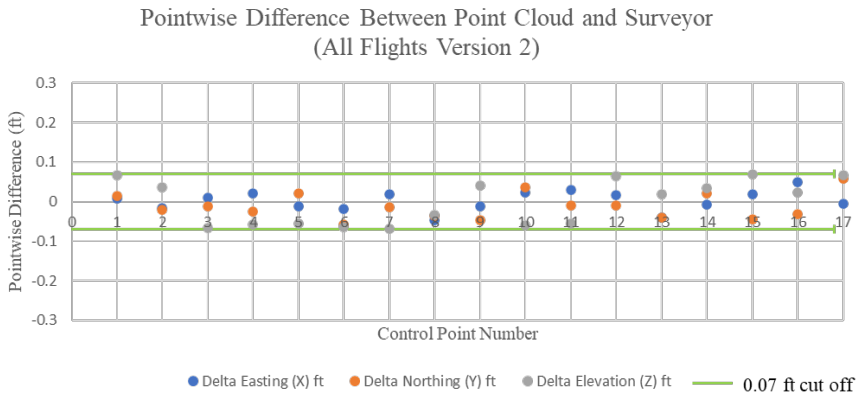


Figure 98: Error Data Compiled for Version 2 of the 3D Point Cloud Processed Using the SR72 Dataset

Figure 98 shows the individual pointwise errors calculated for all the checkpoints and the corresponding error plot. The mean, standard deviation, variance and RMSE of these values are also shown in Figure 98.

Analyzing and comparing the error plots obtained using version 1 (Figure 95) and version 2 (Figure 98) apart from uniformly selecting the GCPs it is equally important to ensure the GCPs are visible in the images captured to obtain accurate results. Blurred or poorly marked GCPs would lead to incorrectly selecting the centers of the mag-nail and therefore has a direct impact on the accuracies of the 3D point clouds obtained as shown in the populated results.

Figure 99 shows the point cloud generated for this case study. The obtained point cloud was converted to a linework 3D drawing using a Microstation plugin called TOPODOT. Figure 100 shows the resultant drawing obtained using the 3D point cloud.

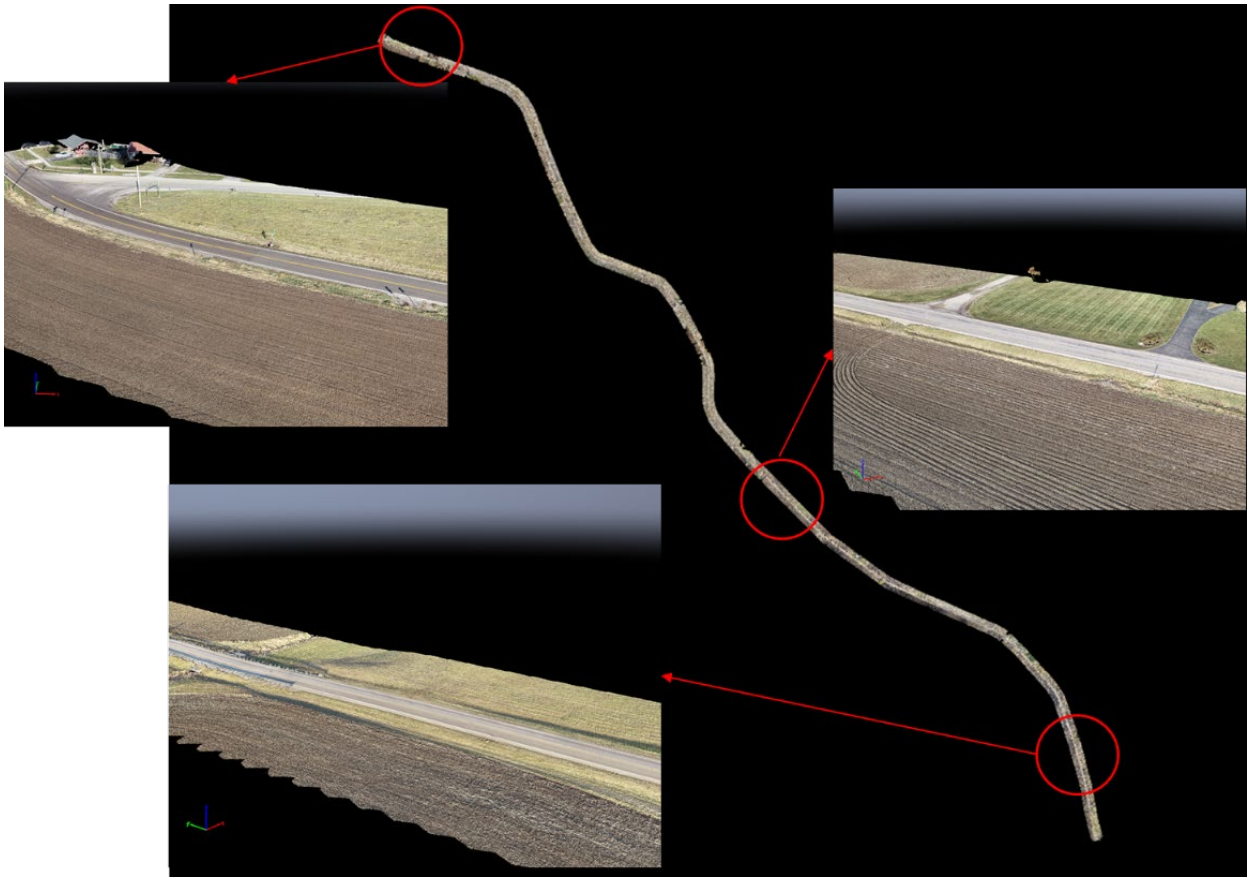


Figure 99: 3D Point Cloud Processed for the SR 72 Case Study

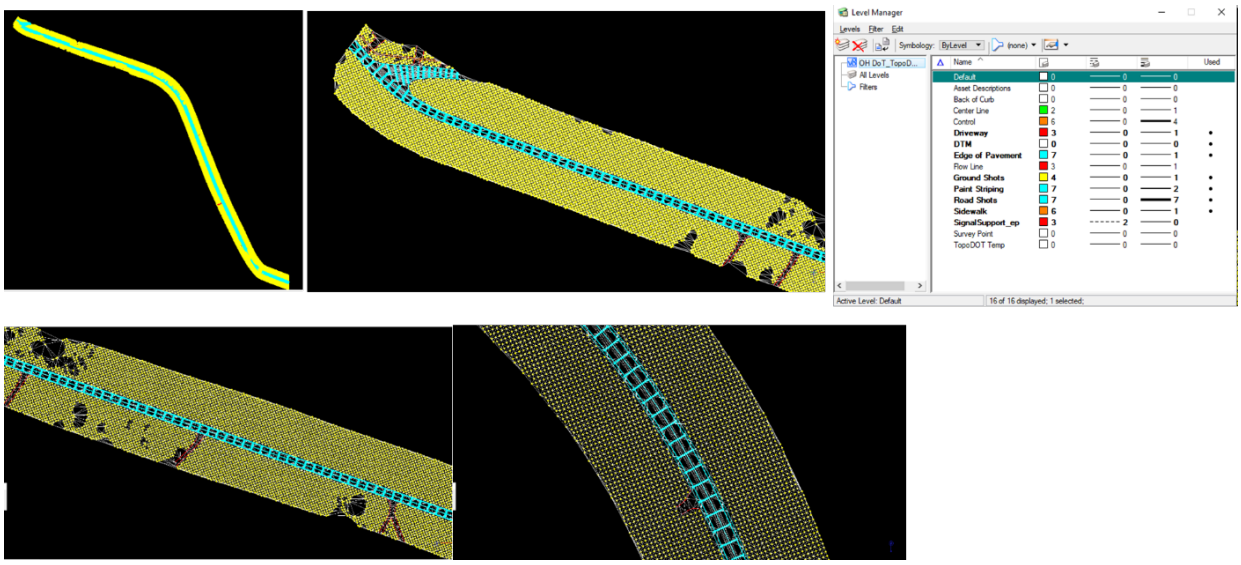


Figure 100: 3D Linework Drawing Generated Using the 3D Point Cloud Show in Figure 99

3.4 Inspection of Bridges and Facilities

The use of UAVs as airborne platforms for aerial mapping was discussed in the previous section. Manny of the characteristics which make UAVs well suited to that application also make them well suited for inspection purposes, particularly for bridges and buildings. High maneuverability, the ability to reach places where it would be difficult for humans to go, that ability to obtain very accuracy 2D and 3D modeling all factor in. As a result, the research team pursued research in the use of UAVs to help augment bridge and facility inspection. Similar to the discussion in Section 3.3, the use of UAVs increased safety for both ODOT employees and the traveling public in that they could be deployed quickly and operated without the need for lane close or the deployment of snooper vehicles.

UAVs do have some limitations, however. These include primarily: (1) weather conditions must be advantageous (e.g., there are operational limitations on wind, precipitation, etc.); (2) operation in GPS denied environments (such as the interior of bridge or building structures requires expert piloting, and (3) the images obtained must be post processed by the application of photogrammetry software (e.g., Pix4D) in order to obtain global 2D and 3D computer models.

The results reported here borrow heavily from the SOPs developed by the research team for use by Ohio UAS Center pilots and personnel as part of this project [5,31] as well as the theses and papers generated by the various graduate students involved in this research project [26]. The reader is referred to these documents for additional information and details.

3.4.1 Flight Planning and Vehicle/Sensor Considerations

There are 3 main sUAS used throughout this project for inspection of Facilities and Bridges. These vehicles and cameras were selected depending on the particular flights being done.

Facilities Inspection: The main sUAS used for facilities was the Matrice 100. Similar to the construction site and aerial mapping the DJI GS Pro was used for flight planning. The main difference to planning these missions is more planning needed to be done to ensure correct GSD and overlap above the buildings. A simple calculator was used allowed the team to plan missions and easily ensure the correct settings. A diagram of this calculator is shown in Figure 101. The GS Pro app allowed the team to specify the overlap and GSD on the ground, but had limitations when planning missions at different altitudes. The main issue is when flying over a building we would have a better GSD, but would loose overlap so we would need to adjust to ensure we had appropriate coverage. This would not only be used with facilities, but also with inspecting bridge decks.

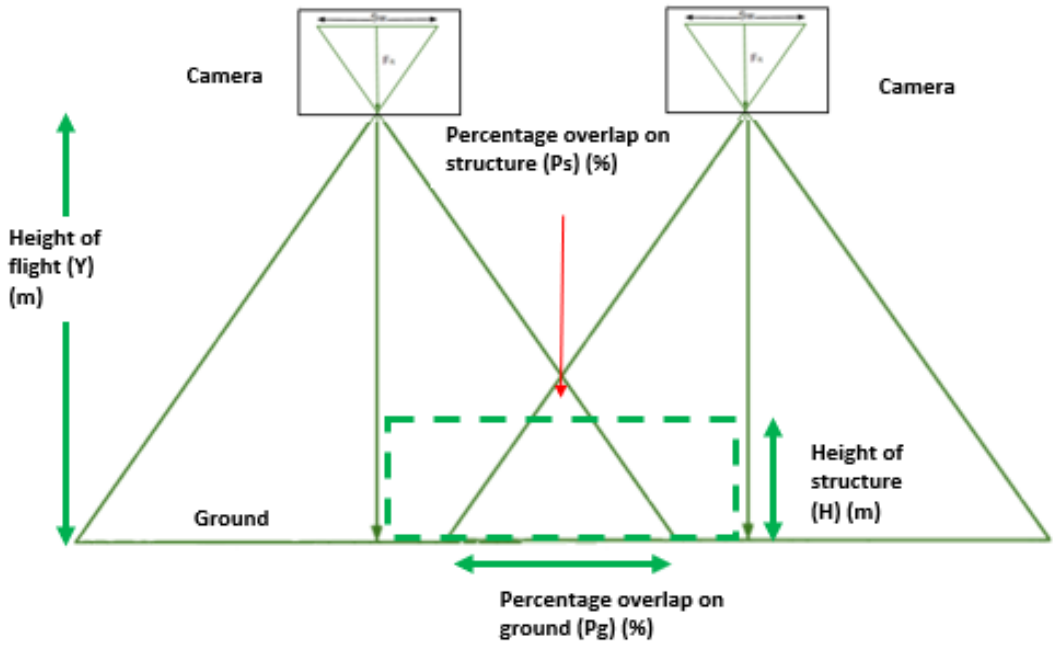


Figure 101: GSD and Overlap Diagram

Many experiments were run to create the most accurate models. The flow chart shown in Figure 102 was used to plan missions. The first step was to get the structure specifics and the flight purpose. This would determine GSD and overlap as well as the need for GCPs, Manual Tie Points, or RTK. We would need input the camera settings and feed this into the calculator. If adjustments were needed, we would adjust GSD or overlap as necessary. Next, we would look at mission check that include checking

for possible obstructions and flight time. Once we know that we will be able to conduct the flight we would upload and fly the mission. Finally, we would use Pix4D to create the appropriate 2D and 3D models.

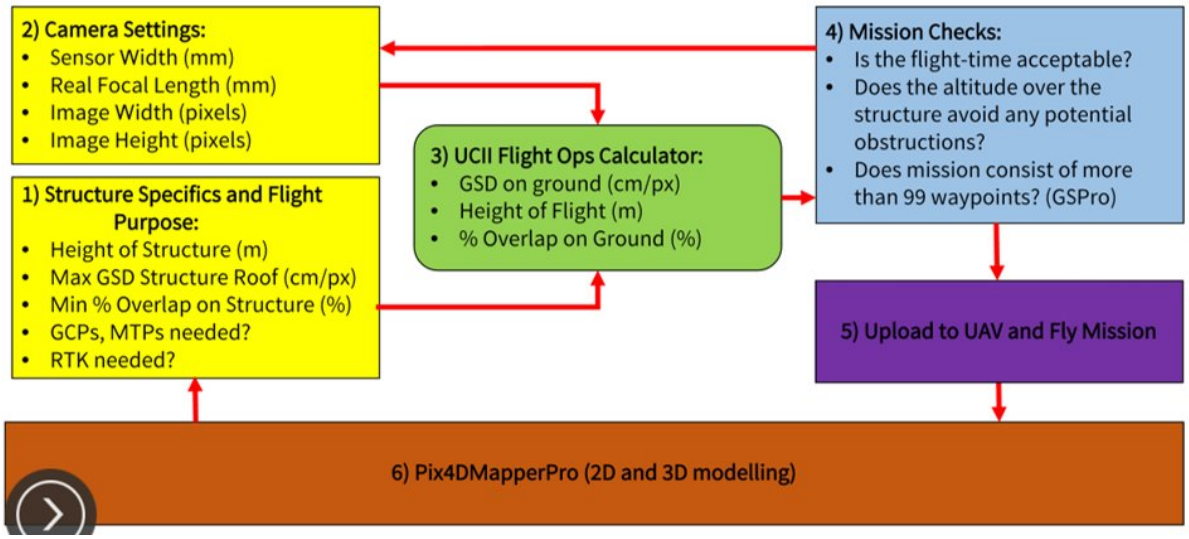


Figure 102: Flow Diagram for Mapping Missions

Bridge Inspections: For bridge inspections three sUAS were used. First is the Matrice 100 with the X5 or XTR camera. This was mostly used during bridge deck inspections. Next, we used the Matrice 210 for inspections on the side or under the bridge. The top mounted camera allowed for less obstructed views and was essential for inspecting overhangs. Lastly, the Flyability Elios was used to inspect confined spaces such as expansion joints, hangers, or the interior of some bridges.

As mentioned above, the thermal camera was used heavily to inspect delamination on the bridge deck. The team would also use the thermal camera to try to identify cracking, but the resolution and the small cracking was difficult to see with the thermal camera. Instead, the team would use either the Zenmuse Z30 or X5S cameras to see smaller cracking.

The team would start by using a crack comparator card (Figure 103) as reference to determine distance away from the structure, camera and zoom used, and the best GSD that can be used to see cracking.

A set of experiments were conducted in the lab to check the effect of distance when documenting cracks. The X5S and Z30 were both used at varying zoom levels from 2ft to 100ft. A heat map was created of theoretical GSD and distance. This heat map shown in Figure 104 allowed the team to select a desired GSD for the mission, then lookup the required camera and specifications and finally get the required distance away from the bridge required.

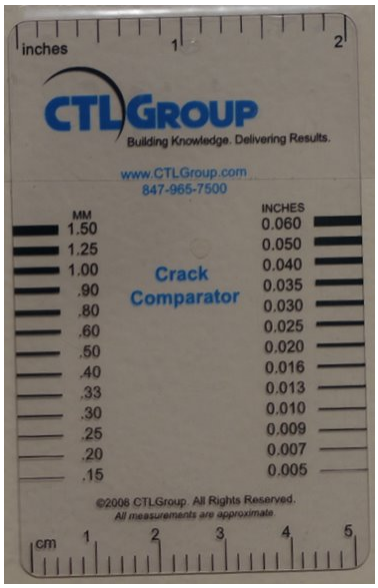


Figure 103: Crack Comparator Card

Crack Detection Experiment: M210 with Z30 and X5S Cameras GSD values (mm/px)

Camera	Z30								X5S	
Zoom Level / Distance (feet/m)	1	2	5	10	15	20	25	30	Lens = 15mm	Lens=45mm
2(0.6096m)	0.348mm/px	0.1739mm/px	0.0695mm/px	0.03478mm/px	0.02318mm/px	0.01739mm/px	0.0139mm/px	0.01159mm/px	0.133mm/px	0.044mm/px
3(0.9144m)	0.5216mm/px	0.2608mm/px	0.1043mm/px	0.0522mm/px	0.03478mm/px	0.02608mm/px	0.02087mm/px	0.017389mm/px	0.1997mm/px	0.066mm/px
4(1.2192m)	0.6955mm/px	0.3478mm/px	0.1391mm/px	0.06955mm/px	0.0464mm/px	0.03478mm/px	0.02782mm/px	0.02318mm/px	0.2663mm/px	0.089mm/px
5(1.524m)	0.8694mm/px	0.4347mm/px	0.17389mm/px	0.0869mm/px	0.0579mm/px	0.04347mm/px	0.03478mm/px	0.02898mm/px	0.333mm/px	0.111mm/px
6(1.8288m)	1.04332mm/px	0.5216mm/px	0.20867mm/px	0.10433mm/px	0.06955mm/px	0.05216mm/px	0.04173mm/px	0.03478mm/px	0.3995mm/px	0.133mm/px
7(2.1336m)	1.2172mm/px	0.6086mm/px	0.2434mm/px	0.12172mm/px	0.08114mm/px	0.06086mm/px	0.04869mm/px	0.04057mm/px	0.4661mm/px	0.155mm/px
8(2.4384m)	1.391mm/px	0.6955mm/px	0.2782mm/px	0.1391mm/px	0.09274mm/px	0.06955mm/px	0.0556mm/px	0.04637mm/px	0.533mm/px	0.177mm/px
9(2.7432m)	1.565mm/px	0.7825mm/px	0.31299mm/px	0.1565mm/px	0.10433mm/px	0.07825mm/px	0.06259mm/px	0.05216mm/px	0.5992mm/px	0.199mm/px
10(3.048m)	1.7389mm/px	0.8694mm/px	0.34778mm/px	0.1739mm/px	0.1159mm/px	0.08694mm/px	0.06955mm/px	0.05796mm/px	0.6658mm/px	0.222mm/px
15(4.572m)	2.6082mm/px	1.3041mm/px	0.5216mm/px	0.2608mm/px	0.17389mm/px	0.13041mm/px	0.10433mm/px	0.0869mm/px	0.999mm/px	0.333mm/px
20(6.096m)	3.4777mm/px	1.7389mm/px	0.6955mm/px	0.3478mm/px	0.23184mm/px	0.1739mm/px	0.1391mm/px	0.1159mm/px	1.3316mm/px	0.444mm/px
25(7.62m)	4.3472mm/px	2.1736mm/px	0.8694mm/px	0.43472mm/px	0.2898mm/px	0.21736mm/px	0.17389mm/px	0.1449mm/px	1.664mm/px	0.555mm/px
30(9.144m)	5.2166mm/px	2.6083mm/px	1.0433mm/px	0.52166mm/px	0.3478mm/px	0.26083mm/px	0.20867mm/px	0.17389mm/px	1.997mm/px	0.666mm/px
35(10.668m)	6.086mm/px	3.043mm/px	1.2172mm/px	0.6086mm/px	0.40573mm/px	0.3043mm/px	0.2434mm/px	0.20287mm/px	2.3302mm/px	0.777mm/px
40(12.192m)	6.9555mm/px	3.4778mm/px	1.3911mm/px	0.6955mm/px	0.4637mm/px	0.3478mm/px	0.2782mm/px	0.23185mm/px	2.663mm/px	0.888mm/px
45(13.716m)	7.8249mm/px	3.9124mm/px	1.564mm/px	0.7825mm/px	0.5216mm/px	0.391245mm/px	0.31299mm/px	0.2608mm/px	2.996mm/px	0.999mm/px
50(15.24m)	8.694mm/px	4.3471mm/px	1.7389mm/px	0.8694mm/px	0.5796mm/px	0.43472mm/px	0.3478mm/px	0.2898mm/px	3.329mm/px	1.11mm/px
55(16.764m)	9.5638mm/px	4.7818mm/px	1.9127mm/px	0.9564mm/px	0.6376mm/px	0.4782mm/px	0.38255mm/px	0.31879mm/px	3.6618mm/px	1.221mm/px
60(18.288m)	10.433mm/px	5.2166mm/px	2.0866mm/px	1.0433mm/px	0.6955mm/px	0.52166mm/px	0.41733mm/px	0.3478mm/px	3.995mm/px	1.331mm/px
65(19.812m)	11.3026mm/px	5.651mm/px	2.2605mm/px	1.1303mm/px	0.7535mm/px	0.56513mm/px	0.4521mm/px	0.3767mm/px	4.3276mm/px	1.442mm/px
70(21.336m)	12.172mm/px	6.086mm/px	2.4344mm/px	1.2172mm/px	0.81147mm/px	0.6086mm/px	0.48688mm/px	0.4057mm/px	4.6605mm/px	1.553mm/px
75(22.86m)	13.0414mm/px	6.5207mm/px	2.608mm/px	1.304mm/px	0.8694mm/px	0.65207mm/px	0.52166mm/px	0.4347mm/px	4.9934mm/px	1.66mm/px
80(24.384m)	13.9109mm/px	6.955mm/px	2.7821mm/px	1.391mm/px	0.9274mm/px	0.6955mm/px	0.5564mm/px	0.46369mm/px	5.3263mm/px	1.775mm/px
85(25.908m)	14.7804mm/px	7.39018mm/px	2.9561mm/px	1.478mm/px	0.985mm/px	0.739mm/px	0.5912mm/px	0.49268mm/px	5.6592mm/px	1.886mm/px
90(27.432m)	15.6498mm/px	7.8249mm/px	3.1299mm/px	1.5649mm/px	1.0433mm/px	0.7825mm/px	0.6259mm/px	0.52166mm/px	5.6621mm/px	1.997mm/px
95(28.956m)	16.5192mm/px	8.2596mm/px	3.3038mm/px	1.652mm/px	1.1013mm/px	0.82596mm/px	0.6607mm/px	0.55064mm/px	6.325mm/px	2.108mm/px
100(30.48m)	17.3887mm/px	8.6943mm/px	3.4778mm/px	1.7389mm/px	1.1592mm/px	0.8694mm/px	0.6955mm/px	0.5796mm/px	6.658mm/px	2.22mm/px

Figure 104: Heat Map used for Camera Zoom vs GSD Values

While the closer the sUAS was to the bridge the better the GSD would be, but this would also limit autonomous flights. Manual flights were flown for the majority of the bridges as this gave the most control. The pilot would focus on operating the sUAS 10-20ft away from the bridge while a camera operator would set the zoom and other camera settings. An example image on one such inspection is shown in Figure 105. Cracking and measurements would be taken by flying patterns to collect hundreds of images per segment.

Figure 105: Example Image from Bridge Inspection

Flyability Elios: The last sUAS used for bridge inspections was the Flyability Elios discussed in Section 3.1.2. The Elios was used to fly in confined spaces that would be impossible for other sUAS and difficult for inspectors to reach. An example is shown in Figure 106. The Elios had a limited flight time and was very susceptible to wind, so it was only used on calm days. The Elios would also be best used for inspections where it would make contact with the bridge to roll and hover where needed.

Figure 106: Flyability Elios Inspection of Expansion Joint

The Elios was also used in a dual operator mode, with one person flying the vehicle and the other controlling the camera and lighting. It was very easy to loose orientation of the Elios so both pilot and inspector would need to be familiar with the Bridge Inspection Manual so they can orient themselves with the structure.

3.4.2 Use of Optical vs IR cameras

Facilities was also the first time the team would use the DJI Zenmuse XTR camera discussed in Section 3.1.4. This camera allowed the team to inspect buildings for heat loss and stagnant water on the roofs. Both visual and thermal inspections of facilities would be studied and compared to show areas of concern. Figure 107 shows some of this comparison where the team created 2D orthomosaics to show non-uniform heating in the building.

