

LOW COST REMOTE SENSING AND NAVIGATIONAL METHODS
FOR NOCTURNAL OPERATION OF COMMERCIAL
UNMANNED AIRCRAFT SYSTEMS

by

JAMIE MICHAEL MOON

PRASAD GOGINENI, COMMITTEE CHAIR
CHARLES R. O'NEILL, CO-CHAIR
PAUL HUBNER
JACOB CHAKARESKI

A THESIS

Submitted in partial fulfillment of the requirements for the degree of
Master of Science in the Department of Aerospace Engineering
and Mechanics in the Graduate School
of the University of Alabama

TUSCALOOSA, ALABAMA

2020

Copyright Jamie Michael Moon 2020
ALL RIGHTS RESERVED

ABSTRACT

This thesis addresses the development of low-cost remote-sensing and navigational techniques for commercial and private small, unmanned aircraft systems (sUAS) to support a multimodal surveying platform capable of navigating in little to no light. The research in this thesis particularly focuses upon systems that would be beneficial for an sUAS equipped with a multimodal sensor suite for visible and thermal remote sensing. The research seeks to develop a low-cost data collection workflow for gathering visible light and thermal orthomosaic maps and digital elevation models (DEM's) without relying on expensive software packages. A series of commercial and open-source image reconstruction software packages are compared and augmented with custom tools to support a low-cost aerial mapping and data analysis technique.

Additionally, this research explores alternative sensors to enable enhanced obstacle avoidance in complex flight environments in little to no light. Two novel sensor prototypes are discussed including a concept for an ultrasonic thermometer as well as a laser-projection system for detecting thin features.

This project seeks to present the building blocks for a hypothetical UAV system capable of rapidly collecting a large amount of aerial imagery and thermal data in a dark environment. Such a platform can have uses in numerous applications such as nighttime delivery services in urban environments, search and rescue missions in forested areas, as well as thermal and structural inspection of buildings.

DEDICATION

To my father
for your endless support and encouragement

LIST OF ABBREVIATIONS

AGL	Above Ground Level
BEC	Battery Elimination Circuit
CMOS	Complementary Metal-Oxide Semiconductor
COA	Certificate of Authorization
CONOPS	Concept of Operations
DEM	Digital Elevation Model
EKF	Extended Kalman Filter
FAA	Federal Aviation Administration
GCS	Ground Control Station/Ground Control Software
GIS	Geographic Information System
GPIO	General Purpose Input/Output
GPR	Ground-Penetrating Radar
GPS	Global Positioning System
IMU	Inertial Measurement Unit
LAFTR	Laser-Augmented Feature Tracking
LIDAR	Light Detection and Ranging
LOS	Line-Of-Sight
PID	Proportional Integral Derivative
RADAR	Radio Detection and Ranging

RGB	Red, Green, Blue (Standard CMOS camera palette)
SAR	Synthetic-Aperture Radar
SNR	Signal to Noise Ratio
sUAS	Small Unmanned Aircraft System
UAS	Unmanned Aircraft System
UAV	Unmanned Aerial Vehicle
UHD	Ultra High Definition
UST	Ultrasonic Thermometer or Ultrasonic Thermometry
WMS	Web Map Service

ACKNOWLEDGEMENTS

Thank you to Drs. Charles and Zheng O'Neill for facilitating the opportunity to pursue my master's and encouraging me to continue research and make my ideas into a reality. Thank you to the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) for their financial support and for making my research in ultrasonic thermometry possible.

Thank you to Josh Lotfi, Andrew Klevitch, Tuan Luong, and the rest of Tuska UAV Design Team for helping me brainstorm my ideas and for generously letting me borrow lab space and tools throughout my projects.

Thank you to Chris Simpson, Chris Simpson, and Kyle Chism for helping me navigate the academic waters and providing valuable feedback on this thesis.

Thank you to Sean Devey for always being the voice of reason and taking well-needed frisbee breaks and hiking trips to the mountains. Thank you to the band, Navier and the Stokes, for the delightful musical breaks and for encouraging me to pursue new skills outside of my normal routine.