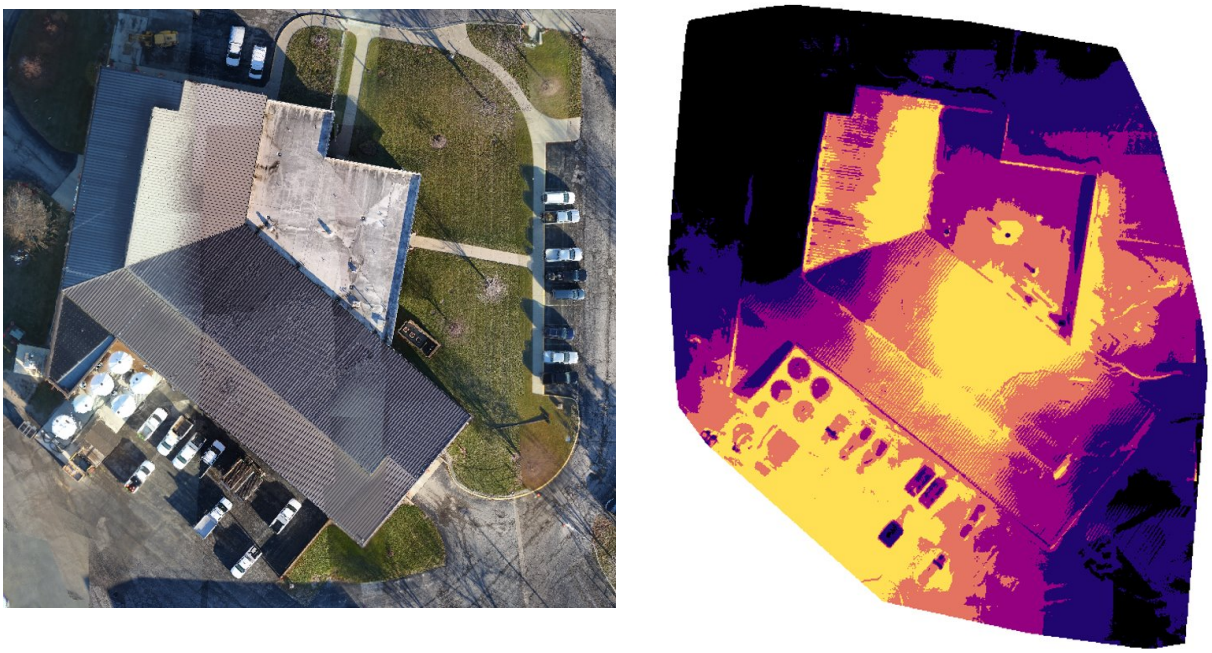


Figure 107: Visual and Thermal 2D Orthomosaics

The thermal camera also allowed the team to see delamination during bridge inspections. Figure 108 shows one of this images. The delamination in the roadway will heat and cool at a different rate than the surrounding areas so taking thermal images in the morning or as the sun is setting allows for the thermal team to pick up these areas.

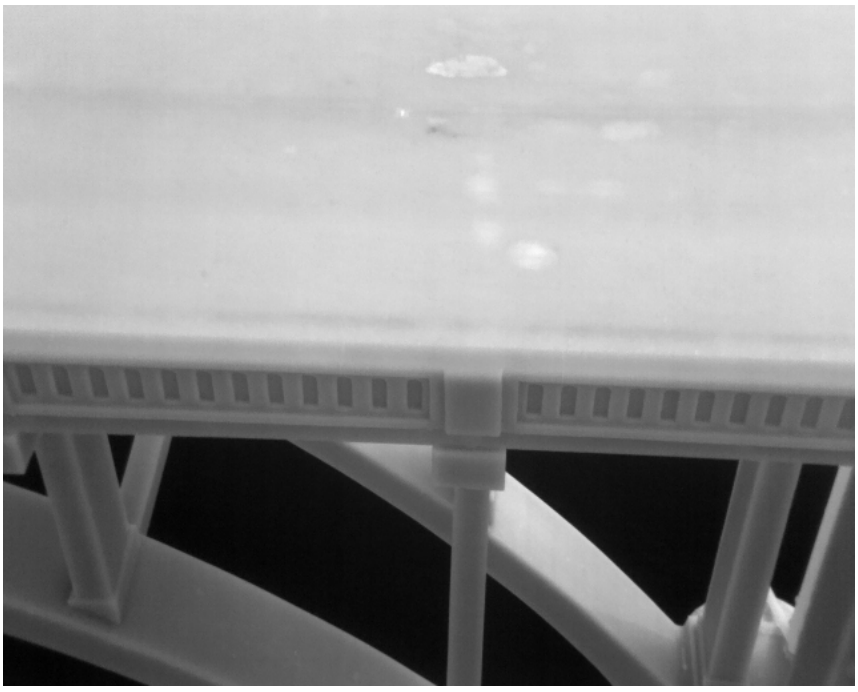


Figure 108: Thermal Delamination Example

3.4.3 Application Examples

Jeremiah Morrow Bridge Demolition: One of the first missions conducted was the demolitions of the old Jerimiah Morrow Truss Bridge located along IR-71 over the Little Miami River in Lebanon, OH. This mission was to observe the demolition from a safe distance and record the video for posterity. In addition, demolition video recorded by the research team was used by ODOT for several purposes: First, it was released to the press by ODOT as there were TFR's imposed in the area on the date of demolition. Second, it was used for analysis and decision-making purposes on the part of ODOT authorities in order to gauge the efficacy and safety of the demolition process before taking next steps at the construction site.

This would also be the first time the team would connect to Milestone, ODOT's traffic camera system. This connection allowed for the team to stream the video from the UAS through a cellular network modem to the milestone server allowing for anyone with the milestone access to view the demolition live. This would eventually lead to the Milestone Mission Box discussed below in Section 3.6.

The UAS used would be the Yuneec Typhoon H. Photos of the operations and a screenshot from the demolition are shown in Figures 109-110.

Figure 109: Jerimiah Morrow Bridge operations and staging

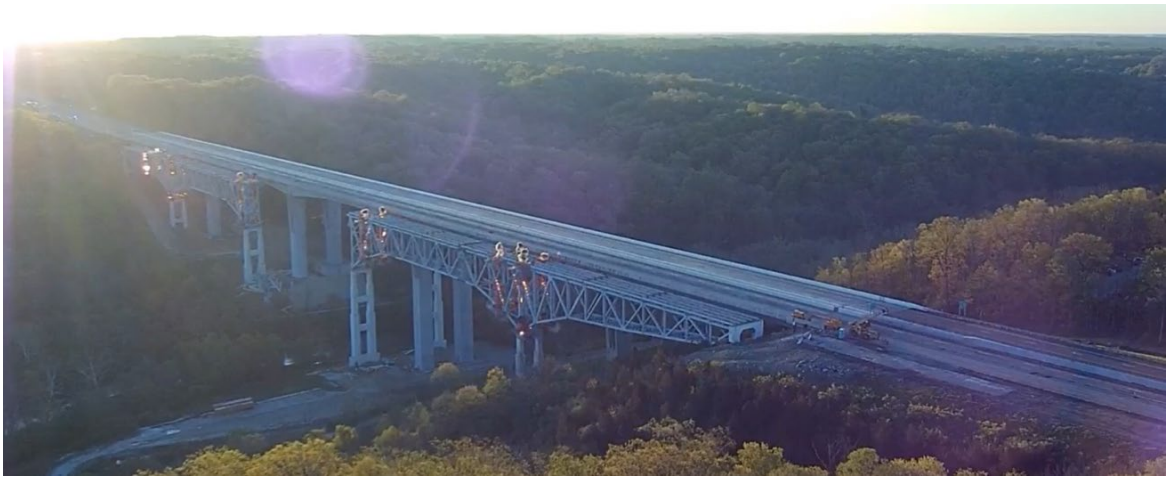


Figure 10: Demolition of Jeremiah Morrow Bridge

Veteran's Glass City Skyway (VGCS) Interior Inspection: The team would also experiment with conducting interior inspections of bridges such as the VGCS in Toledo, OH (Figure 112). This inspection would occur before the Flyability Elios was available so other UAS were needed. A total of 3 UAS would be developed and tested as part of this research project for this purpose. Each UAS would have a different payload and sensors and ultimately would vary in size. All 3 UAS are shown in Figure 113.

Figure 111: The VGCS Bridge in Toledo, OH

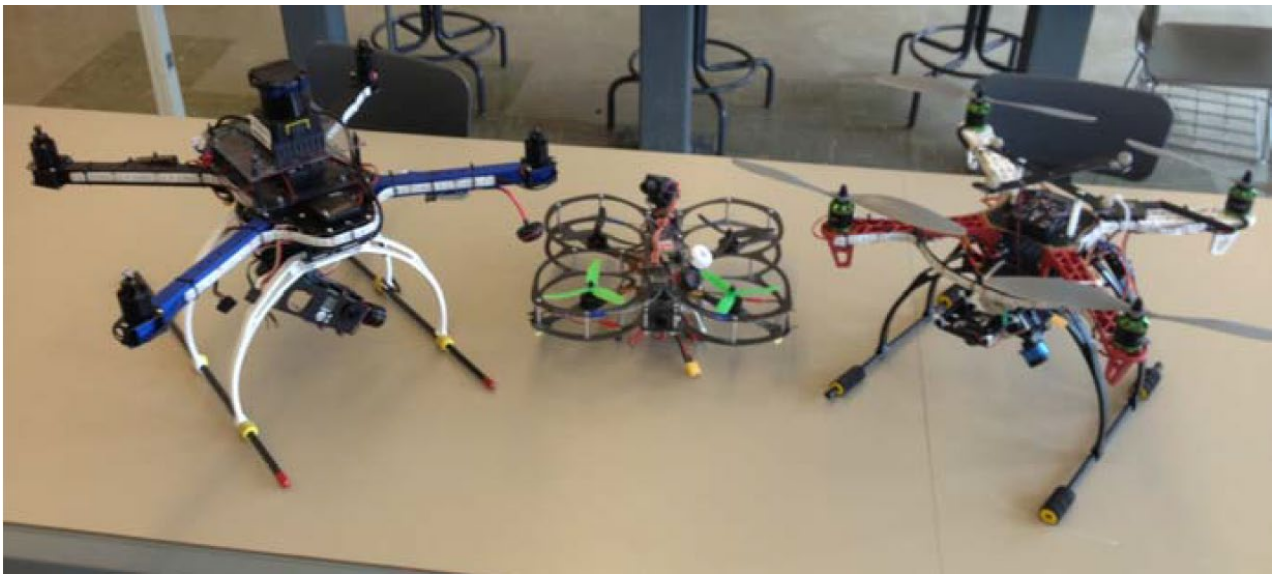


Figure 112: UAS developed for Interior Bridge Inspection

Figure 113: Close up of the middle UAS system developed, tested, and ultimately deployed at the VGCS

The leftmost UAS was equipped with lighting, a dual visual/thermal gimbal, and LIDAR for obstacle avoidance. This vehicle would be the ideal candidate for this type of inspection, but due to time constraints the obstacle avoidance was not fully flushed out by time of the testing. The vehicle on the right was only equipped with lighting and a visual camera so was able to be slightly smaller and have a longer flight time. Both vehicles previously mentioned ended up being too large to fit into many of the areas needed. As a result, the middle UAS (also shown in Figure 113) was built and would include 2 cameras, one for pilot to fly first person view and another for an inspector to control that was on a servo gimbal. Both cameras were connected to video transmitters which allowed either operator to see that camera. The vehicle would also be equipped with onboard lighting as well as having shrouding for propeller protection and collision safety.

The demonstration of the operation of this middle vehicle is shown in Figures 114-115. The UAS (2 dots on the left side of the image) and both video feeds are shown in Figure 114.

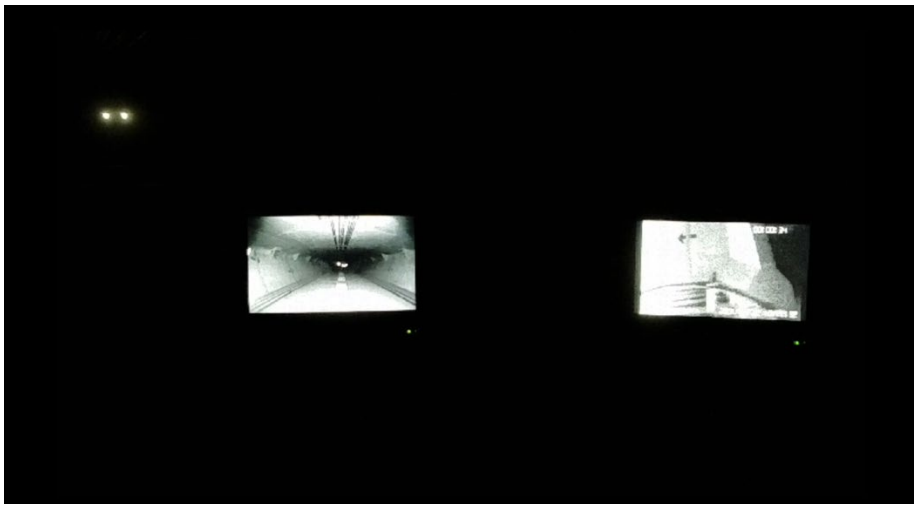


Figure 114: Interior Inspection with Lights off

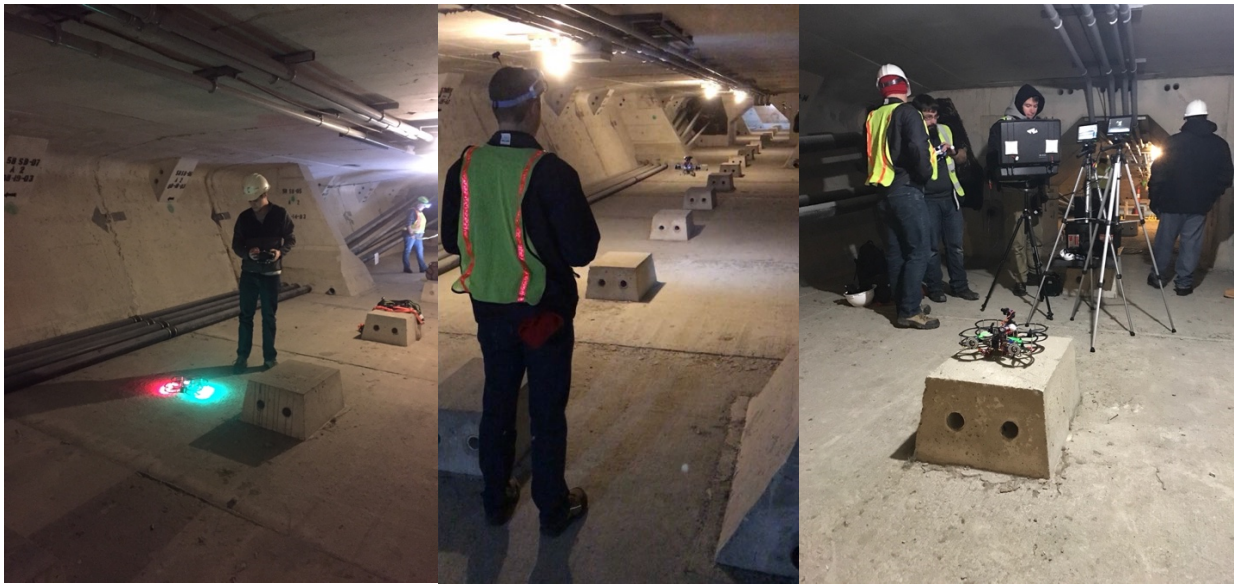


Figure 115: Interior Inspection with Lights on

Fosters Bridge Deck Inspection: This section presents the case study conducted at Fosters Bridge in Maineville, Ohio along Rte 22 as it crosses over the Little Miami River. The objective of this study was to map the bridge deck using oblique and nadir thermal images to identify delamination and compare the results obtained using both the datasets to the results obtained using the chain drag and hammer sounding methods. Thermal and visual images of the deck were captured to cross-refer identified delaminated spots.

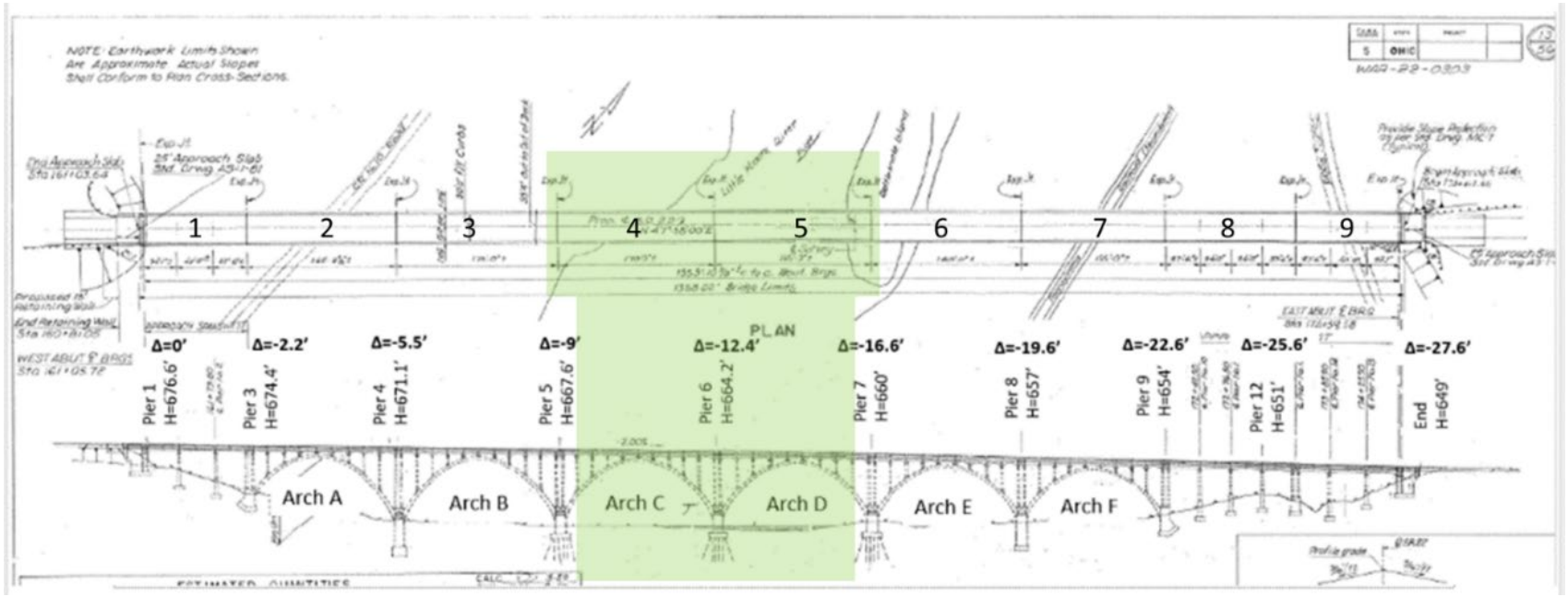


Figure 116: Fosters Bridge Plan Sheet

The plan sheets of the bridge were analyzed prior to designing flight plans. Due to issues of accessibility and line of sight, it was decided to map out the decks of spans 4 and 5 (Figure 116). For this case study, two types of missions were conducted. The first type involved capturing thermal and visual oblique images of the bridge when the bridge was open and functional. This was achieved by capturing images in a single line along the side of the bridge at different camera angles. The second type involved capturing thermal and visual nadir images when the bridge was closed. The thermal images were captured after 12PM during summertime, giving the bridge's deck ample time to heat up allowing users to clearly identify delaminated spots on the bridge's deck. This was done when ODOT bridge inspectors closed the bridge for their routine inspections. The inspection was done lane wise while traffic was controlled in the second lane. Bridge inspectors inspected the bridge for delamination using the chain dragging method, and the trouble spots were marked using white paint. The visual images of the marked deck were captured to compare the delamination identified using the thermal images.

Using the information in the plan sheets a rough scaled illustration was created to understand the oblique thermal and visual image footprints. Thermal and visual images of the bridge deck were captured from both the north and south side of the bridge. DJI Matrice 100 sUAS and DJI XTR thermal camera and DJI X5 15mm visual cameras were used to capture the images. Figure 117 shows the illustrations for the DJI XTR camera that was used to generate the necessary flight plans. If the parameters (flight position and camera angle) are set correctly the thermal images captured would have average GSDs of 2.1cm/px (45^0), 2.4cm/px (47^0) and 2.7cm/px (50^0) respectively and 80% image side overlap and 95% top and bottom image overlap on the bridge deck. Figure118 shows the illustrations for the DJI X5 15mm lens camera that was used to generate the necessary flight plans. If the parameters (flight position and camera angle) are set correctly the visual images captured would have average GSDs of 2.1cm/px (45^0), 2.4cm/px (47^0) and 2.7cm/px (50^0) respectively and 80% image side overlap and 95% top and bottom image overlap on the bridge deck. The camera angles on DJI GS Pro is set by changing the gimbal pitch angle option. The gimbal pitch angle is measured with respect to the horizon and care was taken to set the required angles on the application. Figures 119-120 show the flight plans generated to capture oblique visual and thermal images at an angle of 45^0.

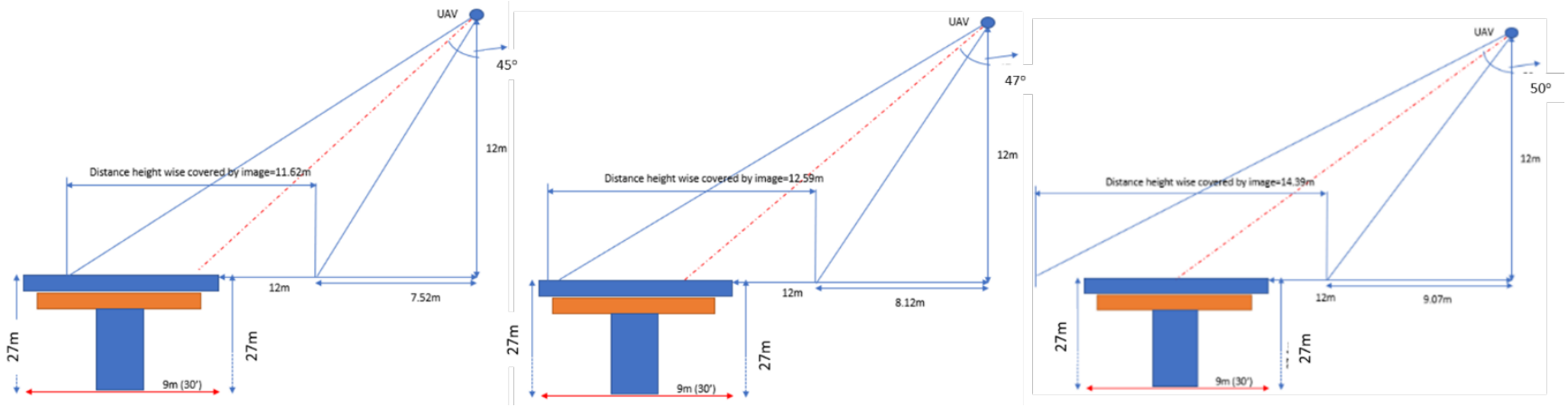


Figure 117: DJI XTR Thermal Oblique Image Footprint Illustration; Left: 45^0, Center: 47^0, Right: 50^0

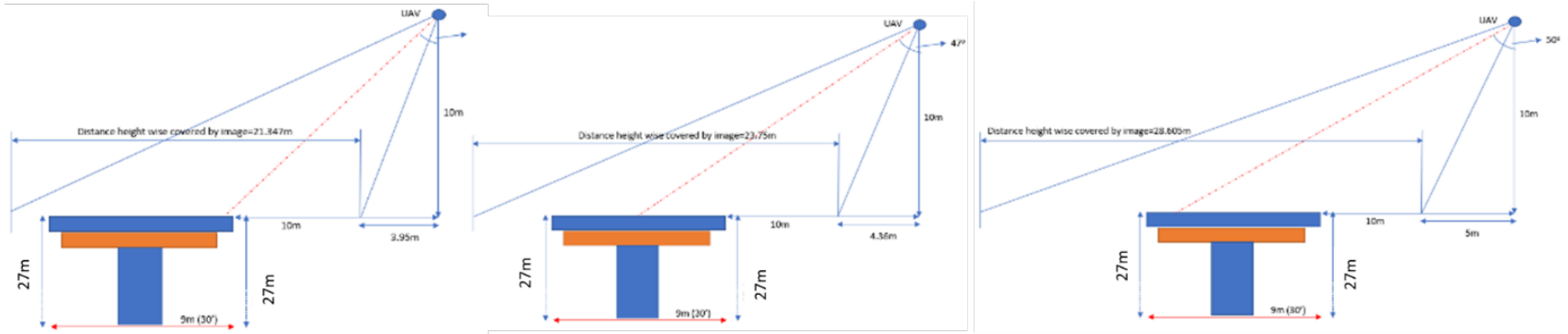


Figure 118: DJI X5 (15mm lens) Oblique Image Footprint Illustration; Left: 45^0, Center: 47^0, Right: 50^0

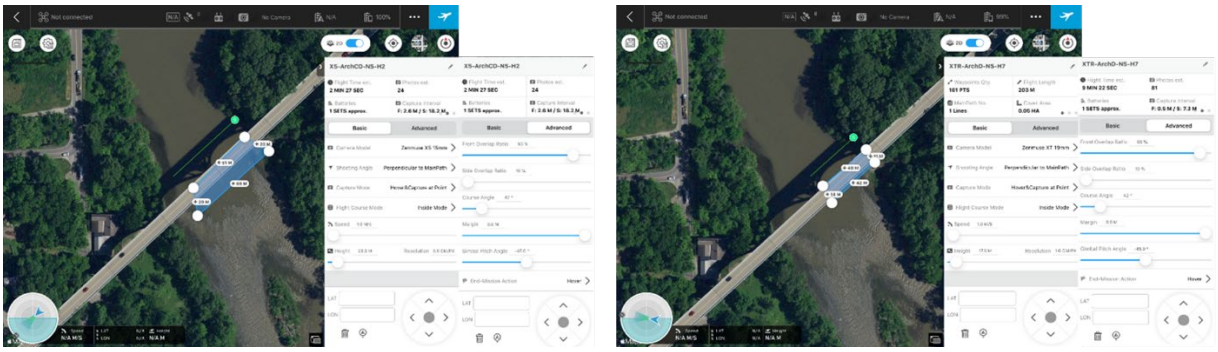


Figure 119: DJI GS Pro Flight Plan to Capture Oblique Images for the Fosters Bridge Case Study; Left: Visual Images, Right: Thermal Images

Image	Mission Type	Avg. GSD (cm/px)	Distance from Face/Above Deck (m)	Side Overlap (%)	Flight Time	Recorded Photos
Visual (North and South Sides)	Oblique (45^0)	0.54cm/px	10m/10m	90%	6min	90
	Oblique (47^0)	0.6cm/px	10m/10m	90%	6min	90
	Oblique (50^0)	0.72cm/px	10m/10m	90%	6min	90
Thermal (North and South Sides)	Oblique (45^0)	2.1cm/px	12m/12m	95%	20min	160
	Oblique (47^0)	2.4cm/px	12m/12m	95%	20min	164
	Oblique (50^0)	2.7cm/px	12m/12m	95%	20min	162

Image	Mission Type	Avg. GSD (cm/px)	Altitude (m)	FO/SO (%)	Flight Time	Recorded Photos
Visual	Nadir Parallel	0.2cm/px	9.3m	75%/75%	6min	180
Thermal	Nadir Parallel	1.5cm/px	17.2m	80%/80%	20min	250

Figure 120: Summary of mission conducted at the Fosters Bridge

The visual images were processed on Pix4D Mapper and the bridge deck's 3D point cloud and orthomosaic were generated. The outputs, using Pix4D Mapper, were generated without GCPs or scale constraints. Figure 121 shows the 3D model of the bridge deck. The thermal images were analyzed using FLIR Tools and Microsoft Word. The contrast of the thermal images was corrected to analyze the temperature differences on the bridge deck easily. The corrected images were laid out side by side on Microsoft Word and the delaminated regions were identified. Figure 122 shows the process followed to analyze the thermal images. The detected delaminated regions were then marked on the orthomosaic (Figure 123) generated.

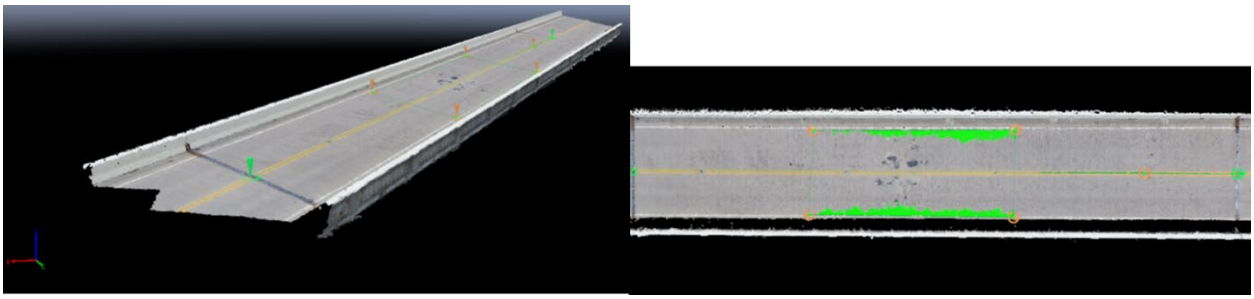


Figure 121: 3D model of Fosters Bridge spans 4 and 5

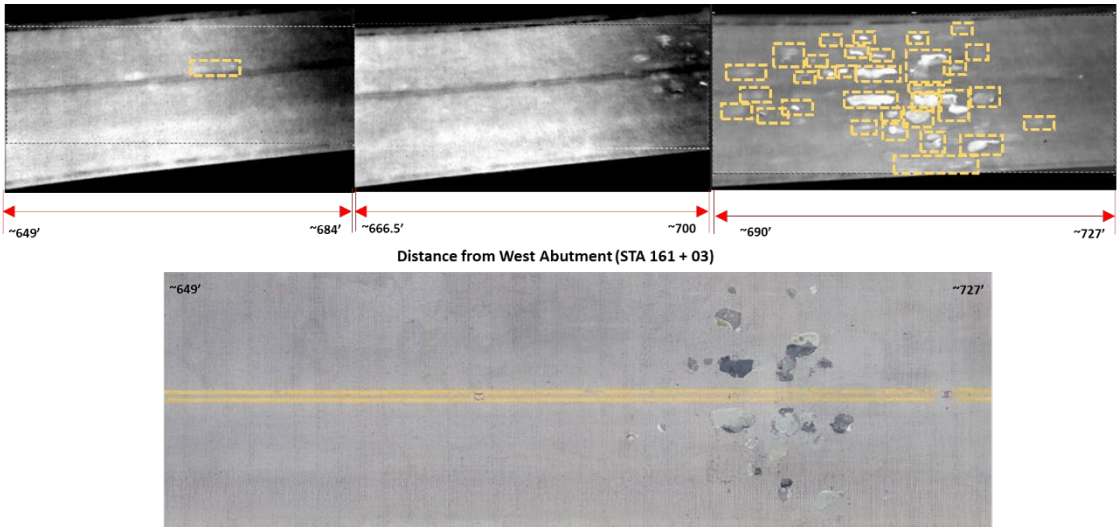


Figure 122: Thermal Image Analysis of the Fosters Bridge Deck; Top: Thermal Images Marked, Bottom: Portion of Visual 2D Orthomosaic

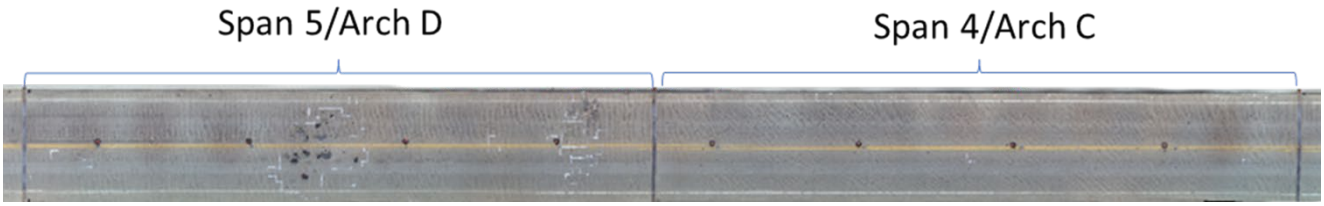


Figure 123: Visual 2D Orthomosaic of Fosters Bridge Deck

The captured thermal images were analyzed and detected delaminated regions were marked on visual orthomosaic images of the spans.

Figure 124 shows the comparison between the inspector marking and the markings obtained by analyzing the nadir thermal images of Span 4. The figure shows the percentage of area covered by the defects and indicates if the defects were seen or not on the results compared.

Figure 125 shows the comparison between the inspector marking and the markings obtained by analyzing the oblique thermal images of Span 4. The figure

shows the percentage of area covered by the defects and indicates if the defects were seen or not on the results compared.

Figure 126 shows the comparison between the inspector marking and the markings obtained by analyzing the nadir thermal images of Span 5. The figure shows the percentage of area covered by the defects and indicates if the defects were seen or not on the results compared.

Figure 127 shows the comparison between the inspector marking and the markings obtained by analyzing the oblique thermal images of Span 5. The figure shows the percentage of area covered by the defects and indicates if the defects were seen or not on the results compared.

The results obtained from this study show that a good estimation of delaminated/wearing surface positions can be obtained using oblique thermal images without closing the bridge or obstructing traffic. Analysis of the results shows that few spots marked by the inspector, on Span 4, were not visible in the oblique thermal images captured. Nadir and oblique thermal images captured almost all the marked delaminated areas. Nadir images captured finer details compared to oblique images.

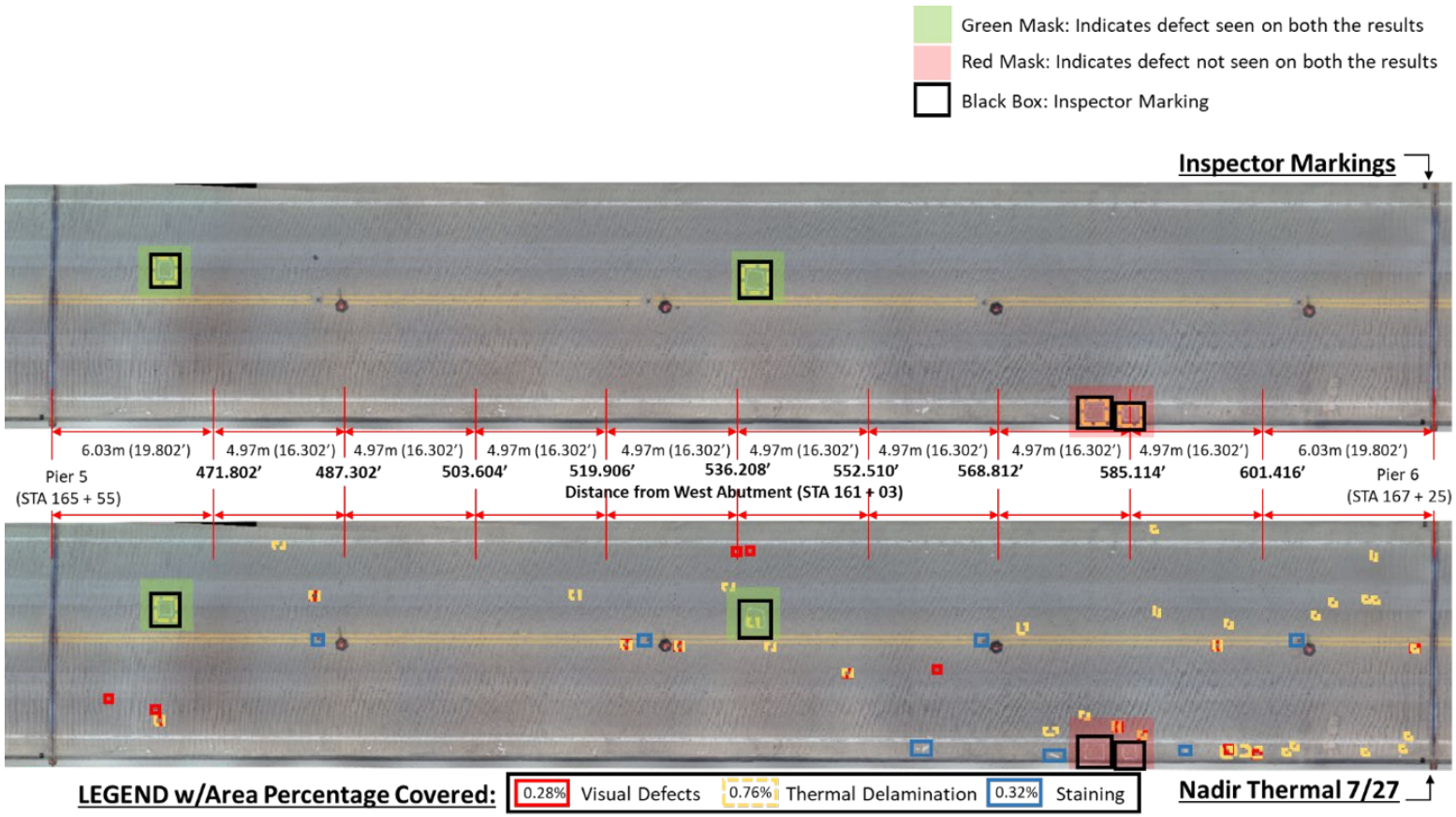


Figure 124: Delamination Marking Comparison of Span 4; Top: Inspector Markings, Bottom: Nadir Thermal Images

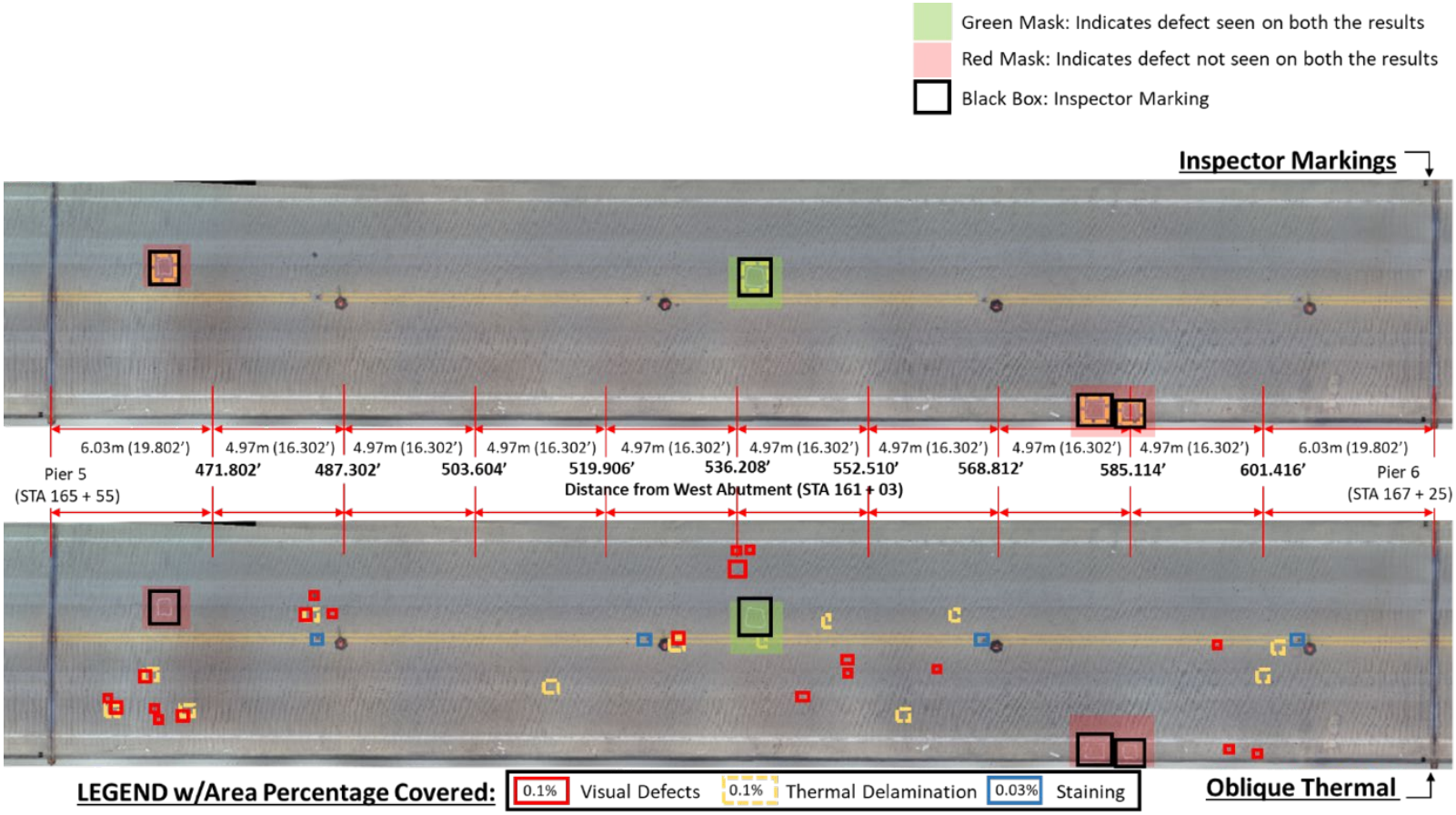


Figure 125: Delamination Marking Comparison of Span 4; Top: Inspector Markings, Bottom: oblique Thermal Images

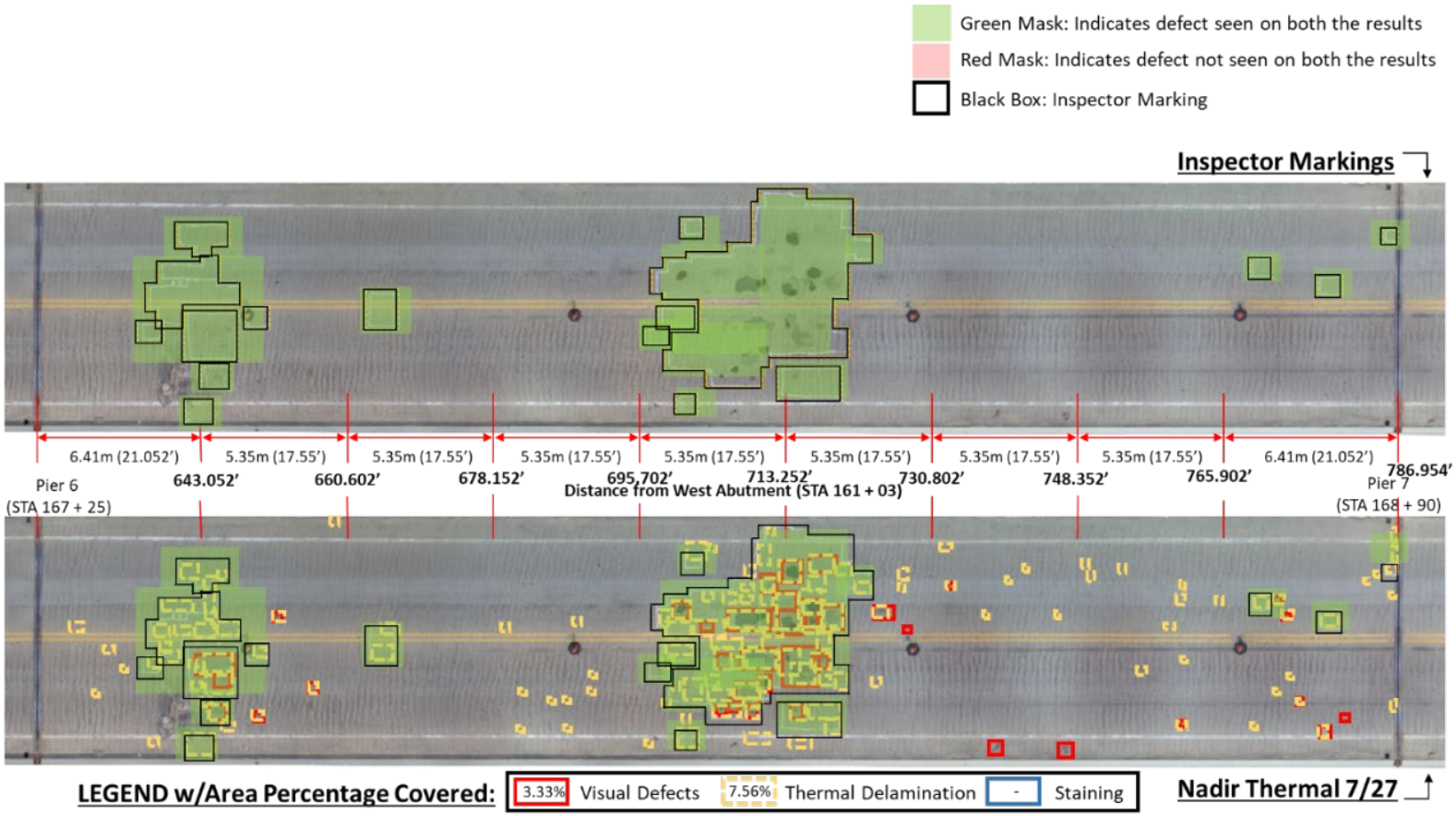


Figure 126: Delamination Marking Comparison of Span 5; Top: Inspector Markings, Bottom: Nadir Thermal Images

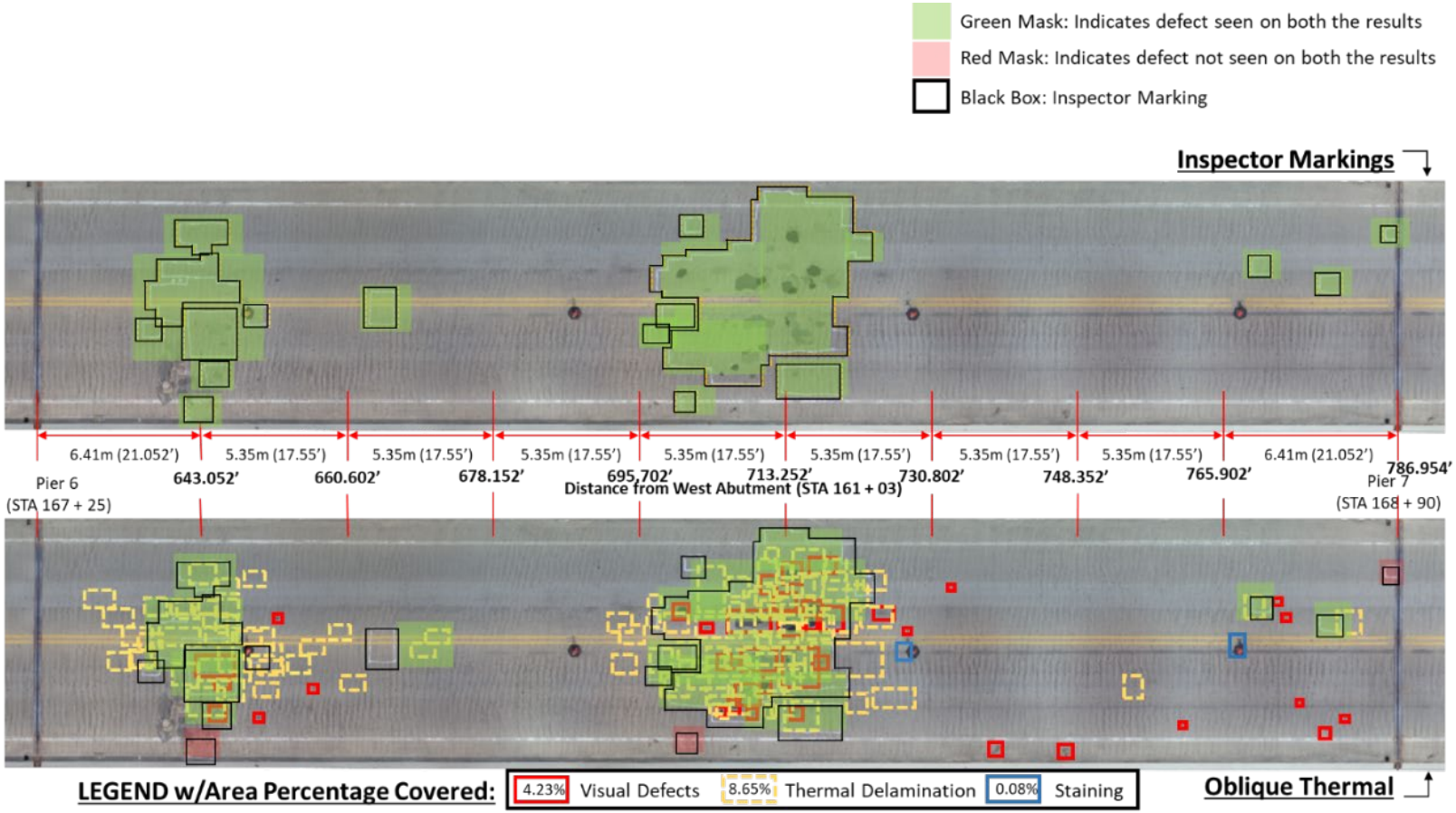


Figure 127: Delamination Marking Comparison of Span 5; Top: Inspector Markings, Bottom: Oblique Thermal Images

Jeremiah Morrow Bridge Segment Modelling: This section presents the case study conducted at Jeremiah Morrow bridge in Oregonia district 6 of the state of Ohio. The objective of this study was to generate high-resolution 3D models of the bridge segments, that would allow inspectors to monitor the cracking on bridge segments. Visual images of the bridge segments were captured at GSDs close to 0.3mm/px to generate high-resolution 3D models.

Alternative to creating these detailed models is to have a bridge inspector overseeing during the flight. This has been shown as seen in Figure 128.

The obtain images with GSDs close to 0.3mm/px the sUAS must be flown close to the bridge, and as it moves closer to a structure the accuracy of the on-board GPS starts to decrease. The decrease in GPS accuracy will affect the positioning of the sUAS midair and as a result, one will not be able to generate automated missions using DJI GS Pro. To obtain the necessary images the sUAS was flown manually. The images of the bridge segments were captured using DJI Matrice 210 sUAS and DJI X5s 45mm camera.

The first-person view (FPV) and the ultrasonic sensor on-board the Matrice 210 helped the pilots to maintain the safe and necessary distance from the structure. Given the height of the bridge (~240 feet) these sensors aided in maintaining the flight perspective during image capture.

Few bridge segments were identified, by an ODOT inspector, to be mapped and are marked in the bridge plan sheet shown in Figure 129. Jeremiah Morrow consists of two bridges, northbound and southbound, and the outer segments of both the bridges were mapped in this study. Among the eight segments that were mapped in this study, one of the segments (P2-3D on the southbound bridge) was inspected and the inspector marked all the cracks visible, which was later used to visually check the accuracy of the generated 3D model. The inscriptions on the segment aided to estimate the required GSD for the missions. The measurement 0.012” was inscribed beside the markings on the segment and was used to decide the distance of flight away from the face of the segment. Since the DJI X5s 45mm camera was used to capture images it was decided to fly at 4m away from the face to obtain a GSD of 0.3mm/px. The ultrasonic sensor on the Matrice 210 sUAS was used to maintain the required distance from the segment. The images were captured in the recommended and pattern and with enough overlaps. Figure 130 shows the positions of the cameras as estimated by Pix4D Mapper.