Thank you to Steve Norris, Baron Johnson, Neal Allgood, and all the rest of the Dynetics Unmanned Systems and Aerodynamics Department for your support during my gradual transition completing this research.

Thank you to Meredith Johnson, whose unique perspectives and genuine thoughtfulness challenges me to pursue good things. Thank you to Sarah Mitchell, whose wonderful music

brightens my day and inspires me to expand my creativity. Thank you for the opportunity to participate in sharing unforgettable music together and providing an excellent distraction from this thesis.

Thank you to Jonathan Plattner for your inspirational perseverance and contagious desire for pursuing God. Thank you to Daniel Hubbard for your wise thoughts and sympathy, challenging me to take leaps of faith. Thank you to Alex Christley for your priceless thoughts and enduring genuine care for me and for others. Thank you to each of you for the endless encouragement and motivation to see the light at the end of the tunnel and for guiding me through the toughest of times.

Thank you to Rebecca “Earl the Cave Bandit” Smith for the outdoor adventures and cooking adventures and teaching me to be more courageous. Thank you to Schylar Reed for your thoughtfulness and genuine interest in the lives of me and others (and in this paper for some reason).

Thank you to Grace Community Church which has become like a family to me and continues to provide a solid framework for me to grow in my relationships with others and with Christ.

Thank you especially to my beloved family for your abundant love and support. To Mom and Dad (Greg and Tina Moon) for your generous support, encouragement, and for providing the means for me to pursue my career. To my brother, Josh Moon, who inspires me to challenge myself to learn more and for sharing wise council as we both pursue our research and tackle life together despite being worlds apart. And to all the rest of my family for your never-ending love and prayers.

Thank you, most of all, to my God who created all things and graciously gave me the ability to observe the world around me and learn deeply; whose creation continually amazes me and inspires me to build better things; and without whom I would have no purpose or inspiration.

CONTENTS

ABSTRACT.....	ii
DEDICATION.....	iii
LIST OF ABBREVIATIONS.....	iv
ACKNOWLEDGEMENTS.....	vi
LIST OF TABLES.....	xiii
LIST OF FIGURES.....	xiv
CHAPTER 1 INTRODUCTION.....	1
1.1 Background.....	1
1.2 Motivation.....	3
1.2.1 sUAS Remote Sensing in Hurricane Irma Aftermath.....	4
1.2.2 Limitations of Existing sUAS Data Collection Method.....	8
1.3 Objective.....	10
CHAPTER 2 OVERVIEW OF SUAS REMOTE SENSING AND NAVIGATIONAL TECHNIQUES.....	13
2.1 Passive Sensing.....	13
2.1.1 Aerial Imaging and Mapping.....	13
2.1.2 Multispectral Imaging.....	16
2.1.3 Photogrammetry.....	19
2.2 Active Sensing.....	20
2.2.1 Radar.....	21

2.2.2 LIDAR	22
2.2.3 Ultrasonic Ranging	23
CHAPTER 3 AERIAL SURVEYING.....	25
3.1 Introduction.....	25
3.2 Data Collection	27
3.2.1 Autonomous Mapping	27
3.2.2 GPS Emulation for DJI Drones.....	30
3.3 Post-Processing.....	32
3.3.1 Orthomosaic Generation	32
3.3.2 Photogrammetric Mesh Generation	38
3.4 Thermal Mapping.....	43
3.4.1 Methods.....	43
3.4.2 Data Processing.....	45
3.4.3 Thermal Mapping Results.....	46
3.5 Conclusions.....	49
CHAPTER 4 ULTRASONIC THERMOMETRY AND NAVIGATION.....	52
4.1 Introduction.....	52
4.2 Theory	54
4.3 Methods.....	55
4.3.1 Ultrasonic Transducer Characterization	55
4.3.2 UST Prototype Design	56
4.3.3 Heated Wall Ultrasonic Thermometry	57
4.4 Results.....	59