Figure 128: Bridge Inspectors during UAS Inspection

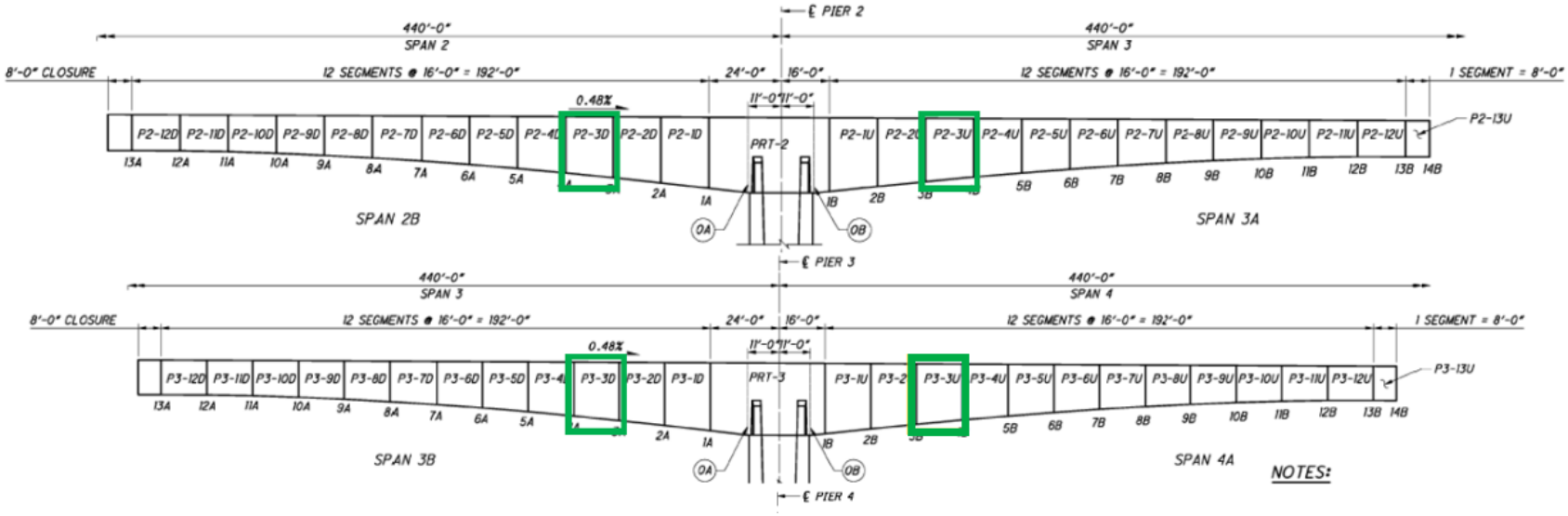


Figure 129: Jeremiah Morrow Bridge Segment Plan Sheets with Mapped Segments Marked

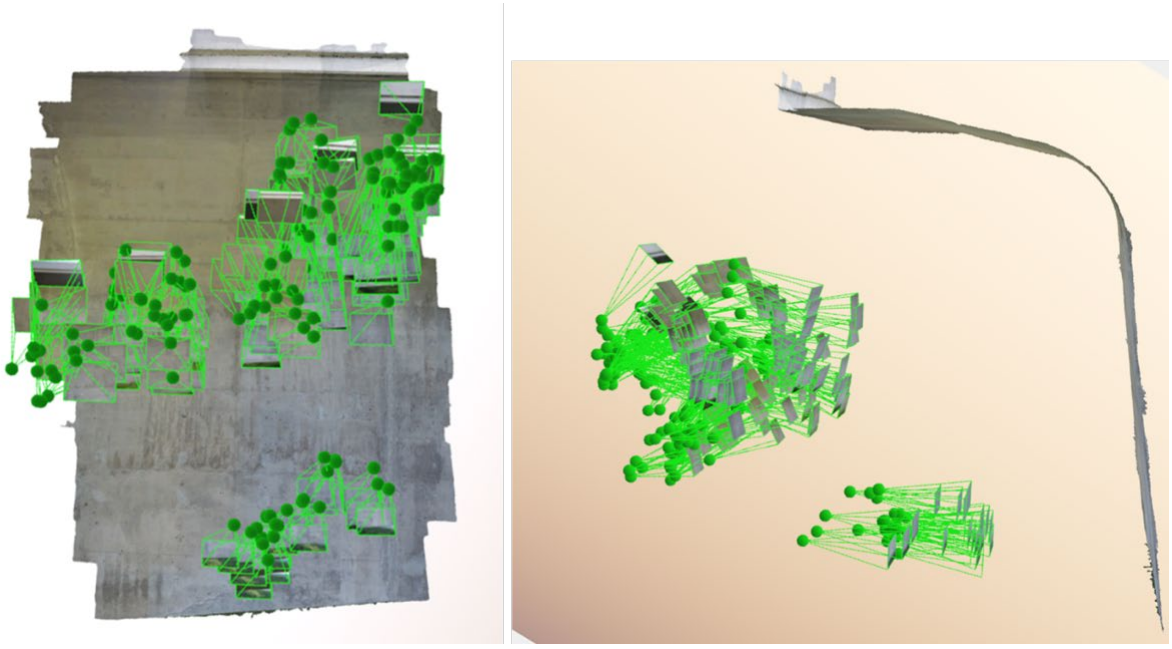


Figure 130: Camera Positions Estimated Using Pix4D Mapper

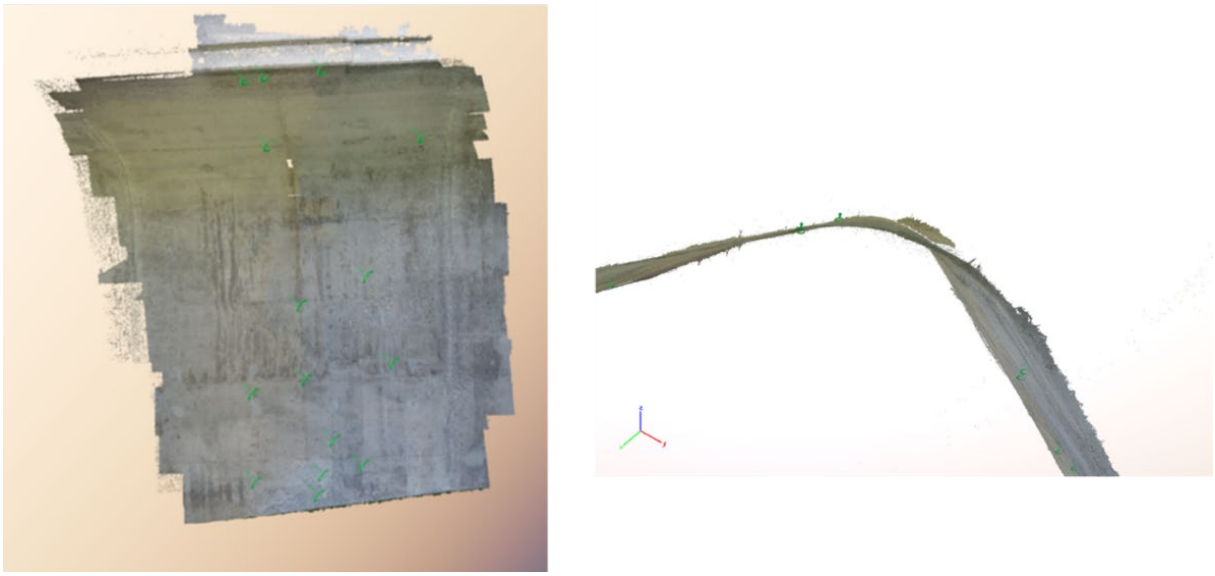


Figure 131: Camera Positions Estimated Using Pix4D Mapper

The captured images were first analyzed before processing and images that were blurry or out of focus were removed from the dataset. This was a necessary step since the image capture process was manual. The images of the P2-3D segment on the southbound bridge were processed first using Pix4D Mapper. The initial model was processed using manual tie points (MTP) and without image geotags. Fifteen MTPs were added uniformly along the corners and the center of the dataset. Figure 131

shows the 3D point cloud that was generated using the fifteen MTPs and it was seen that a few points along the curve of the segment were not correctly positioned. The green points visible in the 3D point cloud indicates the positions of the MTPs. Figure 132 shows the 3D mesh model in the .obj file format.

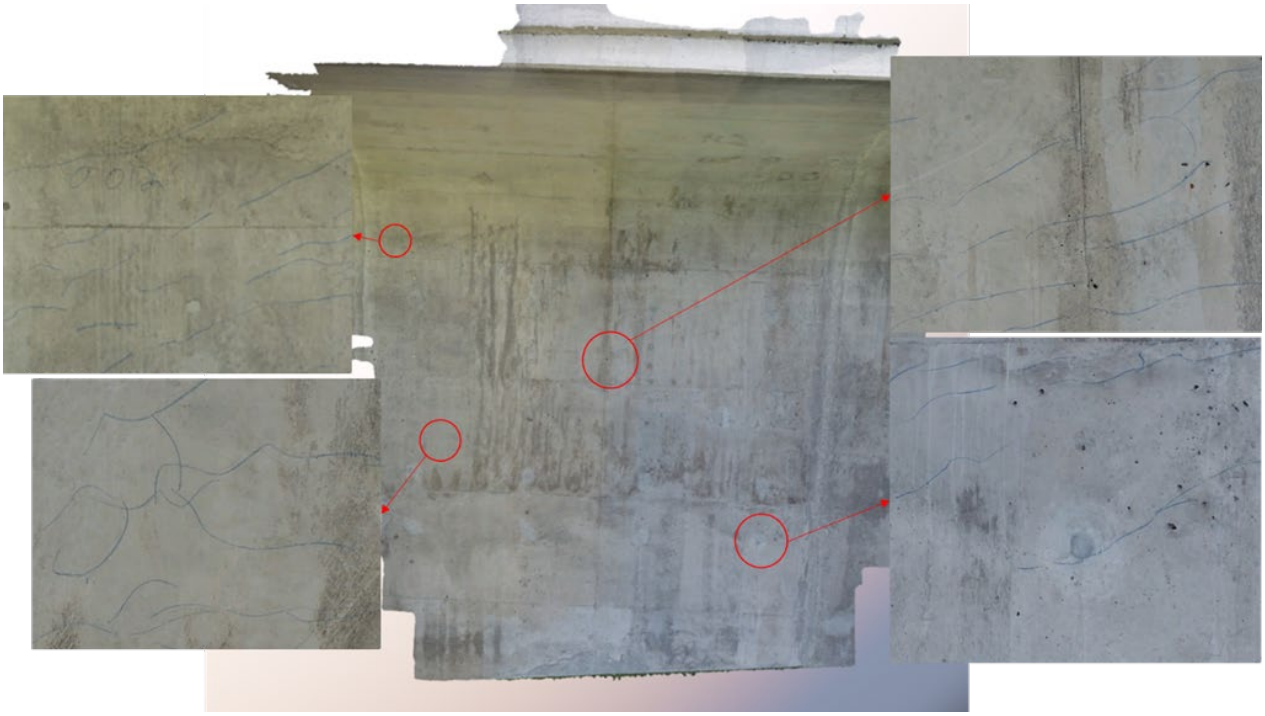


Figure 132: Pix4D Mapper 3D Mesh Model of Segment P2-3D on the Southbound Bridge

For comparison purposes, the image dataset, of segment P2-3D on the southbound bridge, was reprocessed using Context Capture without image geotags and with nineteen MTPs. Figure 133 shows the 3D mesh model generated using Context Capture in the .3mx file format.

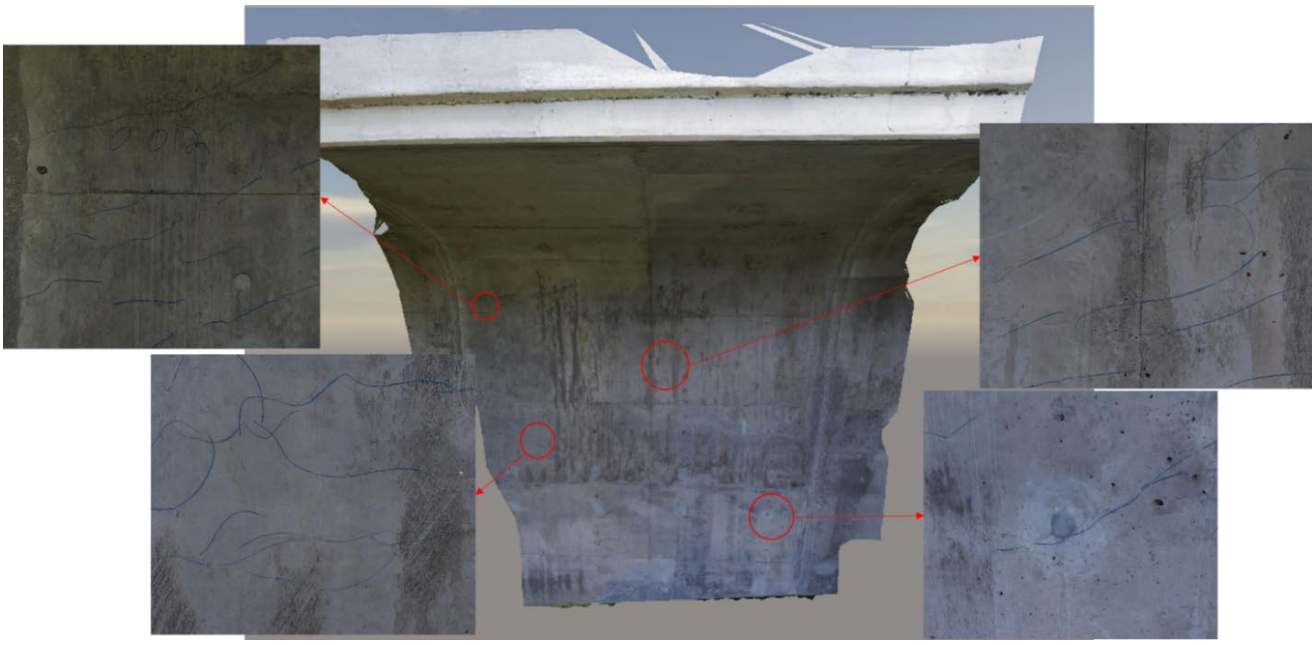


Figure 133: Context Capture 3D Mesh Model of Segment P2-3D on the Southbound Bridge

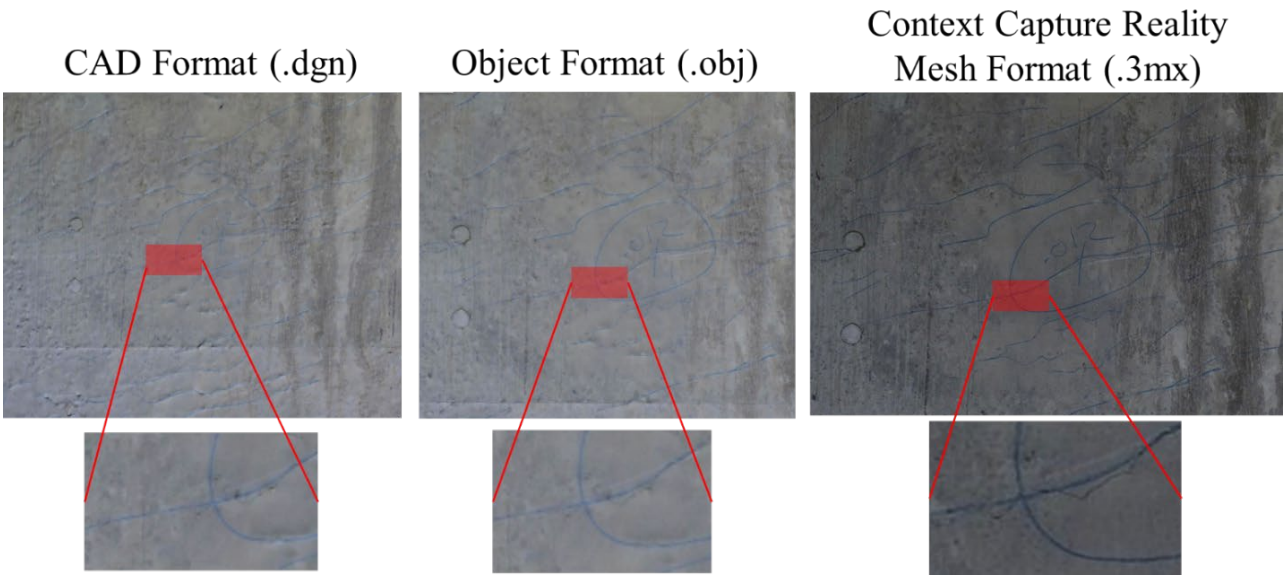


Figure 134: Comparison of 3D Mesh Model in Different File Formats; Left: CAD Format (.dgn), Center: Object Format (.obj), Right: Context Capture Reality Mesh Format (.3mx)

Context Capture was used to generate the 3D mesh model of the segment in three different file formats (obj, .dgn, and .3mx). The quality of the three models was visually compared to find out the best file format (Figure 134). Comparing the models in Figure 134 it is evident that the .3mx file format produces the best 3D mesh model. The CAD format (.dgn), and object format (.obj) files are commonly used and can be viewed using multiple third-party applications. Whereas the Context Capture reality

mesh format (.3mx) is a proprietary file format and can be only viewed using Bentley products.

Analyzing these 3D mesh models, one can easily make out the cracks and given the information inscribed on the segment and the GSD of the images captured the identified cracks have a width greater than and equal to 0.03mm. The cracks are also visible in the orthofacades generated and can be used to conduct future work on automatic crack detection and identification. To aid pilots to decide the combination of camera, the distance away from face and the required GSD a lookup chart (Figure 103) was created for the DJI X5s and DJI Z30 cameras. The chart helps users to decide the height or distance away from the object value based on their selection of required GSD and available camera. The lookup chart was created using the information present on a crack comparator card (Figure 104), a tool used by bridge inspectors to measure the crack widths on bridges. The GSD values that lie above the crack comparator card were blacked out in the chart.

ODOT HQ Roof Inspection: This section presents the case study conducted at the ODOT Headquarters in Columbus, Ohio. The objective of this study was to inspect the roof of the building and attempt to construct a 3D model of the building. The roof was inspected using thermal and visual images where the visual images were used to cross-refer the trouble spots identified using thermal images.

Prior to designing the flight plans for this case study, the building's height was estimated to account for the flight height and image overlap values. Using this information, the missions were planned on DJI GS Pro application and the images of the region were captured using DJI Matrice 100 sUAS and the DJI Z3 visual camera and DJI XTR thermal camera. Images were captured in a grid pattern (parallel and perpendicular) and the mode of capture was set to equal timed mode. Figures 135-136 shows the summary of the missions run at the site. The flight parameters were set to achieve a GSD of 0.8cm/px, and image overlap of 80%, for the visual images, and a GSD of 4.1cm/px and image overlap of 80%, for the thermal images, on top of the roof. Figure 137 shows the screenshot of the visual and thermal images captured, on the GeoSetter application.

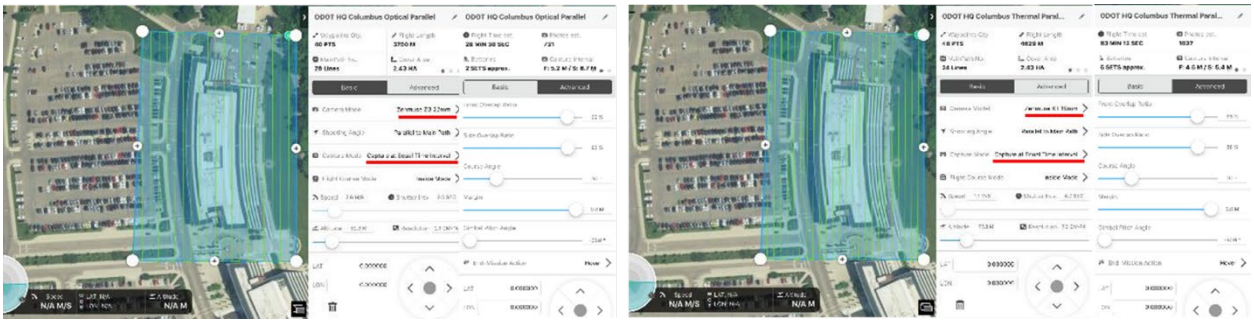


Figure 135: Screenshot of the Missions Conducted at ODOT HQ; Left: Visual Parallel, Right: Thermal Parallel

Image	Mission Type	GSD (cm/px)	Altitude (m)	FO/SO(%)	Flight Time	Recorded Photos
Visual	Parallel	2.06cm/px	52.3m	92%/92%	27min	731
	Perpendicular	2.06cm/px	52.3m	92%/92%	29min	791
Thermal	Parallel	6.96cm/px	77.82m	88%/88%	1hr 25min	1037
	Perpendicular	6.96cm/px	77.82m	88%/88%	1hr 25min	1038

Figure 136: Summary of Missions Conducted at ODOT Headquarters

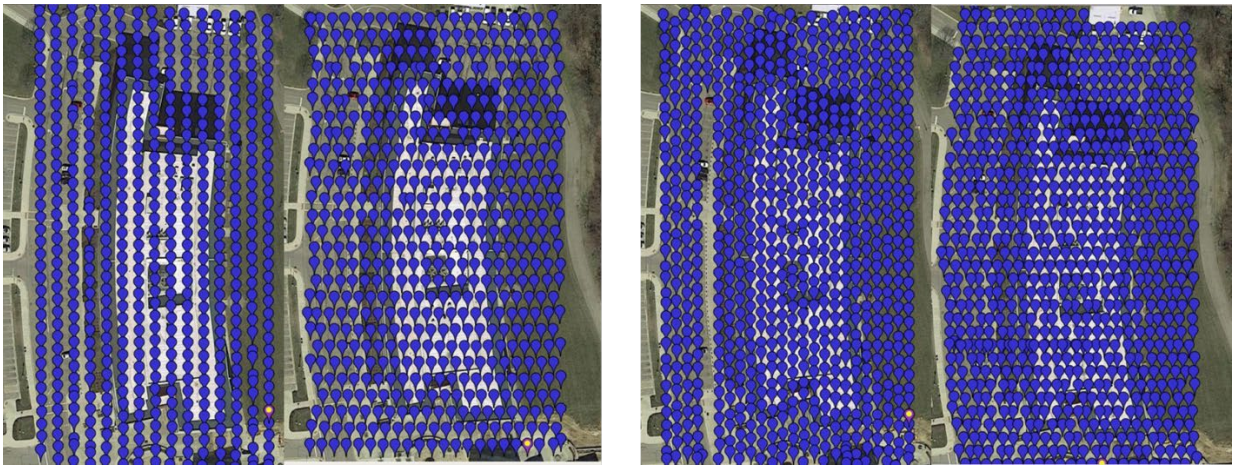


Figure 137: Images Captured for the ODOT HQ Case Study; Left: Visual Images, Right: Thermal Images

Figure 138: 3D Views of the 3D Point Cloud Generated Using the Visual Images of ODOT HQ

The images captured were processed separately, using Pix4D Mapper. 3D point clouds and 2D orthomosaics using the visual and thermal images were generated without ground control points (Figure 138). The visual and thermal orthomosaics were cross-referenced to identify possible problem spots. Figure 139 shows the outputs generated using the visual images. Figures 140 shows the outputs generated using the thermal images.