4.4.1 Ultrasonic Transducer Characterization	59
4.4.2 Heated Wall Ultrasonic Thermometry Experiment	60
4.5 Application for Ultrasonic Navigation.....	64
4.6 Conclusion	69
CHAPTER 5 LASER-AUGMENTED FEATURE TRACKING (LAFTR)	70
5.1 Introduction.....	70
5.2 Theory	73
5.3 Methods.....	76
5.3.1 Dot Grid Image Processing Algorithm	76
5.3.2 Vertical Stripe Image Processing Algorithm	79
5.4 Experimentation.....	82
5.4.1 Dot Grid LAFTR Experiment.....	82
5.4.2 Vertical Stripe LAFTR Experiment.....	85
5.5 Results.....	86
5.6 Conclusion	92
CHAPTER 6 CONCLUSIONS AND FUTURE WORK.....	93
6.1 Conclusions.....	93
6.2 Future Work.....	94
REFERENCES	98
APPENDIX A ADDITIONAL FIGURES	104
APPENDIX B LAFTR: FREE TARGET DETECTION	106
APPENDIX C LAFTR: CLOSE-PROXIMITY TARGET DETECTION AND RANGING....	108
APPENDIX D THERMAL ORTHOMOSAIC VIEWER CODE.....	110

APPENDIX E PHASE OFFSET CALCULATOR CODE FOR UST	111
APPENDIX F LAFTR RASPBERRY PI CODE FOR GRID METHOD	114

LIST OF TABLES

Table 1.1: ARPL UAV fleet	5
Table 3.1: Performance comparison of mapping software	38
Table 3.2: Runtime statistics for VisualSFM workflow	41
Table 3.3: Performance comparison of 3D reconstruction software	43
Table 3.4: Thermal Analysis Validation.....	49
Table 5.1: Laser dot projector characterization	84

LIST OF FIGURES

Figure 1.1: Thermal and visible light aerial images of undamaged home taken from S900	5
Figure 1.2: Aerial image of local flooding from Inspire 1 at 300 ft	6
Figure 1.3: Close-proximity inspection image from Mavic Pro	6
Figure 1.4: 3D mesh of damaged house.....	7
Figure 1.5: Textured and non-textured 3D meshes of structural damage.....	8
Figure 1.6: Illustration of multimodal thermal inspection UAV	11
Figure 2.1 Spectrum of electromagnetic radiation.....	17
Figure 3.1: 2018 Drone brand market share	26
Figure 3.2: Mavic Pro UAV.....	26
Figure 3.3: Autonomous waypoint execution in QGroundControl	28
Figure 3.4: Reference image of UA Arboretum	29
Figure 3.5: Litchi autonomous mapping grid as seen on mobile GCS	29
Figure 3.6: Example of DJI geospatial data log.....	31
Figure 3.7: Aerial maps generated using Photomerge	34
Figure 3.8: Aerial map generated with ReCap Photo	35
Figure 3.9: Aerial map generated with WebODM and imported to Google Earth.....	35
Figure 3.10: Aerial maps generated with DroneMapper	37
Figure 3.11: DEM's generated with DroneMapper	37
Figure 3.12: 3D Mesh generated with ReCap Photo	39
Figure 3.13: Point-cloud generation using VisualSFM	40
Figure 3.14: Mesh generation of VisualSFM point-cloud using Meshlab.....	41

Figure 3.15: Mesh generation using WebODM.....	42
Figure 3.16: Georeferenced orthomosaic from WebODM overlaid on Google Earth	42
Figure 3.17: Autonomous flight path as seen in Mission Planner and imported to Google Earth Pro	45
Figure 3.18: Heatmap orthomosaic overlaid on Google Earth Pro and linearized orthomosaic of thermal aerial survey generated from WebODM.....	46
Figure 3.19: Example of thermal map analysis tool	47
Figure 3.20: Calibration image with ice water container and hot water container	48
Figure 3.21: Hot and cold water calibration image as seen in thermal analysis tool.....	49
Figure 3.22: Multispectral Aerial Mapping Workflow Overview	51
Figure 4.1: 40 kHz ultrasonic transducers and setup for preliminary ultrasonic transducer analysis	55
Figure 4.2: Ultrasonic thermometry (UST) prototype	57
Figure 4.3: Heated wall ultrasonic thermometry experiment construction.....	58
Figure 4.4: Acoustic intensity ultrasonic transducers	59
Figure 4.5: Frequency response of ultrasonic transducers.....	59
Figure 4.6: Directional responses of HC-SR04 and UR-1640-TT-2	60
Figure 4.7: Heated wall transient temperature profile	61
Figure 4.8: Correlation of echo phase difference and differential temperature	62
Figure 4.9: Linear approximation of phase-temperature correlation	63
Figure 4.10: Thermal boundary layer thickness estimation.....	64
Figure 4.11: Simulated simple pulse signals with low noise and high noise.....	66
Figure 4.12: Simulated matched filters of corresponding simple pulses with low noise and high noise.....	66
Figure 4.13: Concept for ultrasonic thermometry array	67
Figure 5.1: CONOPS of laser projector illumination on aerial platforms	72
Figure 5.2: LAFTR operation for free target detection.....	73