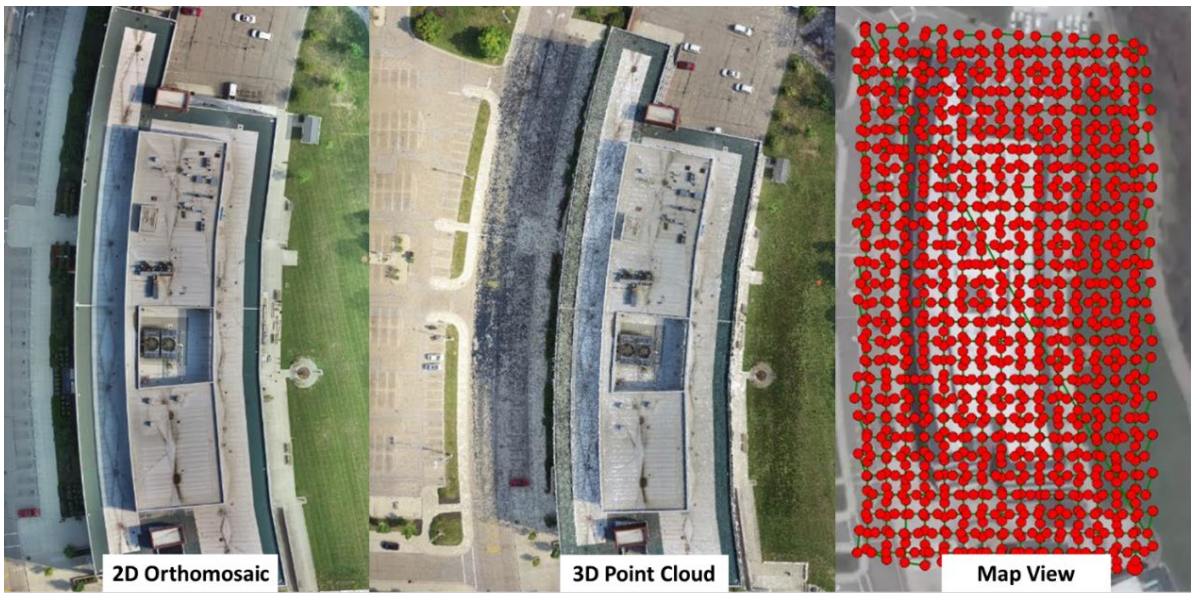


Figure 139: Top View of the Outputs Generated on Pix4D Mapper Using the Visual Images; Left: 2D Orthomosaic, Center: 3D Point Cloud, Right: Map View on Pix4D Mapper

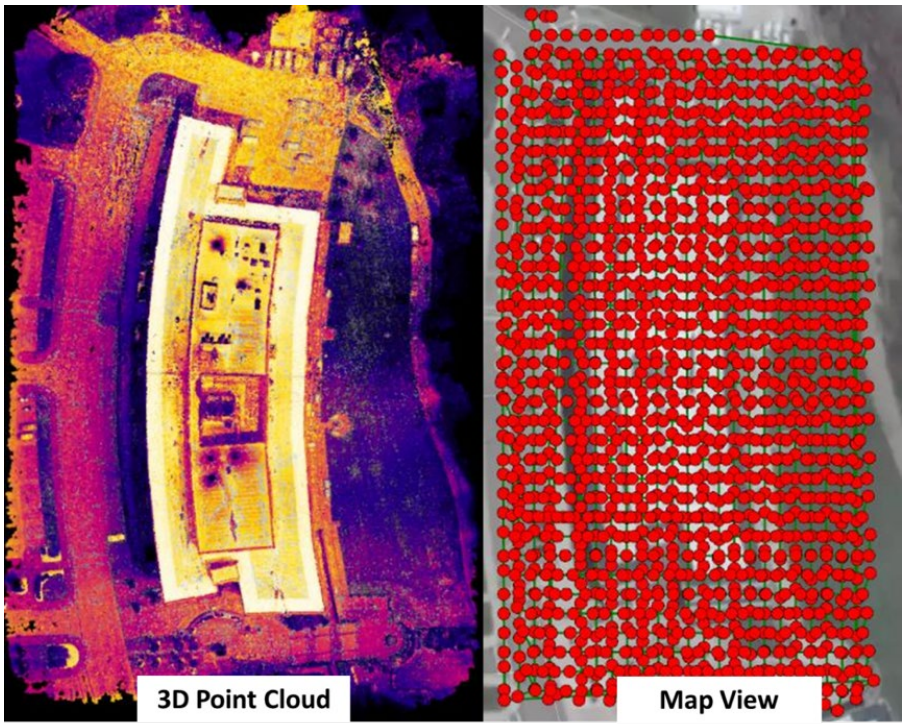


Figure 140: Top View of the Outputs Generated on Pix4D Mapper Using the Thermal Images; Left: 3D Point Cloud, Right: Map View on Pix4D Mapper

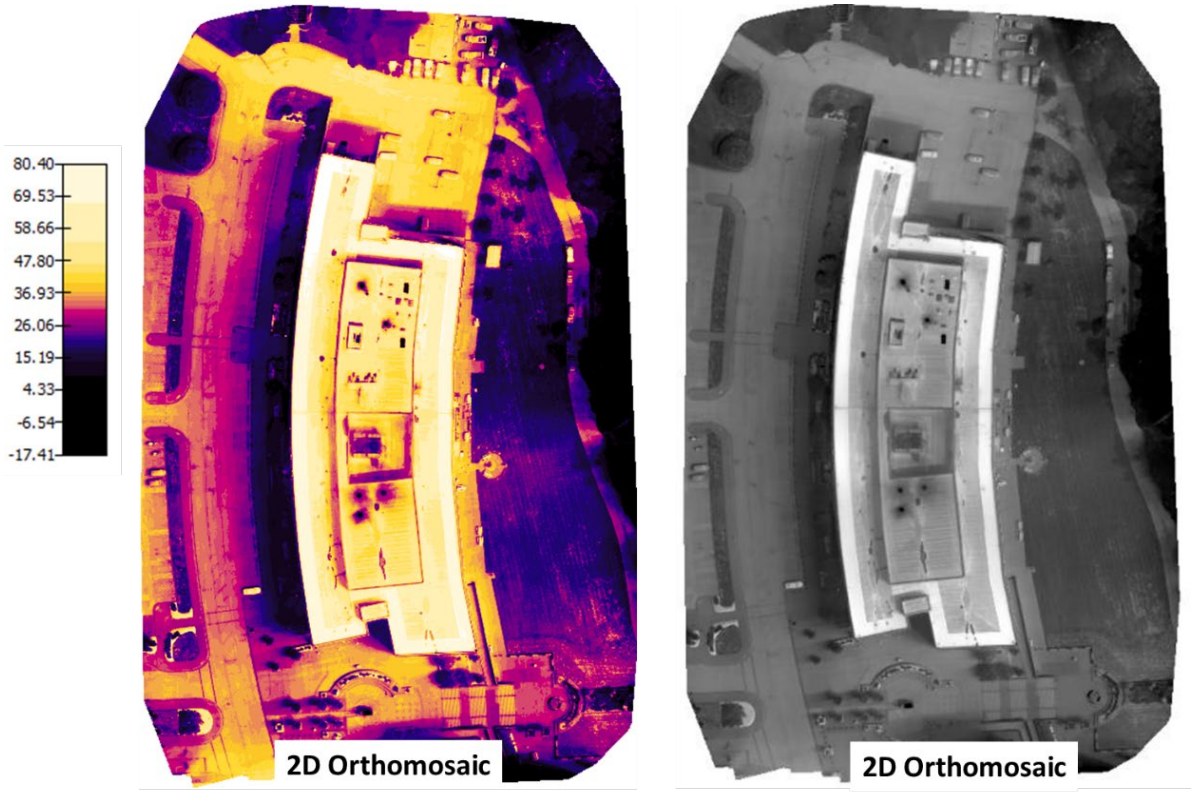


Figure 141: Top View of the Outputs Generated on Pix4D Mapper Using the Thermal Images; Left: 2D Orthomosaic with Reflectance Map, Right: 2D Orthomosaic without Reflectance Map

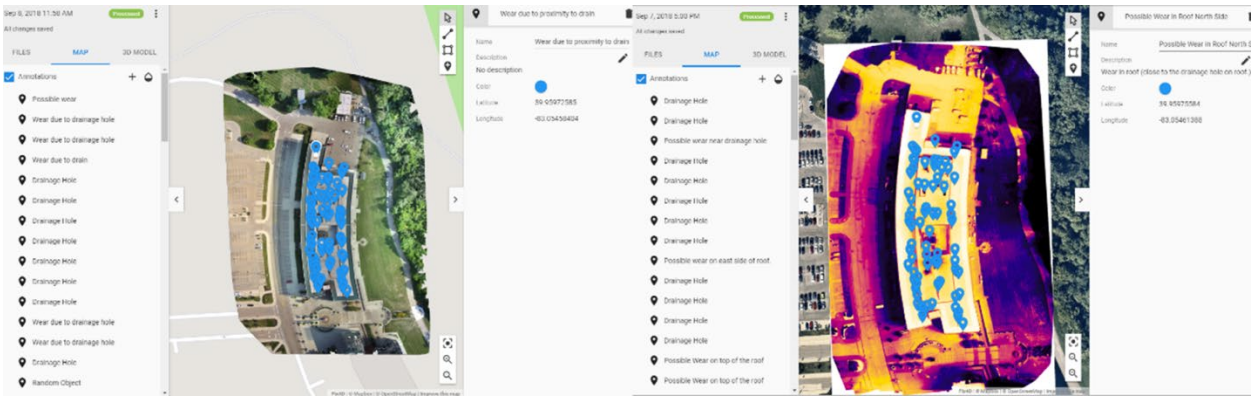
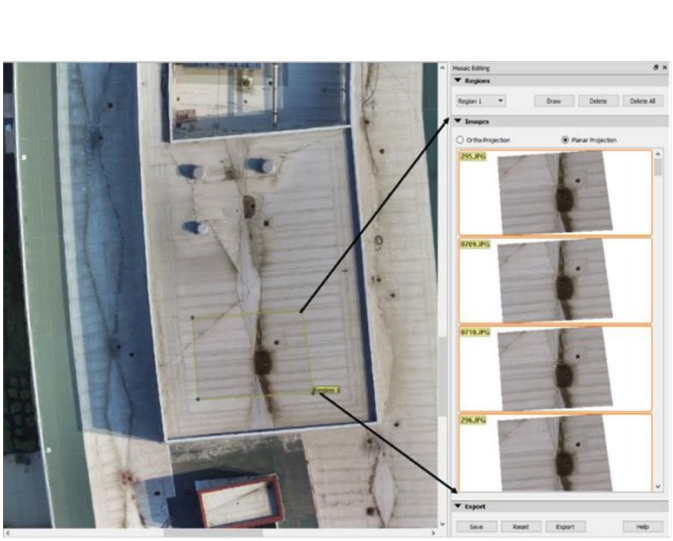
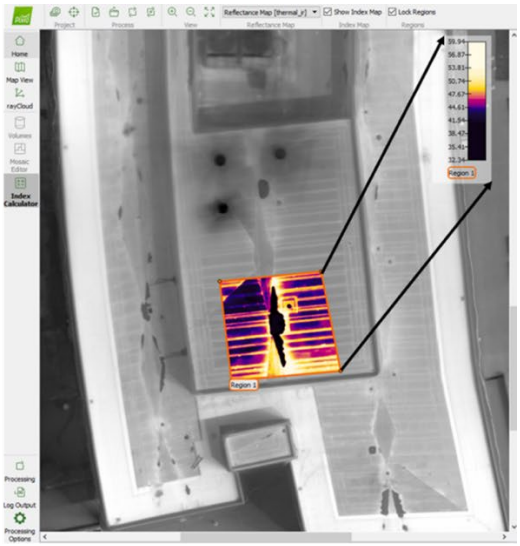


Figure 142: Visual and Thermal Orthomosaics Annotated Using Pix4D Cloud

Analyzing the thermal orthomosaic it was observed the temperatures recorded varied greatly. This was mainly due to the presence of various objects, in the region mapped, that had a wide range of emissivity values. The thermal and visual orthomosaics were compared side by side (Figure 141) and trouble spots were identified and annotated on Pix4D Cloud (Figure 142). Pix4D Mapper was also used to conduct an in-depth analysis where regions of interest were drawn on the orthomosaic and the individual images of the area were extracted or the temperature bar on Pix4D was constrained to the temperature recorded in the region (e.g., Figure 143).





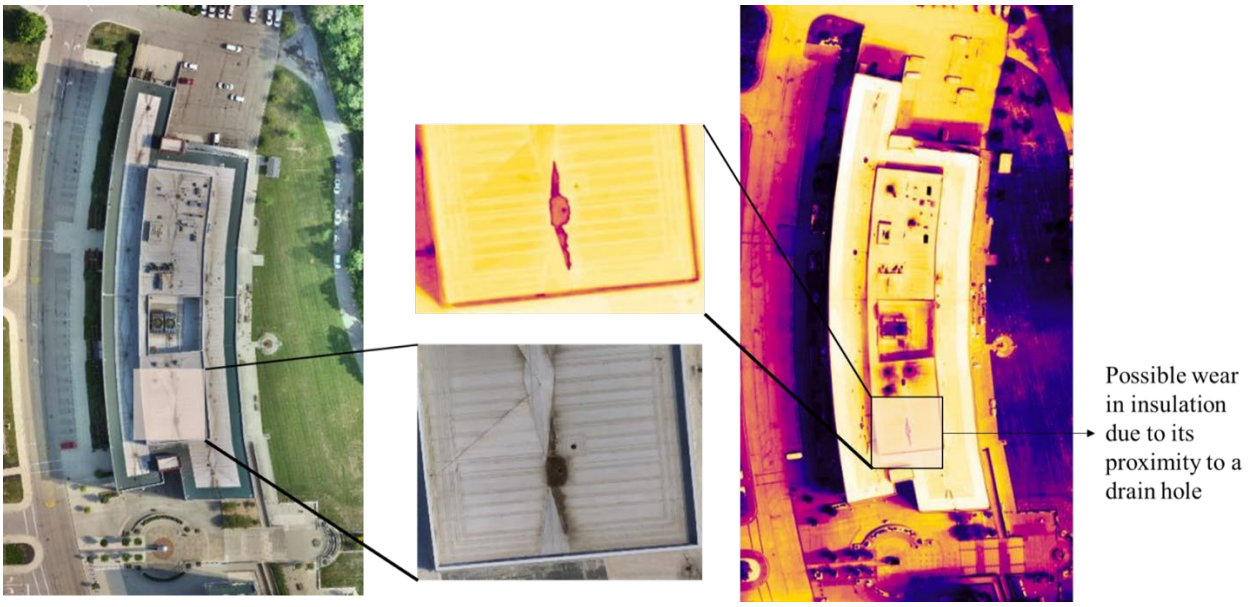


Figure 143: Cross Referring Visual and Thermal Images to Identify Problem Spots on the Roof

Using Flyability Elios for Bridge Inspections: This section presents the case study for using the Flyability Elios V1 for bridge inspections. The objective of this study was to use a small collision tolerant UAS to aid in the inspection of expansion joints, cracking on steel girders, and hangers. The results obtained will show how this UAS can aid inspectors to see hard to reach areas of the structure.

The Flyability Elios [22] is a small UAS that is protected by an outer cage. This cage is designed to allow the vehicle to do up close inspections without crashing. Originally intended for inspections of confined spaces it has also proved to be quite capable as a structure's inspection tool (see Figure 9).

The Elios has a visual camera, a thermal camera, and onboard lighting making it suitable for low light conditions as well as for up close inspections.

The Elios is based off a DJI autopilot but does not include GPS or other positioning control so may be difficult for even experienced pilots. Specific Elios training is required to operate and will not be covered in this SOP.

Figure 144: Bridge markup

It is recommended to use the Elios in a dual operator mode, with one person flying the vehicle and the other controlling the camera for inspection. Due to the design of the Elios this may prove difficult as it is easy to lose orientation. Communication between pilot and inspector during flight is important and will help prevent any unnecessary time wasted.

To help facilitate this communication it is important that both pilot and inspector be familiar with the ODOT Manual of Bridge Inspection [31]. This manual covers the nomenclature on bridges and helps both pilot and inspector orient themselves to the structure. As a reference Figure 144 is an example of a markup on an inspected area.

This particular bridge is a N-S highway; therefore, the red numbers are counting the beams from left to right looking upstation. The cross-braces are numbered from south to north after each pier.

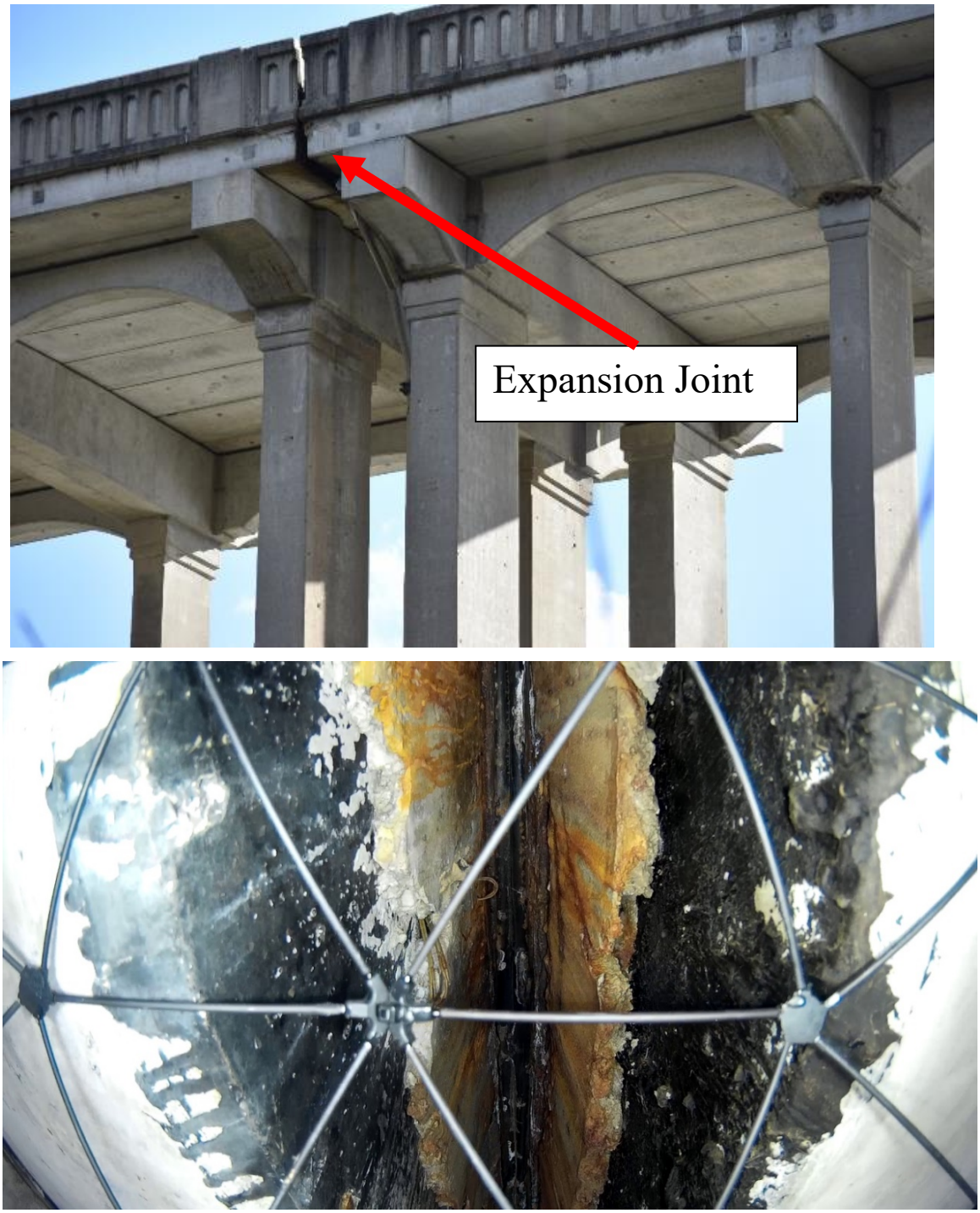


Figure 145: Bridge expansion joint and Elios close-up view

One example of using the Elios is the inspection of expansion joints. One such example would be the bridge shown below in Figure 145. This expansion joint will be difficult for an inspector to reach from either the ground or a snooper. Using the Elios, the resulting image was taken. It is important to adjust the exposure and lighting of the Elios as oversaturating the image is possible.

Figure 146: Inspection of Stress Crack

Another example use for the Elios would be the inspection of stress cracks on beams. The marked areas in purple in Figure 144 show locations of such areas to inspect. For this example, we are looking at beam 5, cross-brace 5. It is important to record which section you are inspecting and follow a standard pattern for each inspection. In other words, we would want to keep it consistent that we are inspecting the south-west, north-west, north-east, and south-east corners of beam 5, cross-brace 5. Figure 146 shows an example of this area.

DJI Matrice 210 RTK V1 in GPS Denied Environments for Bridge Inspection: This section presents the case study for using the DJI Matrice 210 RTK V1 in a GPS-denied environment. The objective of this study was to use the Matrice 210 for inspection of bridges where you will receive none or limited GPS. Most of this utilizes the Matrice 210's upward gimbal for inspecting the sides of bridges (see Figure 8). The bridge in questions was the new Jeremiah Morrow Bridge located along IR-71 over the Little Miami River in Lebanon, OH.

When operations like this that do not have GPS coverage and are flying close to a structure it is good practice to utilize dual operators. In this case use one person to fly the vehicle and the other to control the camera.

Figures 128, 144, and 145 are examples of areas where you would want to have the upward facing gimbal. These structures have a large overhang that the UAS would be flying under. It requires coordination between the pilot and the inspector to capture these images. On the example shown in Figure 128 the pilot is keeping the

UAS in one location approximately 15 feet away from the structure. While the vehicle is stationary the inspector tilts the camera up ensuring that overlap is considered. This completes one column pass, the pilot then moves the vehicle over so still getting overlap with the last column and repeats the process. This can take 10 or more passes to accomplish. To create this model, it took approximately 500 photos and 40 minutes.

A few items to note is because there is no position information the vehicle will rise and lower so the inspector must keep track of overlap between the pictures. The vehicle will also yaw during the mission so at the completion of each column the pilot must check the yaw before proceeding.

Figure 147: Sample 3D Model taken by Upward Facing Gimbal

The model shown in Figure 147 is a good representation of a model that is created in this method. Figure 148 shows the number of photos seeing the scene, taken from the ContextCapture Quality Report [20]. The colors indicate the number of photos that potentially see each area. As you can see the areas on the edge of the

model are less covered and therefore it is imperative to cover the adjacent segments that you are interested to get the most area covered in the 3D model. It can also be seen in Figure 148 that more than 50 photos cover the center areas. The communication between pilot and inspector is important to prevent the areas to be lacking.

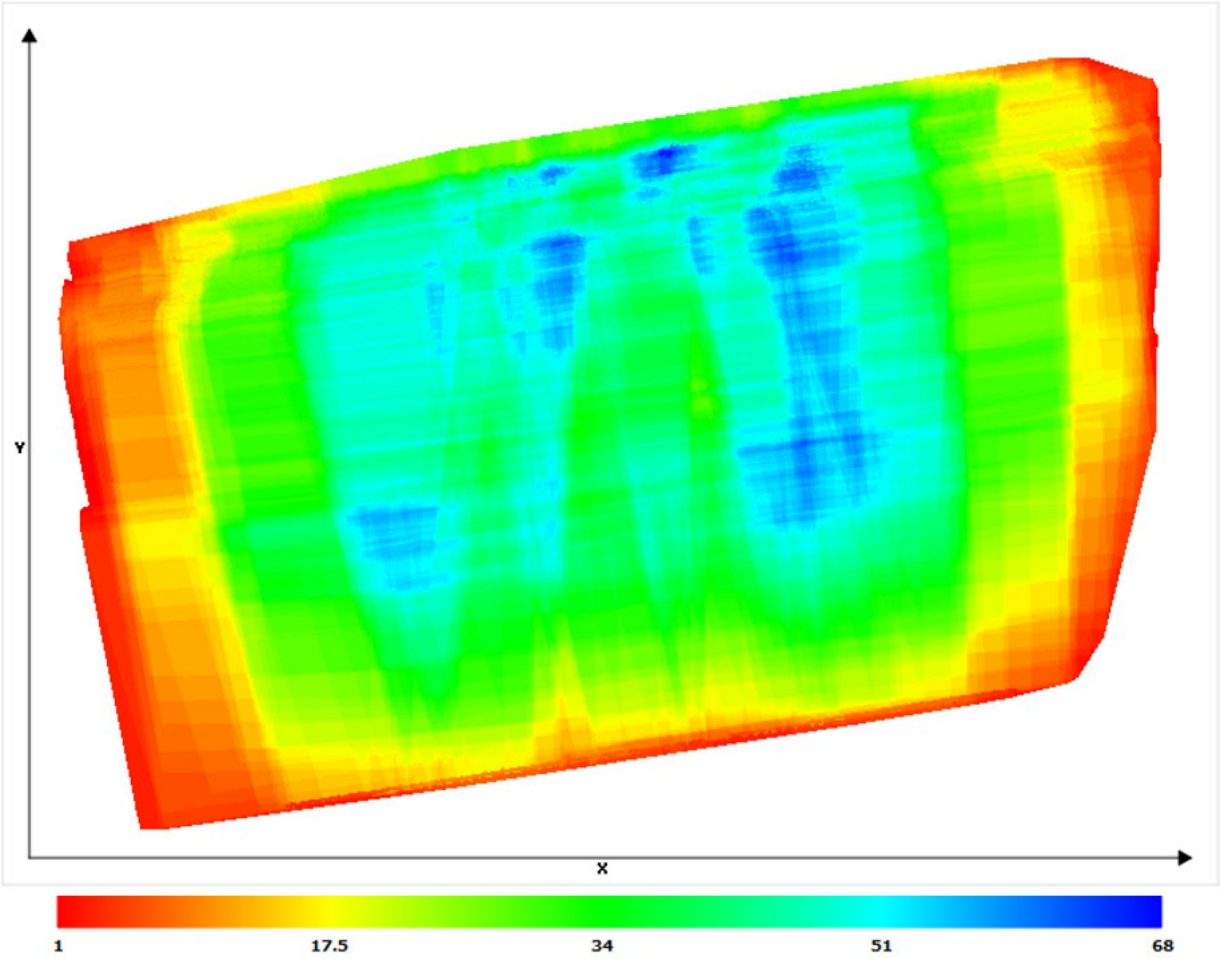


Figure 148: Number of photos seeing the scene: Top view (XY plane) display of the scene, with colors indicating the number of photos that potentially see each area

3.5 Common Operating Platform

As can be seen from the discussions above in Sections 3.3 and 3.4, the use of UAVs as airborne platforms for 2D and 3D modelling is a data-intensive process. The need for a UAV-independent computing platform that could accept imagery and manage the modeling and visualization process was also identified early on in this

project. The result was the proof of concept system developed and deployed on server systems both at our university labs and on servers within the ODOT IT ecosystem. The development of this system, called the Common Operating System, is reported on in this section.

The modern decision-making process is a data-centric environment. Data enables informed decision making based on facts, statistics etc. Therefore, the processing of data has become an integral part of the modern workspace. Users may leverage several tools and computational machinery for their everyday needs. But sometimes the computation required maybe intensive requiring more powerful computers. Or it may involve the use of software that is hard to install and maintain. This work describes a centralized platform that runs computational workflows for users on the cloud. The system consists of a frontend user interface where users can interact and schedule projects for the system to process and a backend framework that carries out the actual computational processing. Such a system would have to be able to support multiple users, scale depending on the workload and above all process data. In this work, it will be shown how such a system can be used to process 3D models from images acquired by flying UAVs over structures or for aerial mapping operations (e.g., construction monitoring, etc.) using proprietary and open-source software.

The results reported here borrow heavily from the SOPs developed by the research team for use by Ohio UAS Center pilots and personnel as part of this project [36] as well as the theses and papers generated by the various graduate students involved in this research project [33,34,35]. The reader is referred to these documents for additional information and details.

3.5.1 Multi-User, Multi-Server Based Approach

The Common Operating Platform (COP) is a dynamic web application created with the purpose to have a common interface for processing various image processing workflows that need expensive computer hardware. The services provided by it are as follows:

- Process uploaded projects consisting of aerial imagery to produce densified point-cloud-based 3D models.
- View densified point-cloud-based 3D models on the application using a 3D model viewing software package Potree [10].
- Upload, store and share multiple les up to 4GB in a given instance of time.

Initially, such a system was developed and tested at UC and later at the UAS Center on a single server system. Even though the application served its purpose, there were some shortcomings as it was based on a single server system architecture discussed below:

- The architecture of the platform is designed in such a way that not all components are scalable and hence it cannot be scaled to a multi-server based system if the need arises.
- After server maintenance is performed on the servers running the application, it does not restart automatically. It needs to be restarted manually by logging on to the servers.
- The upload limit of 4GB makes it difficult to process models consisting of project datasets greater than 4GB even though the application is capable of processing such models.
- Files deleted from the application interface remain present on the file storage of the application.
- There is no tool to log the server metrics to monitor high system usage on the servers.

Distributed computing has been significantly improved with the technological strides that have occurred. The advancements made in cloud computing has improved the latency and scalability of the infrastructure used in various distributed systems. The advent of container-based and orchestration-based technologies has made it easier to package the application code and maintain the entire application as it provides a layer of isolation and keeps the application code environment consistent. Since these advancements have been made quite recently, there is a technological gap between applications observed today, and the ones observed prior to the advancements made. It is necessary to upgrade the older applications with these new technologies as it will improve the maintenance and scalability and decrease the cost of buying expensive hardware.

In order to make the system multiuser compatible and to resolve the previously discussed limitations, the following steps are performed:

- The architectural changes are brought about by the addition of container orchestration tool Docker Swarm, cloud storage Azure File Share and Oracle Database. This will allow the system to be a distributed cloud-based system that will be highly available, multi-server system compatible and easily scalable.
- The use of container orchestration tool Docker Swarm along with the automation server tool Jenkins [11] allows the application to be automatically restarted in the event of a failure occurring.
- Integrating cloud storage based tool Azure le share and making changes to configuration les will help in increasing the upload limit.

- The addition of features like delete automation will help improve the performance of the application and help delete redundant les on the le storage.
- Server metric logging will help monitor the application throughout its lifecycle.

3.5.2 Software System Architecture

The development of COP started on a high-end hardware at UCII, University of Cincinnati. Post the rigorous tests and benchmarking, the working version of Common Operating Platform is moved to ODOT IT's ecosystem and is currently live on single server environment called Proof-Of-Concept (POC) server. The COP architecture consists of mainly two components (see Figure 149):

- A web interface which enables users to upload and interact data to the system
- A backend processing system responsible for processing uploaded data

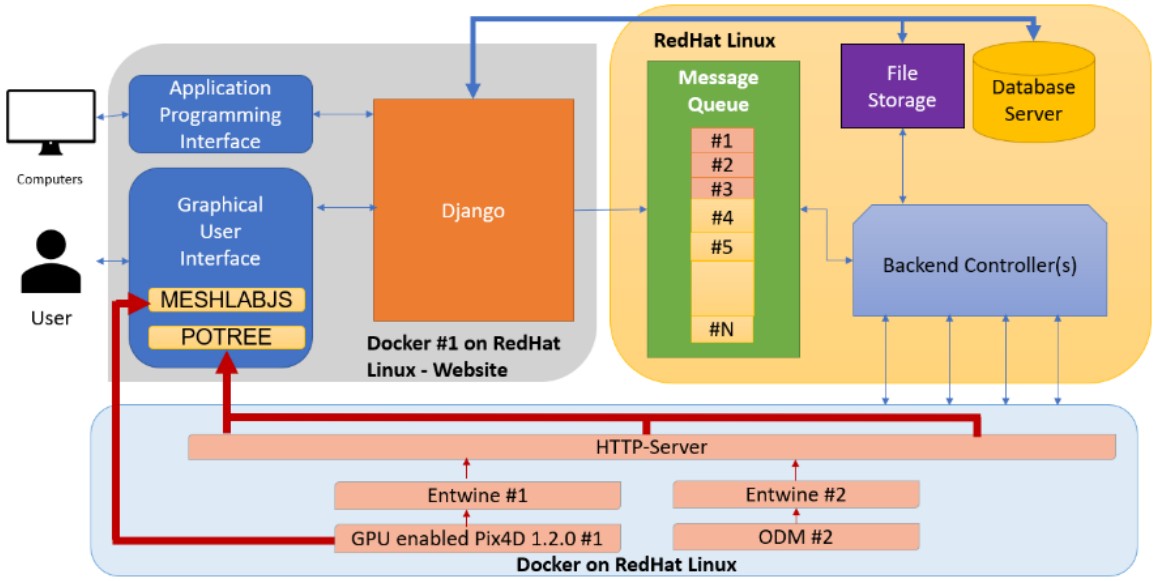


Figure 149: COP Single-Sever Architecture

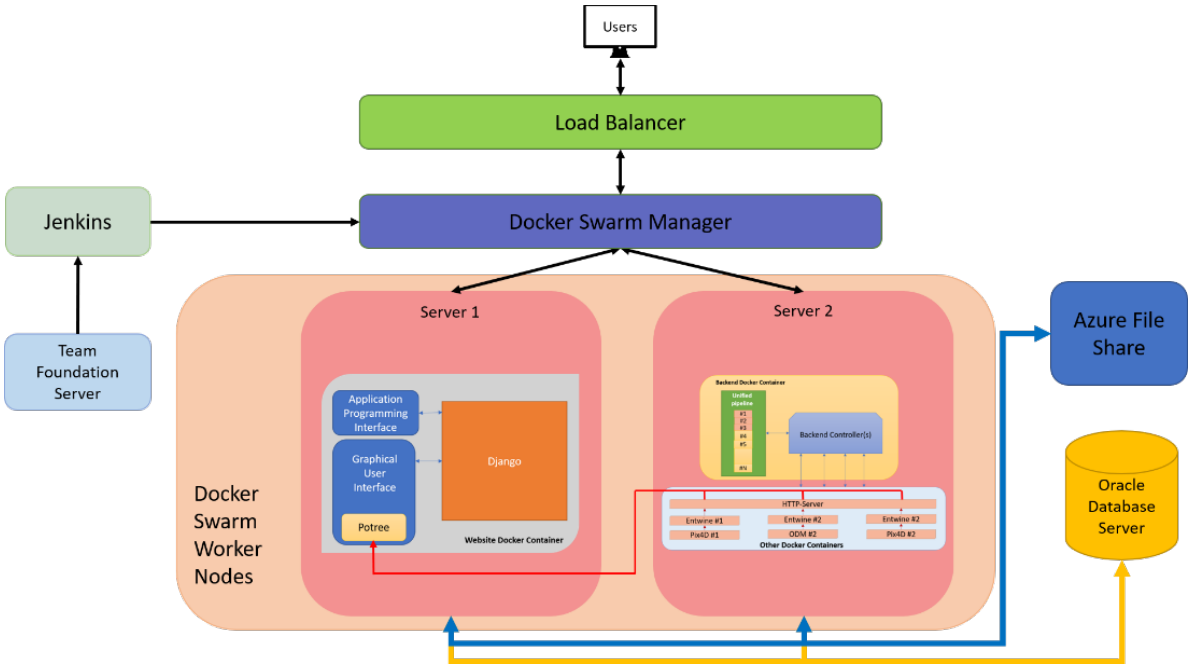


Figure 150: COP Multi-Server Architecture

The website is created using a Django backend with enterprise edition Oracle database. The website interface is written in HTML, CSS, and JavaScript using Twitter's Bootstrap frontend interface library. J-Query and Ajax are used for implementing website functionality. The COP also facilitates API programming. HTTP-based REST APIs have been used in tandem with the Microsoft HoloLens for visualizing 3D models (processed by the backend).

The backend processing system is implemented using a combination of Python, and Bash scripts. It is responsible for serving the website, the APIs, and the processing of the 3D models. For the processing of 3D models, COP uses Pix4Dengine 1.2.0, a proprietary 3D model processor and OpenDroneMap, which is open source. The backend system was designed to contain multiple blocks which can run independently of each other. Together they create a workflow of numerous independent systems that work together towards a common objective.

The previous system architecture of COP is designed to run on a single-server as all the components are not scalable. Also, there are limitations to the size of files uploaded and deletion of files on the file storage. To resolve these issues and make the application multi-server compatible, certain architectural changes were needed (see Figure 150). While working on single server architecture, UC extensively researched on improving the application scalability for users to process multiple projects simultaneously. For this purpose, development work was done on two different servers, named DEV and TEST, representing development and testing environments for the application. To achieve this, a distributed cloud system-based architecture with advanced container orchestration and cloud storage approach is used. The system backend is containerized using Docker. These containers are

orchestrated and scaled using container orchestration tool called Docker Swarm. As the application requires high storage, cloud-based Azure File Share is used to store processed projects and various system logs. With this configuration, the deployment of the COP is easy and is automated using CI/CD based tool called Jenkins. For technical details, please refer to the SOP for a Common Operating Platform for Online 3D Modeling - v11.0. For details on how to use the platform, please refer to SOP for a Common Operating Platform for Online 3D Modeling – User Guide.

3.5.3 System Implementation and Testing

As seen in Figures 149-150, the Common Operating system contains several components each of which was configured and tested individually and collectively as described here.

Pix4Dengine Server: COP uses Pix4Dengine server from the commercial, Pix4D suite of applications. The Pix4Dengine server is a set of programming modules developed to generate 2D and 3D models on a cloud infrastructure. The Pix4Dengine can be easily integrated with enterprise applications using its Software Development Kit (SDK). It does not come with a Graphical User Interface (GUI) and the users often develop a custom interface to build their platform. The Pix4Dengine produces three output models - 3D Densified Point Clouds, 3D Textured Mesh, and 2D Orthomosaic.

Pix4Dengine 0.1.0 was used until COP v2.0. The Pix4Dengine 0.1.0 is the first release of the Pix4Dengine. Pix4D has released various advanced versions of this software and it no longer supports 0.1.0. Hence, it was essential to upgrade the engine to enable processing workflows with and without the Ground Control Points (GCP). UC conducted a detailed study of Pix4Dengine 0.1.0 and 1.2.0. and identified that the new engine comes with optimized processing algorithms and more options to tune these algorithms. The Pix4Dengine 1.2.0 optimizes the use of available resources such as RAM, disk memory etc., and thus, cramps up less memory than the earlier version.

UC developed a custom algorithm to allow users select Pix4Dengine 1.2.0 or ODM using the COP website. This algorithm forks into two pipelines, processing with and without the GCPs, based on the inputs given by the user (refer to Figure151). Please note that the processing pipelines refer to the options available to process with the Pix4Dengine 1.2.0. These pipelines should not be confused with the workflows on the COP.

UC developed another algorithm to process projects with and without GCPs using Pix4Dengine 1.2.0 as shown in Figure 152. To begin processing, the user is required to create a special Pix4D project using the web interface. Once the project is created, the user must decide whether he wants to engage the GCPs in the processing. To process without GCPs, the user is simply required to upload images to the project and submit it for processing. However, processing with the GCPs is a complex task, where the user is required to preprocess the project on Pix4Dmapper and then submit it to the Common Operating Platform for further processing. For now, let's assume that the pre-processing is done. The user should create a Pix4D

project on the COP and upload images and 2 input files. One of these files should contain information on GCPs and the coordinate systems to be used, and the other should contain information on the image marks. Based on these inputs, the algorithm identifies whether it should process with or without the GCPs.

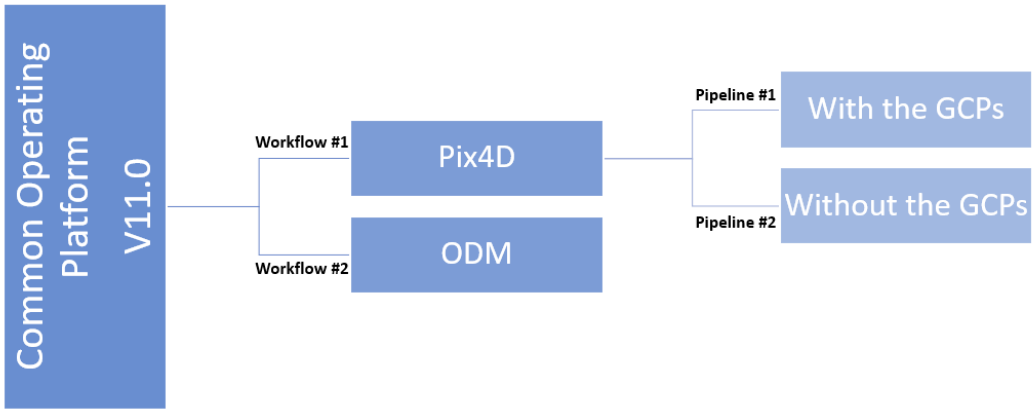


Figure 151: Algorithm to identify Pix4Dengine processing pipeline

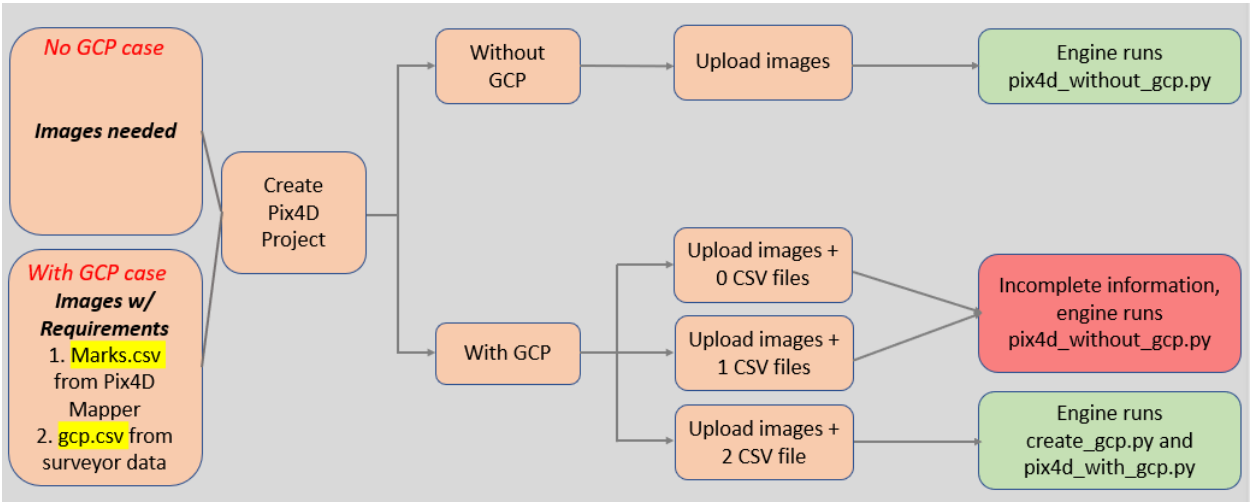


Figure 152: Algorithm to identify Pix4Dengine processing pipeline

Open Drone Map (ODM): OpenDroneMap is an open-source photogrammetry toolkit to process aerial imagery into maps and 3D models. UC pulled the ODM image from Docker hub and plugged it with the COP. ODM can process 3D Point Clouds and 3D Textured Mesh models.

Benchmarks for Pix4Dengine 1.2.0 and ODM: To benchmark the capabilities of the COP, tests were conducted on two different servers running the exact copies of the software. One of them was a production class Microsoft Azure server provided sponsored by ODOT and the other was a powerful desktop Alienware PC. The

specifications of the servers as given in table. Two image datasets were compiled for the tests. The first one titled 'small project' consisted of 46 images and the second one titled 'medium project' consisted of 199 images. They were processed using Pix4Dengine and OpenDroneMap on both servers. Figures 153-154 show the results of the testing. The more powerful ODOT RHEL Server outperformed the Alienware server on the 'small project' but the results are comparable when it comes to the larger 'medium dataset'.

	ODOT RHEL Server	**Alienware Server**
CPU(s)	16	12
Memory	125 GB	16 GB
Operating System	RedHat Enterprise Linux 7	Ubuntu 18.04 LTS

Figure 153: Pix4D/ODM hardware test configuration

Server	**Component**	**Small Project**	**Medium Project**
ODOT RHEL Server	Pix4Dengine	15m	1hr
ODOT RHEL Server	ODM	15m	1hr
Alienware Server	Pix4Dengine	30m	1hr 5m
Alienware Server	ODM	30m	1hr

Figure 154: Pix4D/ODM processing test results

Pix4Dengine 1.2.0 is also tested for its performance for projects with and without the GCPs. Three sets of projects - small, medium, and large, were successfully processed using engine on the ODOT server. Figure 155 shows the test results for projects run with and without the GCPs. The results show that the Pix4Dengine 1.2.0 can smoothly process projects with at least 1000 images and 17 GCPs

GPU-enabled 3D model processing: Graphical Processing Unit or GPU is a special circuit designed to process time-intensive tasks. GPU comes with high memory and advanced parallel computing techniques that can handle thousands of tasks at any given time. GPU plays a very important role in 3D modeling. Since 3D modeling is time and resource-intensive, it becomes essential to use the power of GPU for faster processing. While generating 3D models, the time required to process a project

depends highly on the number of images, image size, image content, and processing options selected. As a result, GPU handles each project differently and the performance varies because of these differences.

Project Size	#images	Without GCPs	With GCPs
Small	289	2hr 45m	3hr 45m
Medium	629	4hr 25m	4hr 35m
Large	961	8hr 20m	9hr 20m

Figure 155: Pix4DEngine performance benchmark testing with and without GCPs

GPU is used at different stages of processing by Pix4Dengine. Once CUDA-enabled NVIDIA GPU is installed, the Pix4Dengine 1.2.0 automatically detects and uses it for processing. COP on POC uses NVIDIA Corporation GP100GL [Tesla P100 PCIe] along with CUDA packages on RedHat Enterprise Linux 7.

COP Benchmarking on NVIDIA Corporation GP100GL [Tesla P100 PCIe] GPU: The performance of the Pix4Dengine 1.2.0 with and without the use of GPU is benchmarked on different number of CPU cores. The test project was small and contained 67 images. The results are shown in Figures 156-157. It can be concluded that the processing speed increases by 2x when using a GPU. It is also evident that the as the number of CPU cores increases, processing time decreases. Thus, total time taken by the Pix4Dengine to process a task is highly dependent on the number of CPU cores as well as the GPU used.

CPU Cores	Time taken with GPU	Time taken without GPU
4	1hr 04m	1hr 30m
8	26m	50m
12	19m	38m

Figure 156: Performance analysis of the Pix4Dengine with and without the use of GPU

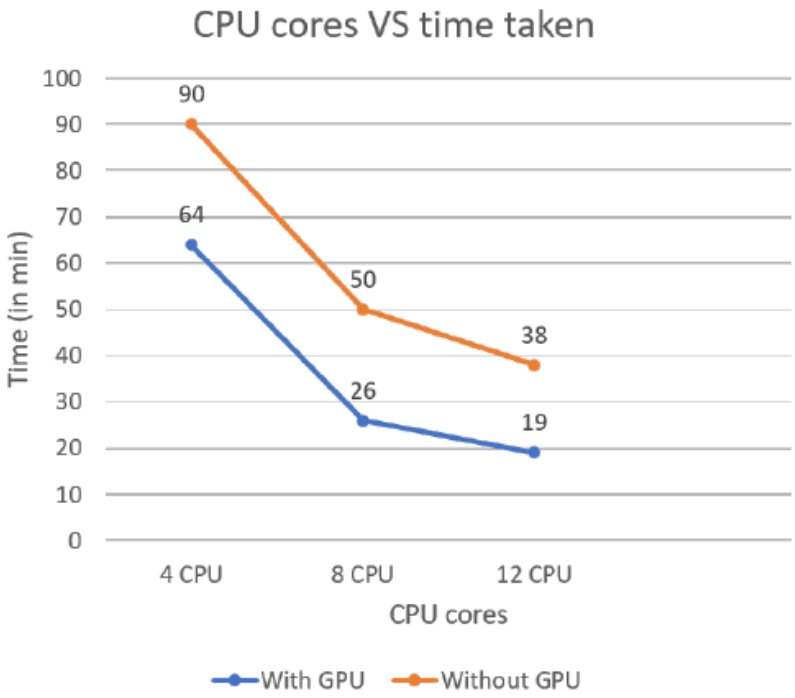


Figure 157: Plot of time taken by Pix4Dengine to process small project with and without GPU against the number of CPU cores

Potree: Potree is an open-source web based point cloud visualization tool. It has been used in the common operating platform for visualizing point clouds generated by the 3D model processing engines as well as user uploaded point cloud files. .LAS and .LAZ are the supported formats. Figure 158 shows a point cloud of ODOT Headquarters, Columbus, Ohio. The point cloud file was generated elsewhere and uploaded to the COP for visualization. Figure 159 shows a point cloud of a shed on ODOT premises at District 8, Ohio. The point cloud file was generated using Pix4D Server Engine. on the COP and visualized on the online Potree viewer. Users can share links to view 3D Point Clouds with others within ODOT's network.

Figure 158: 3D point cloud of ODOT Headquarters, Columbus, OH

Figure 159: 3D point cloud of storage shed at ODOT District 8 Headquarters, Lebanon, OH

MeshLabJS: A vast variety of open-source and commercial software tools are available to view 3D Textured Mesh produced by Pix4Dengine 1.2.0. A study was conducted to shortlist the available tools and identify the best fit as per the requirements. The outcomes of this study are summarized below (see Figure 160):

- Rhino, Autodesk 3ds Max, Blender, and Sketchfab provide rich Python-based API and can be integrated with existing Python-based applications like the COP. However, custom workflows and user interface must be developed to integrate these APIs.
- MeshLab is a desktop-only application written in C++ and is incompatible with web browsers. Hence, it cannot be integrated with the COP.
- MeshLabJS immediately comes across as the right fit since it is a JavaScript, client-side mesh processing tool that comes with a ready-to-use user interface. It can be easily integrated with an existing web application by making a few enhancements to its source code.

Tool	API Integration	Source Code	Browser Compatible	Type
Rhino	Yes	No	Unknown	Commercial
Autodesk 3ds Max	Yes	No	Unknown	Commercial
Blender	Yes	No	Yes	Open-source
Sketchfab	Yes	No	Yes	Commercial
MeshLab	No	Yes	No	Open-source
MeshLabJS	No	Yes	Yes	Open-source

Figure 160: Study of mesh viewing tools

Based on this study, MeshLabJS was chosen to view meshes on the Common Operating Platform v11.0. UC performed a detailed study of the tool to identify its compatibility with COP and device a roadmap to integrate it with the platform. Key findings include:

- The MeshLabJS source code is available on GitHub and it must be re-complied into JavaScript using Emscripten 1.38.40 toolchain. The back-end of MeshLabJS is written in C++, further compiled into JavaScript
- MeshLabJS renders a mesh model on client-side and does not send any data to the server
- Since mesh models use high amounts of RAM while rendering, MeshLabJS must be allowed to use memory (RAM) dynamically
- MeshLabJS supports .OFF, .PLY, .OBJ and .STL file formats
- MeshLabJS is compatible with Google Chrome, Mozilla Firefox, Microsoft Edge
- MeshLabJS, in its current state, does not support multi-texture mesh models
- MeshLabJS cannot render mesh models processed by Bentley's Context Capture

UC enhanced MeshLabJS source code to integrate it with COP. This enhancement allows COP to identify name of the mesh model that the user wants to view and then send this model to MeshLabJS using XMLHTTPRequest. The front-end of the website is also enhanced to accommodate MeshLabJS alongside Potree and help user easily navigate between these visualization tools. Very much like Potree, users can share links to view 3D Textured Mesh models with other within ODOT's network.

MeshLabJS Benchmarking: Rendering of the meshes in MeshLabJS depends on the client's Operating System, system RAM and the browser used. Various tests were conducted to benchmark the performance of MeshLabJS on different configurations of OS-RAM-Browser. Figure 161 shows the project details used for benchmarking. These projects were chosen based on the size of dataset and complexity of objects in the

data. Both the projects have single texture, meaning only one .JPG is produced by the Pix4Dengine for each project.

Project Name	Project Size	No. of images	GCPs	Texture Size
A	Large	1373	11	8192*8192 pixels
B	Large	1464	0	8192*8192 pixels

Figure 161: MeshlabJS benchmarking projects

Figures 162-163 show the benchmarking results for projects A and B. The green cell shows that MeshLabJS was able to render the model without any errors. The red cell shows that the model could not be opened in the given configuration. Key take-aways from these tests are -

- MeshLabJS is not compatible with Internet Explorer.
- Rendering mesh models is a memory-intensive and computation-intensive task.
- MeshLabJS will fail to render large projects on systems with less than 8GB RAM. Hence, the users are advised to use computers with at least 8GB RAM to view projects.

Operating System	RAM	Google Chrome	Mozilla Firefox	Microsoft Edge	Internet Explorer
		v86.0.4240 .183	v82.0.2	v86.0.622 .63	v11.0.210
Ubuntu 16.04	4GB			N/A	N/A
	16GB			N/A	N/A
	20GB			N/A	N/A
Ubuntu 18.04	48GB			N/A	N/A
Windows 10	8GB				
	16GB				
	48GB				

Figure 162: OS-RAM_Browser configuration testing for Project A

Operating System	RAM	Google Chrome	Mozilla Firefox	Microsoft Edge	Internet Explorer
		v86.0.4240 .183	v82.0.2	v86.0.622 .63	v11.0.210
Ubuntu 16.04	4GB			N/A	N/A
	16GB			N/A	N/A
	20GB			N/A	N/A
Ubuntu 18.04	48GB			N/A	N/A
Windows 10	8GB				
	16GB				
	48GB				

Figure 163: OS-RAM_Browser configuration testing for Project B

MeshLabJS can be used to view mesh models processed outside the Common Operating Platform. With that in mind, projects A and B were used to study the RAM utilization in an additional use case where the user uploads multiple mesh models using the option to "Open mesh file". The idea is to find the breakpoint when same project is uploaded again and again without clearing the window (Test results for this use case are shown in plots in Figures 164-165. Figure 164 shows RAM utilization for Project A on three browsers- Google Chrome, Mozilla Firefox, and Microsoft Edge. Figures 165 shows RAM utilization for Project B on three browsers - Google Chrome, Mozilla Firefox, and Microsoft Edge. In both the test cases, Google Chrome and Mozilla Firefox rendered the same model 5 times before crashing. However, Microsoft Edge could not render more than 3 models. Key takeaways of these tests include:

- Rendering multiple models in the same window results into overlapped models
- The RAM utilization by MeshLabJS does not depend on GCPs used
- A computer with more RAM will be able to render more number of mesh model
- Of all the browsers, MeshLabJS uses maximum RAM while running on Google Chrome, eventually resulting into crashed browser or irresponsive computer. On the other hand, Mozilla Firefox does good job at memory management by using the lowest amount of RAM. Thus, Mozilla Firefox is a better choice when the user is running out of RAM.
- Multiple instances of MeshLabJS actively rendering a mesh model clog the RAM and CPU. Thus, the users are advised to open one instance of MeshLabJS at any given point.

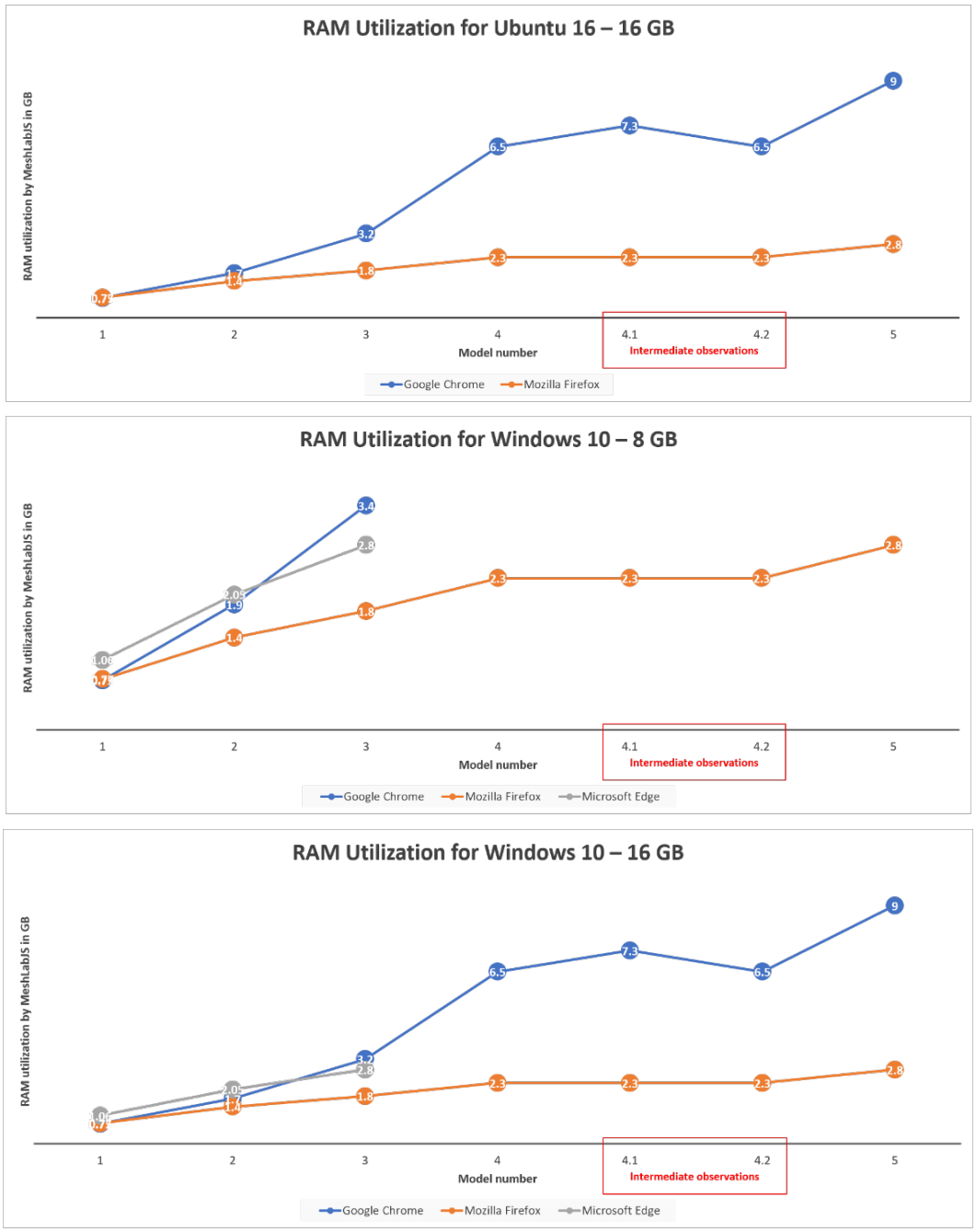
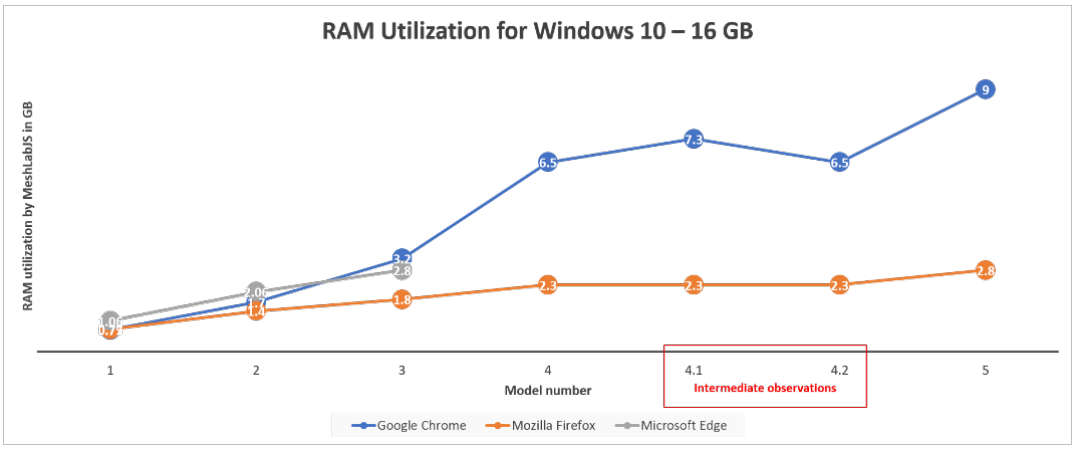
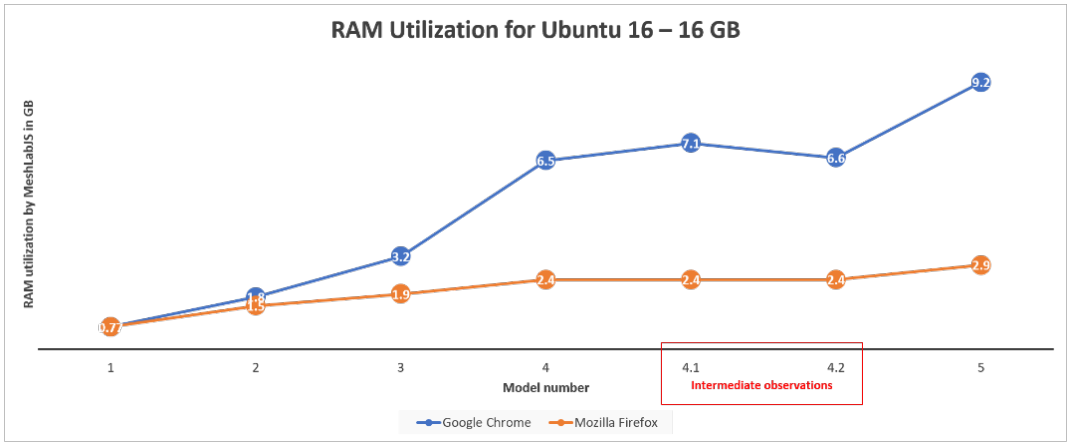


Figure 164: Project A test results





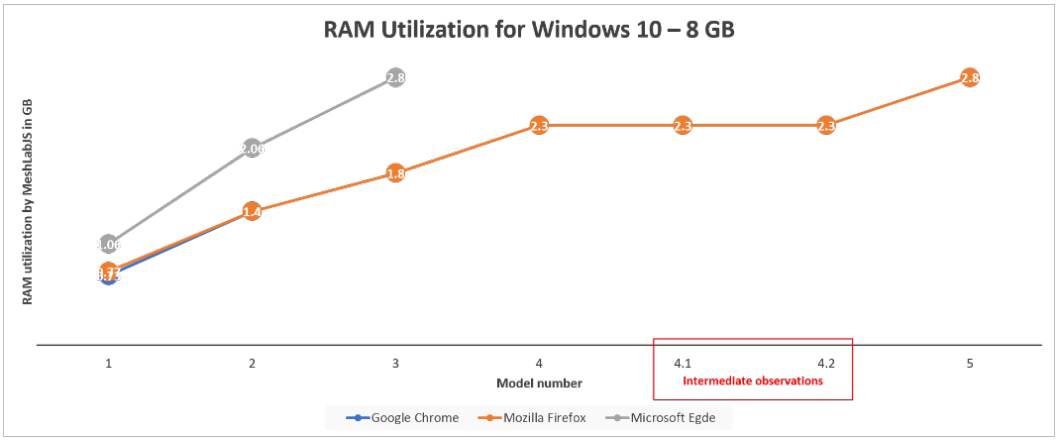


Figure 165: Project B test results

Docker Swarm: Docker Swarm is a container orchestration tool offered by Docker itself. Since it is promoted by Docker, it uses the same command line commands as Docker. Being a container orchestration tool, it manages and deploys containers at the lowest level of the architecture hierarchy. On a higher level, it manages a group of Docker hosts to run as a single virtual host where it deploys containers. In order to integrate this, a Docker-compose file was created which acts as a configuration file for Docker Swarm.

Azure File Share: Azure File Share is one of the many storage-based products offered by Microsoft Azure. As the name suggests, it is a file-share-based system that operates over the Azure cloud network. Since it is a cloud-based platform, it cannot be directly used as volume mounts to run with the container. To integrate Azure File Share, a docker volume plugin named Cloudstor plugin was installed on the server. This plugin helps Docker containers to directly use an Azure File Share account to create multiple volumes for storage.

Oracle Database: The COP initially used the PostgreSQL database which was an open-source database. However, the ODOT IT team did not support its usage, as they provided support for using the Oracle Database. Thus, the COP was upgraded from the PostgreSQL Database to the Oracle database. In order to achieve this, the necessary Oracle packages were installed as they allow command-line connectivity to the Oracle database. Since Django creates the database tables using the schema mentioned in its model's component, the website container code was updated to accommodate the changes.

Automate Deployment: The initial COP code consisted of multiple hard coded values. In order to automate the deployment process, the application code in both Website and Backend containers were updated to dynamically take values. Apart from this, CI/CD tool Jenkins was used to automate the deployment. After setting up the configuration on Jenkins, the application was able to be deployed with a single click on the Jenkins interface.

Increased uploads: After shifting the storage from the local storage to the cloud-based storage the upload cap of 4GB persisted. The COP used the Nginx as the webserver to handle all the requests and serve files to the users. It also managed all the networking configurations with regards to data uploaded to the application and timeout values for the connections being made. Also, the load balancer used a Nginx configuration file. In order to increase the uploads, the Nginx configuration files for the application and load balancer were studied and updated. Figure 166 displays the test results for uploads on COP.

Sr. No.	No. Of images	Dataset size	Status
1	33	220 MB	Successful
2	67	455 MB	Successful
3	115	917 MB	Successful
4	550	4.44 GB	Successful
5	643	5.25 GB	Successful
6	772	6.3 GB	Successful
7	902	7.4 GB	Successful
8	1036	8.45 GB	Successful
9	1132	9.24 GB	Successful
10	1213	9.9 GB	Successful

Figure 166: Upload test results

Delete automation: One of the underlying issues of the application was its inability to delete files from the file storage even after the files are deleted on the application. This led to the file storage being completely utilized rendering the COP unable to process anything. In order to resolve this issue, couple of Python and Bash scripts were created on both the website and Backend containers. These scripts compare files in the database and the file storage to delete all additional files. Below screenshots display logs from the delete automation scripts.

3.6 System Integration Technologies

In addition to exploring various field activities (along with the associated vehicle hardware and software) discussed above (e.g., traffic, mapping, construction, bridges, facilities, etc.) and the development of software platforms such as the COP,

the research project also explored the development of a small number of ancillary technologies that allowed for integration and support of these activities. A discussion of these efforts is the topic of this section

3.6.1 Milestone Mission Box, Real-time Communications

Dating back as early as the documentation of the Jeremiah Morrow bridge demolition, it was clear that the ability to be able to transmit real-time UAV imagery such as video off-site would be extremely advantageous. This would allow, for example:

1. Various participants to simultaneously view UAV footage in real-time with having to look over the pilot's shoulder or crowd around a monitor set up at the site.
2. It would also allow UAV footage to be uploaded directly to ODOT Milestone video system for archival and further communication to third parties such as the press.
3. It would allow others, not onsite to view and participate in UAV operations. For example, ODOT UAV pilots could have a live video uplink with specialists at ODOT Headquarters, ODOT consultants, etc. without everyone having to be in the field.

As a result, the research team investigated the use of cellular modems in order to establish a connection between the AUX output on the UAV controller in the field and the ODOT Milestone closed circuit video system. This system was developed as a repository for statewide system of traffic cameras such as those that can be viewed on www.ohgo.com. The system resulting from these ideas came to be called the Milestone Mission Box. Further details can be found in [41,42].

Figure 167 shows an early implementation of the Milestone Mission Box concept. All the necessary components were laid out and connected for functional testing and use during the Jeremiah Morrow demolition operations.

Figure 167: Milestone Mission Box, early concept

In short, the Mission Box is a portable console that aids the ODOT UAS center in surveillance and inspection missions. These missions include but are not limited to infrastructure inspections (bridges and buildings), surveillance of a construction project, and natural disaster surveillance (landslides). The system will aid remote pilots to store UAV/UAS data, stream video captured using UAVs.

The following lists the primary design objectives that are featured in the final version of the Milestone Mission Box:

- View and record the live video feed
    - Allow the user to view live footage from the UAV during its mission. The user can record video when needed and can play back any video file using the UI. This would provide the user with the flexibility to record only the necessary part of the mission and review the clips at any point.
- Stream live video wirelessly
    - Allow user to view live footage on any device that has wireless internet capabilities that include mobile phones, laptops, tablets, etc. Wireless streaming is accomplished through the Mission Box local area network (LAN).
    - Allow the user to view live footage using First Person View (FPV) goggles. Transmit analog live footage over 5.8 GHz signal.
    - Allow the user to save the video and provides playback/storage options through ODOT's Milestone video streaming service.
- Monitor modem connection status
    - Allow the user to monitor critical system data from the cellular modem including signal quality, WAN IP address, GPS connections, etc.
- Monitor internal system components
    - Allow the user to monitor the power consumption of the entire system as well as specific devices
    - Monitor statuses of internal relays used to turn on and off devices. This will help the user diagnose if one or more components in the mission box are not working.
- User-friendly
    - Provide the user with a simple, easy to use interface to control and monitor the system.
- Portable and rugged
    - The mission box system is packaged in a Pelican case with wheels, which is rugged and will allow users to easily transport them.

Figures 168 and 169 show the resulting ruggedized system and interface panel. Figure 170 shows the top-level hardware system design. This gives a general understanding of how all the parts used in the mission box communicate with each other and how they are connected.

Figure 168: Milestone Mission Box, final version

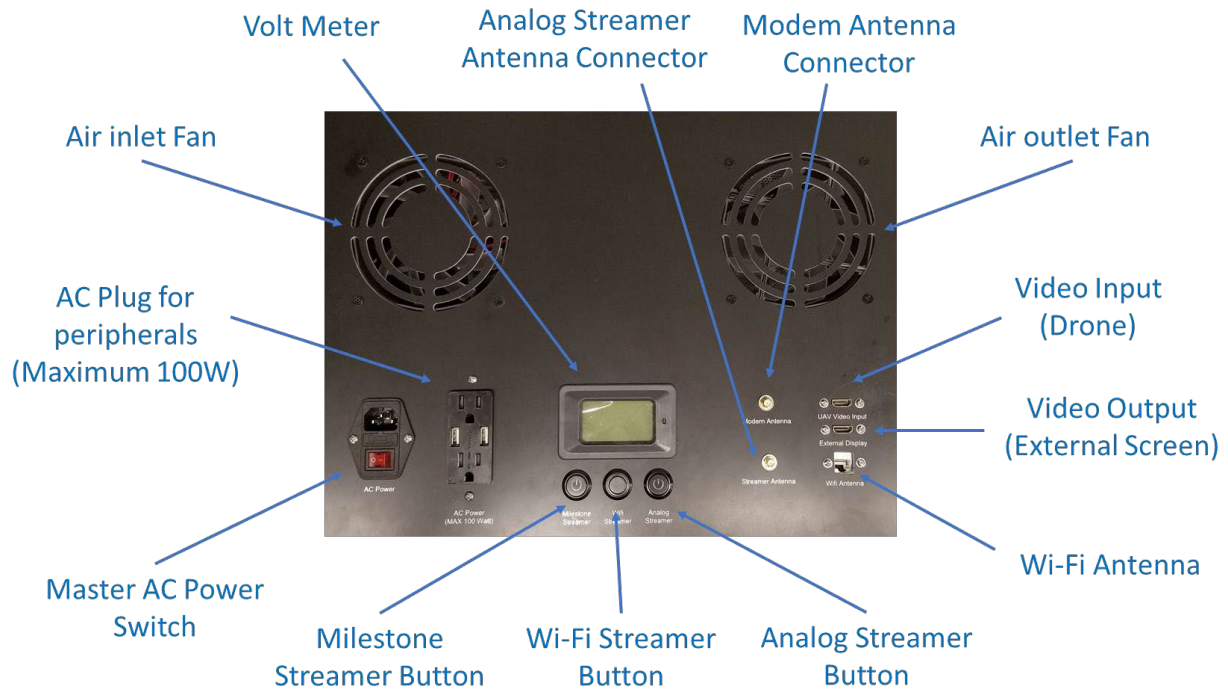


Figure 169: Milestone Mission Box, front panel layout

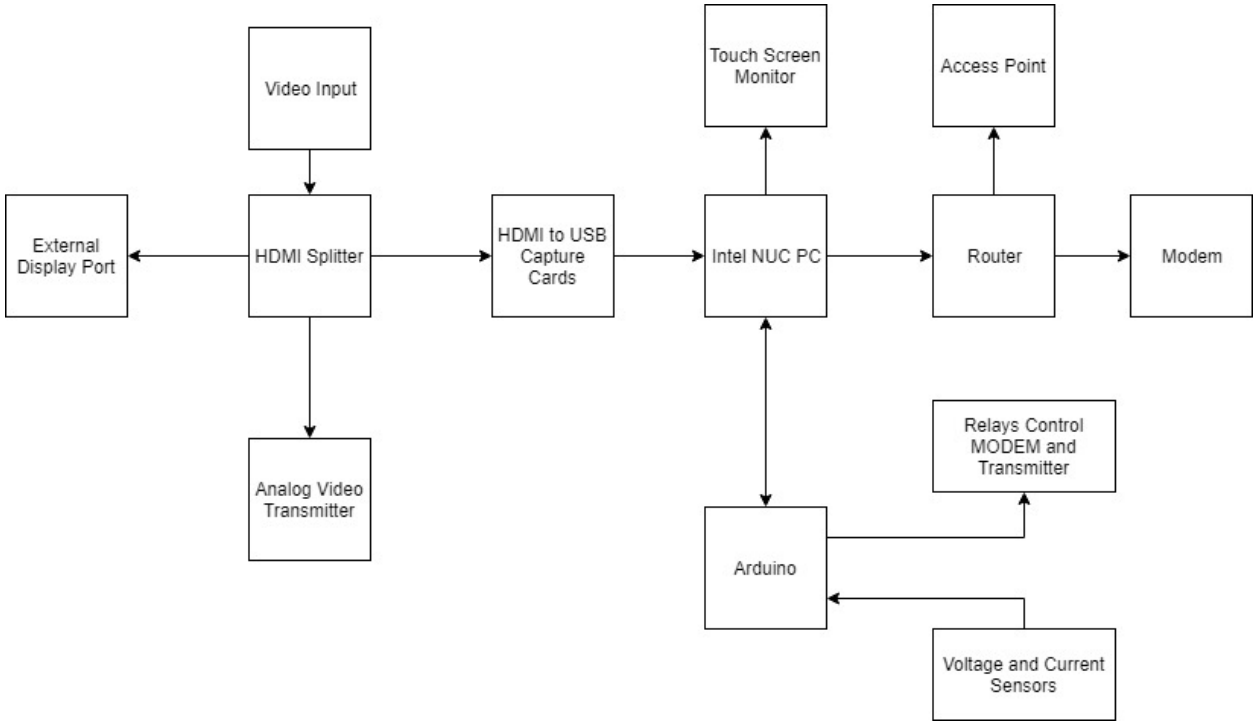


Figure 170: Top-Level Hardware System Design

For testing, demonstration, and proof of concept purposes, the Milestone Mission Box system was deployed during ODOT emergency training operations conducted at the Muscatatuck Urban Training Center (MUTC) as shown in Figure 171. Figure 172 shows a view of the training exercise ongoing from above taken by the UAV and recorded via the Milestone Mission Box.

Figure 171: Top-Level Hardware System Design

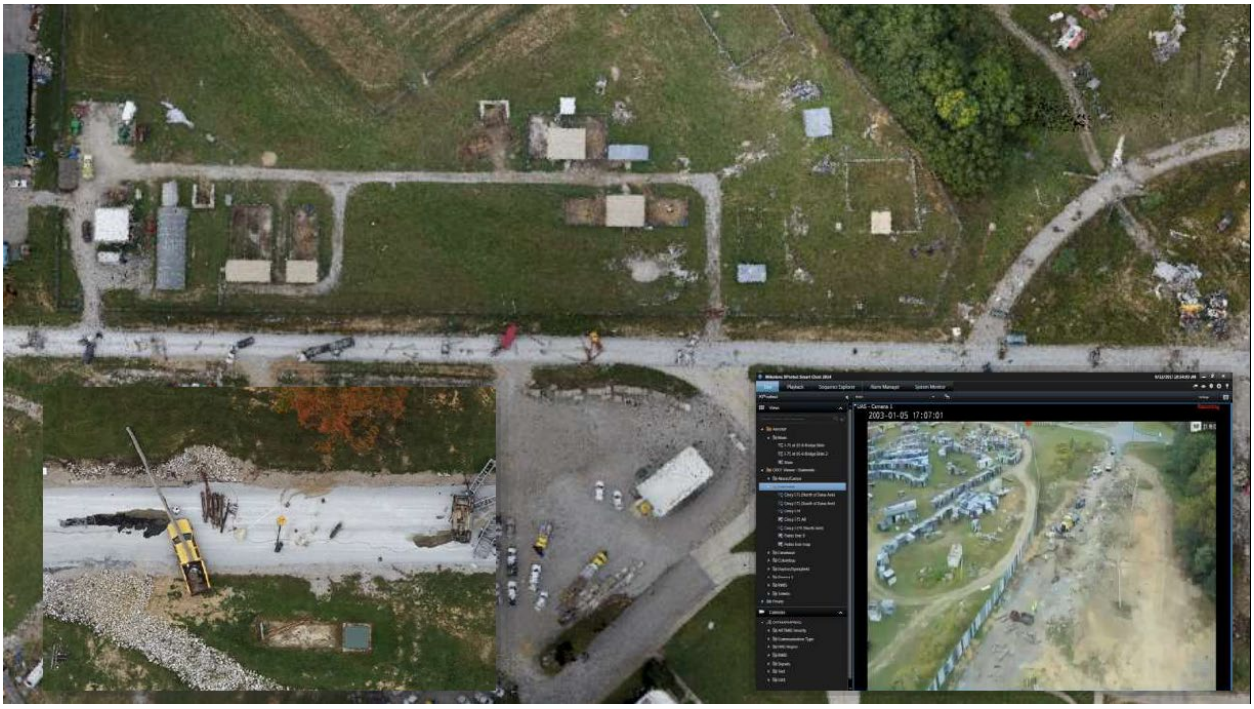


Figure 172: Aerial view of MUTC training exercise with inset of real-time Milestone recording

3.6.2 Augmented Reality, Visualization, and Hololens

The discussions above in Sections 3.3, 3.4, and 3.5 highlight the use of UAVs in developing a wide array of 3D models by using photogrammetry techniques applied to UAV imagery. Such models can be viewed on 2D screens as described in Section 3.5 for the COP, however, the development of hardware and software to permit 3D visualizations is also currently under way under the auspices of Augmented Reality (AR) and Virtual Reality (VR) techniques. Such visualization methods were the subject of a brief study during the course of this research project. After a thorough literature review on existing AR/VR technologies, it was decided to use the Microsoft HoloLens for the project as it is far ahead of the competition. Unlike HoloLens, other AR/VR technologies were not completely wireless, do not run autonomously on an operating system, need continuous HDMI connection, remote controllers for inputs, have poor battery backup, etc. We primarily investigated the use of MS HoloLens, generation 1, as a visualization tool for the 3D model. Further details can be found in [43].

HoloLens is the flagship augmented reality headset from Microsoft. It was launched in early 2016 and has a heads-up display which incorporates all the necessary sensors mounted on the device as seen in the 173 below. These sensors include:

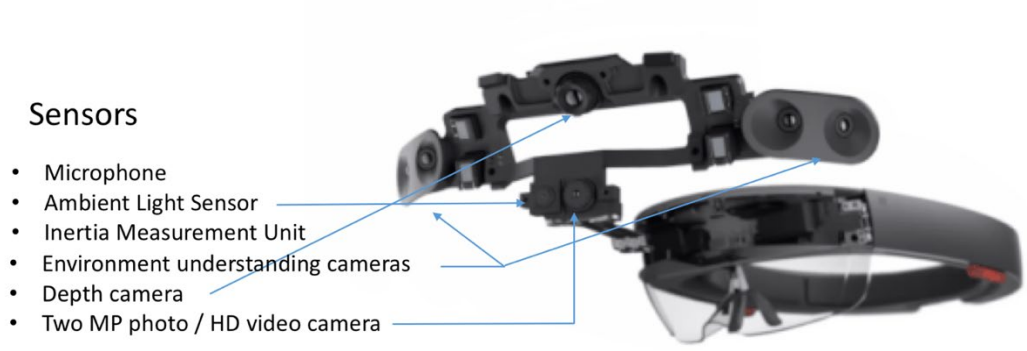


Figure 173 - Exploded view of HoloLens[1]

- Four Microphones are used for giving voice commands to Cortana and for dictation. It can also be used to create custom voice commands in applications for any type of action.
- Ambient light sensor helps in auto adjusting the brightness of the display. When the room is brightly lit, the brightness of holograms will be high, so the user can see it clearly and when the room is dim (poorly lit), the brightness of holograms will be low to reduce stress on eyes.
- IMU (Inertia Measurement Unit) helps in tracking the user's head movement. This helps in rendering user's avatar in shared session which will rotate and reposition with user's movement in the virtual space.
- The environment understanding cameras and the depth camera work together to understand user's surrounding and generate a 3D map of it, so it can simultaneously locate and map user's location.
- The front camera is a simple 2-megapixel camera capable of capturing mixed reality, so it sees what a user is seeing. We can take pictures, videos and even live stream what the user is looking at.

HoloLens, generation 1, runs on the Windows 10 operating system and does not need HDMI connection a computer system for functioning. This means that the HoloLens can run almost all the Universal Windows Platform (UWP) apps in the Store as 2D apps. UWP is a developer's platform introduced with Windows 10 that runs applications on all devices like Desktop, Mobile, Xbox, and HoloLens as well as future Windows devices. Moreover, HoloLens has a big community hence support from forums was available.

On a computer, the user gives inputs and gets output from the system. Inputs are from events like mouse movement, click, keyboard button press up or down. On HoloLens, there are three ways you can give input. They are the gazes, gestures, and voice commands.

The following additional software tools were needed for interfacing and software development:

- Unity 3D game engine
- Visual Studio
- HoloLens emulator
- Windows SDK

Note that due to the onboard memory size and limitations of computation power, it was impossible to visualize heavy models on HoloLens by downloading and rendering at run time while maintaining higher fps. As a result, the Hololens was investigated as a proof of concept using and early version of the COP webserver. Specifically, the COP also provides a REST application programming interface (API) over HTTPS. The platforms HTTP based REST APIs have been used in tandem with the Microsoft HoloLens for visualizing 3D models (processed on the platform).

The communication between HoloLens and web server or the COP was used to access a list of the stored 3D models. It works on the REST APIs hosted on the server as per Figure174.

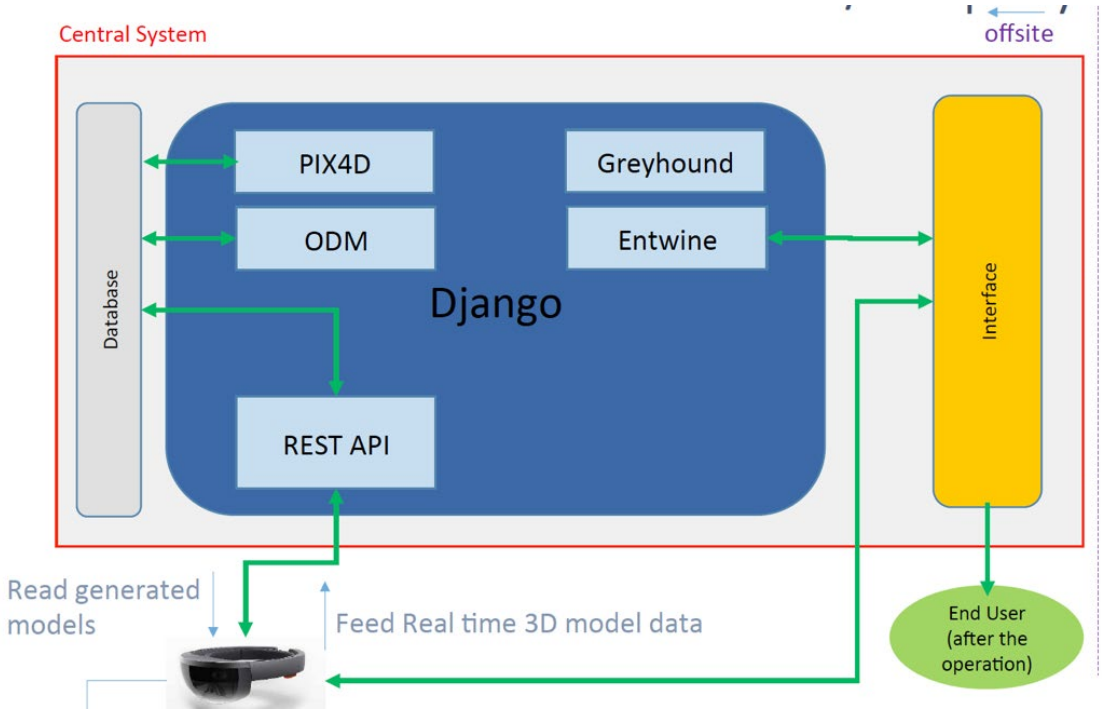


Figure 174: HoloLens connectivity with Common Operating Platform

HoloLens connects to the central system only via REST API. When the scene opens, HoloLens pings server and gets the list of all models for current user as seen on the left side of the following Figure 175.

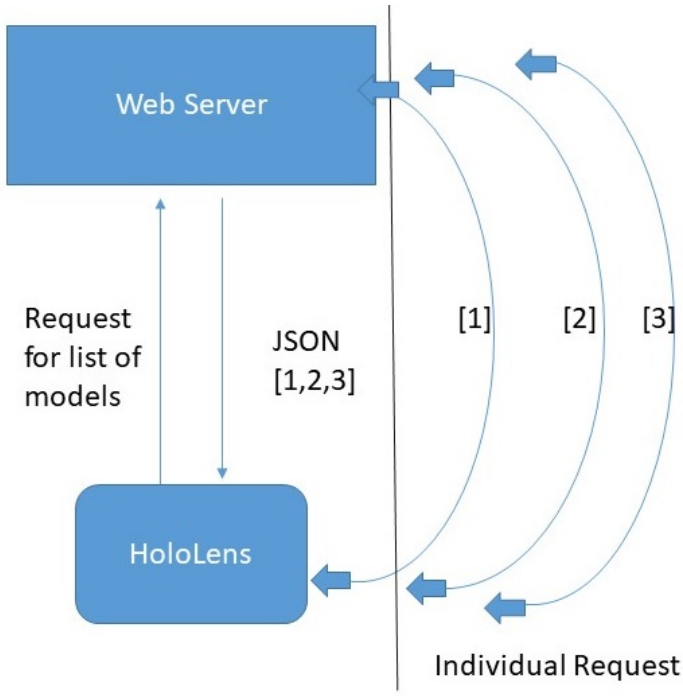


Figure 175: 3D model request from HoloLens to Common Operating Platform

This list of 3D models used by the user interface is populated in the drop-down menu. When the user selects a 3D model from the drop-down, it is downloaded using the REST API as seen on the right side of the Figure 175 above.

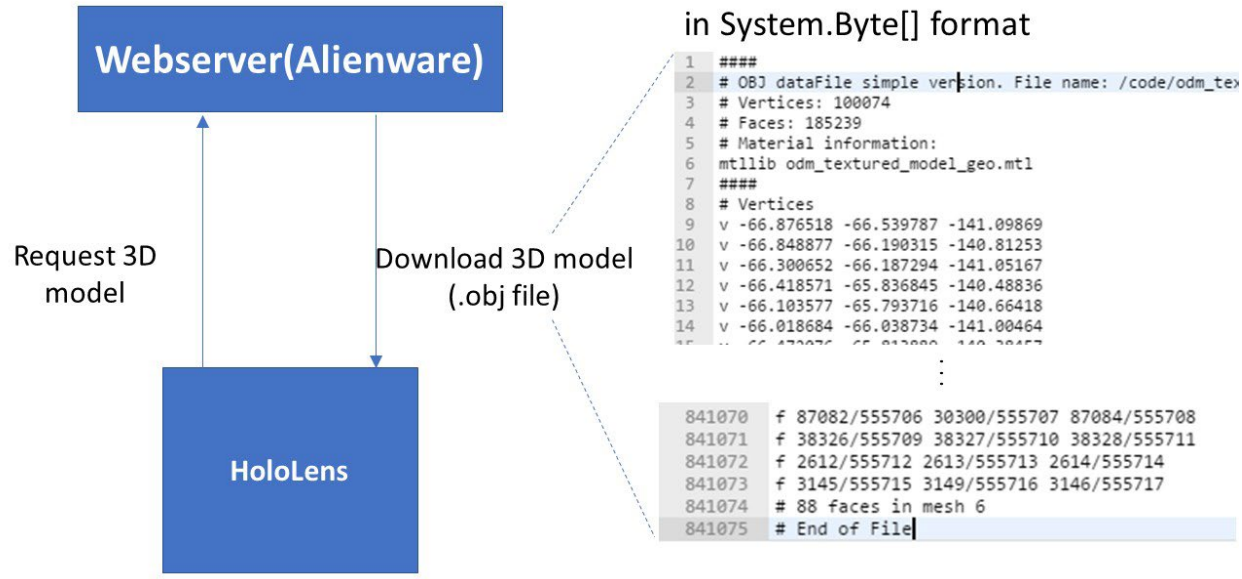


Figure 176 – 3D model request and response between HoloLens and Common Operating Platform.

The 3D model is downloaded in an object (.obj) file and its contents are seen in the Figure 176 above which is a collection of thousands or millions of vertices, triangles, and faces. The object file contains the address of the material file (.mtl) which in turns contains the address of the texture file (usually .jpg). The code then downloads these files. All vertices are stitched together to form a mesh and then, using the material file, the corresponding texture is applied at runtime. And the model is visualized in front of the user on the heads-up display of HoloLens.

This can be done in shared session as well (Figure 177), meaning when two users with HoloLens are connected to a server over a socket connection and when one user selects a model another user will see it in front of them.

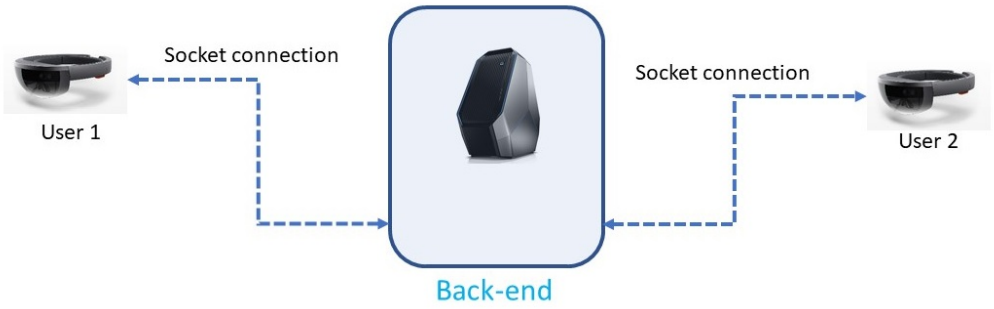


Figure 177 – Representation of shared session connectivity between HoloLens

They can also look at each other's avatar in shared virtual space and voice chat. They can drop notes/markers on the 3D model and type onto those notes in real time using the virtual keyboard, they also can scale the model bigger and smaller to have an interactive session. And the best part is synchronicity. After you make all changes to the model, move or scale just air tap on the "Sync Model" button on the interface and it will sync all the UI components like sliders, toggle buttons and synchronize the models among everyone.

Figure 178: User interface for interacting with the 3D model

In Figure 178, the buttons “Bigger” and “Smaller” will scale the model up and “smaller” will scale it down. The slider will move the model up and down. Mute *toggle* will mute your microphone and “Use Markers” will enable disable the markers functionality.

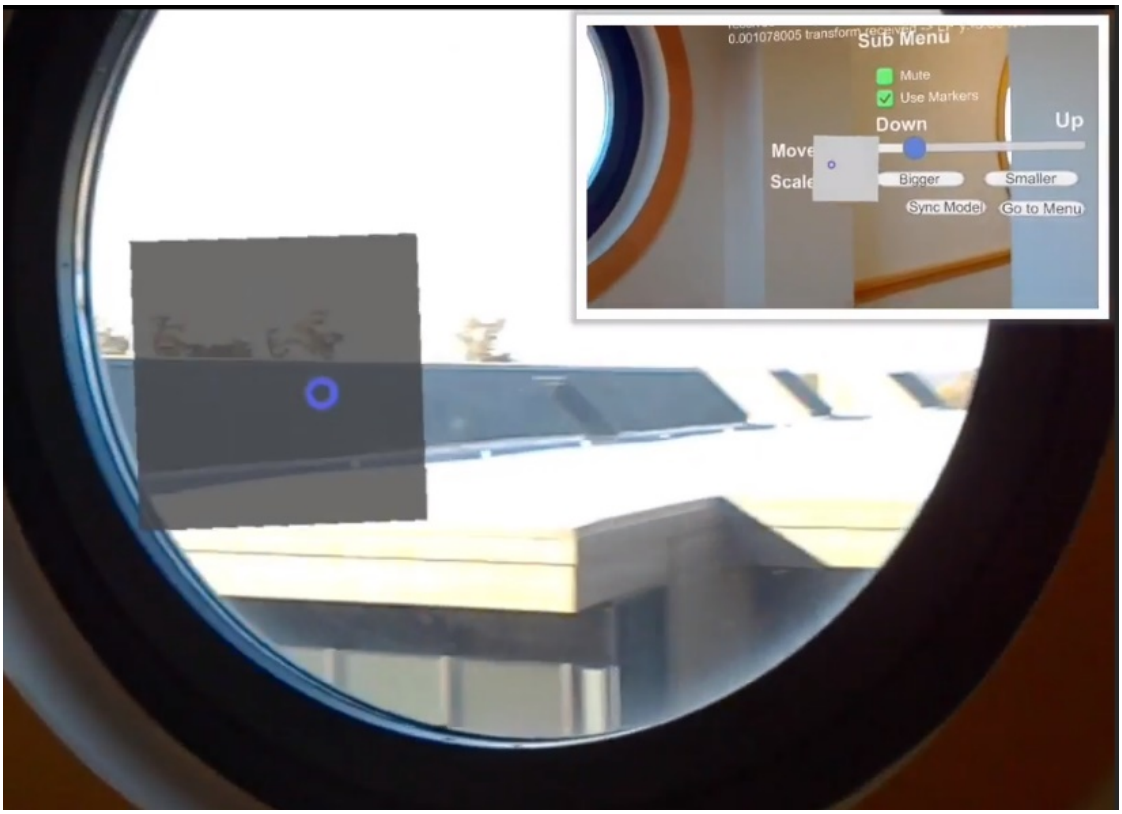


Figure 179: User's avatar in the virtual space of a shared session

Figure 179 shows the user's avatar. It is represented by a white cube. The avatar moves in 3 dimensions in virtual space when a user wearing it moves in the real world.

This work serves as a proof of concept for AR/VR technologies which could be carried forward onto a project of its own. Since the time these demonstrations were developed, Microsoft has released significant updates to both the Hololens hardware and software capabilities.

4.0 Implementation

Over the course of this project the research team endeavored to generate results which would have practical implications for ODOT and the Ohio UAS Center in keeping with the project objectives spelled out above in Section 1 and Figure 1. As part of this theme, the research team has developed a series of implementation-oriented SOP manuals which follow up on the discussions given above in Section 3 and provide further insights into the specifics of flight operations and data processing for the various application areas/proof-of-concept areas explored during this research. These include [5,6,7,31,36,41,43], see Figure 180.

Figure 180: Standard Operating Procedures

In addition to the SOPs, the research team participated with the UAS Center in several training sessions.

These included certification of 2 platforms, the Intel Falcon 8+ and the Flyability Elios.

This training also included the basic flight operations and maintenance of the sUAS. The team would also conduct its own training for the UAS Center with the tethered system (Figures 181 and 182). This training occurred in conjunction with the Multi-UAV Operations as discussed in section 3.2.5.

The UAS Center pilots in attendance were already familiar with DJI controls so there were not many differences when switching over to the Matrice 100. The main training involved the unpacking, connection of tether to the UAS, and starting of the system.

The research team would first walk through the basic steps and demonstrate the operation of the tether prior to letting UAS Center pilots try. This training also provided the UAS Center pilots with knowledge of the tips and tricks to use to control the tethered system.

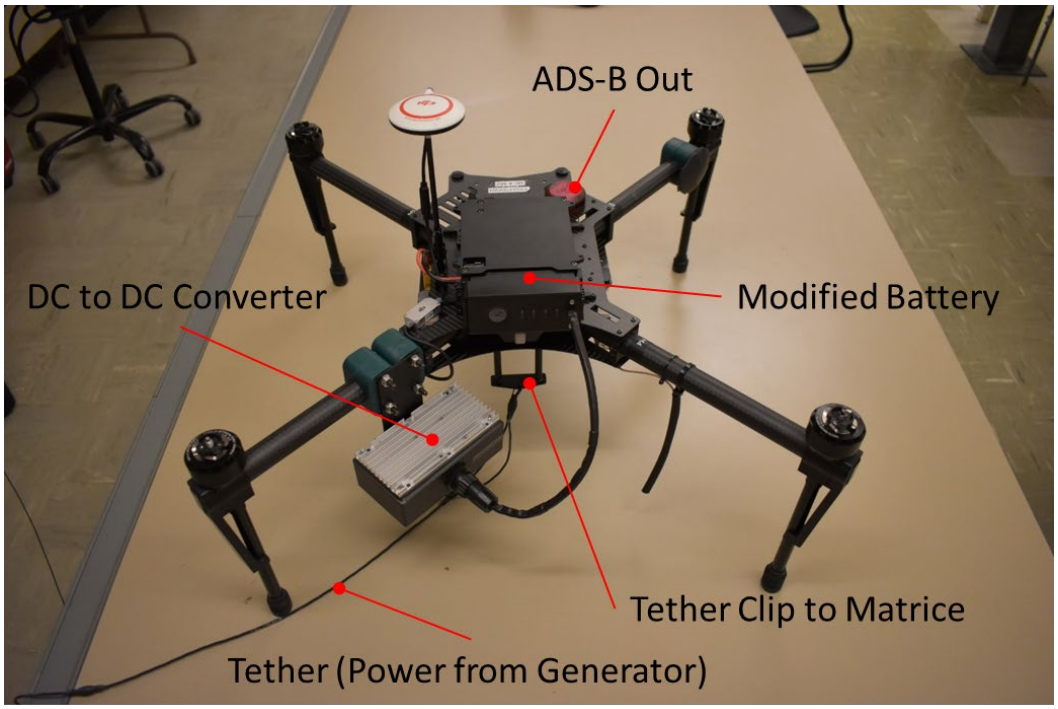


Figure 181: Tethered Aircraft System Diagram

Figure 182: Tethered System Layout

Further training was provided to UAS Center personnel (at the Ohio UAS Center in Springfield, OH and online) and to ODOT district pilots (via multiple online sessions) on the use of the Common Operating Platform (Figure 183).

Figure 183: Tethered System Layout

References

1. "Drone Aviation." [Online]. Available: http://www.droneaviationcorp.com/aerostat/.
2. Mica, John L. "Text - H.R.658 - 112th Congress (2011-2012): FAA Modernization and Reform Act of 2012." Congress.gov, 14 Feb. 2012, www.congress.gov/bill/112th-congress/house-bill/658/text.
3. FAA. SUMMARY OF SMALL UNMANNED AIRCRAFT RULE (PART 107). June 2016, www.faa.gov/uas/media/part_107_summary.pdf.
4. UAS Data Exchange (LAANC), 11 Dec. 2020, www.faa.gov/uas/programs_partnerships/data_exchange/.
5. *Standard Operational Procedures for Small Unmanned Aerial Systems*. Publication. UC Infrastructure Institute and UAV MASTER Lab, University of Cincinnati. 1st ed. 2020.
6. *Standard Operating Procedure (SOP) for Uninterrupted Traffic Monitoring Using a UAS and Computer Vision Software*. Publication. UC Infrastructure Institute, University of Cincinnati. 4th ed. 2020.
7. *Interim Results Presentation*, ODOT Project Webinar, Nov 2020.
8. Chiddarwar, A. *Application of Computer Vision Algorithms for Uninterrupted Traffic Monitoring Based on Aerial Images and Videos* [Master's Thesis]. Cincinnati (OH): University of Cincinnati; 2019.
9. Khehare, P. *Application of Computer Vision Algorithm and Deep learning for roundabout capacity evaluation using UAV aerial imagery and video* [Master's Thesis]. Cincinnati (OH): University of Cincinnati; 2020.
10. Traffic data EQUIPMENT: JAMAR TECHNOLOGIES, inc.: United States. (n.d.). Retrieved March 03, 2021, from https://www.jamartech.com/
11. J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," Dec. 2016.
12. J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," Apr. 2018.
13. "Traffic Recorder Instruction Manual: Classifying Vehicles." [Online]. Available: http://onlinemanuals.txdot.gov/txdotmanuals/tri/classifying_vehicles.htm. [Accessed: 10-Jan-2019].
14. Transportation Research Board, "Highway Capacity Manual, Sixth Edition: A Guide for Multimodal Mobility Analysis," 2016. [Online]. Available: http://www.trb.org/Publications/hcm6e.aspx. [Accessed: 06-Nov-2018].
15. ODOT. "OHIO TRAFFIC FORECASTING MANUAL Volume 2: Traffic Forecasting Methodologies." Ohio ODOT, Aug. 2018, www.dot.state.oh.us/Divisions/Planning/SPR/ModelForecastingUnit/Documents/180420%20Volume%202%20-%20Forecast%20Methodologies%20%5BDRAFT%20FINAL%5D.pdf.
16. Troutbeck, R. J. (2014). "Estimating the mean critical gap." Transportation Research Record, 2461(2), 76–84.

17. Troutbeck, R. J., of Technology., Q. U., and Centre., P. I. (1992). "Estimating the critical acceptance gap from traffic movements." Physical Infrastructure Centre, Queensland University of Technology, Brisbane.
18. Lin, T. Y., Goyal, P., Girshick, R., He, K., and Dollar, P. (2020). "Focal Loss for Dense Object Detection." IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(2), 318–327.
19. Fizyr (2020). "Keras RetinaNet." (Aug)
20. P. Khekare, S. Bonthu, V. Hunt, A. Helmicki, and K. Lee, "Novel Computer Vision Application with Deep Learning to Evaluate Roundabout Capacity using UAVs." Under review, ASCE Journal of Computing, 2021.
21. "DJI - Official Website." DJI Official, www.dji.com/.
22. SA, Flyability. "Elios - Inspect & Explore Indoor and Confined Spaces." Flyability, www.flyability.com/elios/.
23. "Powerline Tethered Drone Systems." PowerLine, www.ntpdrone.com/.
24. B. Schoeder, Kittelson&Associates, private communications, 2019.
25. *Standard Operating Procedure (SOP) for Construction Site Mapping and Producing their accurate 3D Model Representation*. Publication. UC Infrastructure Institute, University of Cincinnati. 3rd ed. 2020.
26. Balasubramian, A. *Application of Small Unmanned Aerial Systems and Photogrammetry to Monitor and Inspect Structural Health and Construction Sites* [Master's Thesis]. Cincinnati (OH): University of Cincinnati; 2020.
27. Balasubramian A, Helmicki A, Hunt V, *"Identifying and Quantifying UAV Camera Limitations: Case Study using DJI XTR Camera,"* IEEE STRATUS Conference, February, 2019.
28. Venkatesh, C., Balasubramaniam, A., Kumar, R., Brown, B., Norouzi, M., Helmicki, A., Hunt, V., Cohen, K., Kumar, M., Judson, F., *"Construction Site Evaluation Employing 3D Models from UAV Imagery,"* Proceedings of ASNT Research Symposium, Orlando, FL, March, 2018.
29. "Technical Support." Support, support.pix4d.com/hc/en-us.
30. Office of CADD & Mapping Service Ohio, "Survey & Mapping Specifications," ODOT, 2016.
31. *Standard Operating Procedure (SOP) for Mapping Structure (Bridges and Buildings)and Producing their accurate 3D Model Representation*. Publication. UC Infrastructure Institute, University of Cincinnati. 7th ed. 2020.
32. Ohio Department of Transportation, "Manual of Bridge Inspection (ORC 5501.47)," vol. 2014, p. 300, 2014 ODOT Bridge Inspection Manual.
33. Niranjan R Krishnan, titled "A Web-Based Software Platform for Data Processing Workflows and its Applications in Aerial Data Analysis" [Master's Thesis]. Cincinnati (OH): University of Cincinnati; 2019.
34. Nikita Sandip Saraf, "Leveraging Commercial and Open-Source Software to Process and Visualize Advanced 3D Models on a Web-Based Software Platform" [Master's Thesis]. Cincinnati (OH): University of Cincinnati; 2020.
35. Sohan Karkera, "Implementation of distributed cloud system architecture using advanced container orchestration, cloud storage, and centralized

database for a web-based platform. [Master's Thesis]. Cincinnati (OH): University of Cincinnati; 2020.
36. *Standard Operating Procedure (SOP) for a Common Operating Platform for Online 3D Modelling*. Publication. UC Infrastructure Institute, University of Cincinnati. 11th ed. 2020.
37. Pix4D SA, "Pix4DEngine Server," 2019. [Online]. Available: https://www.pix4d.com/enterprise/pix4dengine.
38. "Drone Mapping Software - OpenDroneMap." [Online]. Available: https://www.opendronemap.org/. [Accessed: 28-Feb-2019].
39. "Entwine & Potree." [Online]. Available: http://potree.entwine.io/. [Accessed: 28-Feb-2019].
40. Connor Manning. GitHub - potree/potree: WebGL point cloud viewer for large datasets. url: https://github.com/connormanning/potree https: //github.com/potree/potree/ (visited on 11/01/2020). Jenkins Developers. Jenkins User Documentation. 2018. url: https://www.jenkins.io/doc/ (visited on 10/29/2020).
41. *Standard Operating Procedure (SOP) for Mission Box Operation*. Publication. UC Infrastructure Institute, University of Cincinnati. 3rd ed. 2019.
42. Balasubramian, A., Yan, J., *Mission Box Design Report* [Senior Design Project Report]. Cincinnati (OH): University of Cincinnati; 2019.
43. *Standard Operating Procedure (SOP) for Augmented Reality Operations using Microsoft Hololens*. Publication. UC Infrastructure Institute, University of Cincinnati. 2nd ed. 2020.