Figure 5.3: LAFTR operation for indoor detection and ranging.	75
Figure 5.4: LAFTR dot grid image processing algorithm flowchart	77
Figure 5.5: Section of captured image with red points marking tracked dot centroids	78
Figure 5.6: Sample of tracked target with occlusion region	78
Figure 5.7: LAFTR vertical stripe image processing algorithm flowchart.....	80
Figure 5.8: LAFTR in operation with thin horizontal occlusion placed at 200 cm	81
Figure 5.9: Occlusion target point detection and categorization	82
Figure 5.10: Target and shadow peak detection	82
Figure 5.11: Unfiltered and bandpass filtered spectra of 532 nm laser	83
Figure 5.12: Narrow, medium, and wide laser dot spacing	84
Figure 5.13: LAFTR test setup	86
Figure 5.14: Lateral moving human target at 5 ft	87
Figure 5.15: Multiple lateral moving human targets at 6 ft and 10 ft.....	87
Figure 5.16: Lateral moving target at 6 ft with no ambient light.....	88
Figure 5.17: Sequence of target tracking and ranging with estimated ranges listed.....	89
Figure 5.18: Sample image of LAFTR viewer with target overlay and range estimation.....	90
Figure 5.19: Estimated range relationship	91
Figure 5.20: Linear region of estimated range.....	91
Figure 5.21: Associated error of range estimation.....	91

CHAPTER 1

INTRODUCTION

1.1 Background

Small Unmanned Aircraft Systems (sUAS) have become increasingly prevalent in recent years due to advances in miniature sensor and computer technology, lithium battery and radio technology, and flight control software. A small UAS, for the context of this thesis, is an unmanned aircraft weighing less than 55 lbs that is operated autonomously or by a remote pilot. Such aircraft typically are categorized as fixed wing, helicopter, or multirotor UAVs and often operate under electric power. Originating from hobby aircraft, these aircraft are becoming increasingly sophisticated in the past decade. The explosion of mobile phone technology in the past years led to the rapid development of MEMS (Micro Electro-Mechanical System) sensors as the demand for miniaturized sensors and integrated circuit packaging grew. MEMS sensors such as accelerometers, gyroscopes, barometers, and compasses added remarkable functionality to mobile devices, enabling advanced camera stabilization, location and state estimation, and overall haptic user experience. Inertial Measurement Units (IMUs) consisting of MEMS accelerometers, gyroscopes, and compasses have been developed to provide developers with a compact state-estimation system on a small circuit board. In addition to cell phones, these IMU's have found extensive use in automotive, robotic, cinematic, and gaming applications [1].

The miniaturization of integrated circuit components has also led to the introduction of microcontrollers that enable basic computational processing in a small hardware footprint. These single-chip devices possess CPUs, memory, and input/output peripherals required to perform

processing tasks independently of a larger system. The availability and affordability of these components led to the rise of open-source prototyping boards such as those designed by the Arduino project [2] which began in the early 2000's. Arduino boards expand the functionality of ARM or ATmega microprocessors by incorporating voltage regulators, digital and analog converters, input/output breakouts and serial communication.

The availability of microcontroller boards such as Arduino allows hobbyists, researchers, and companies to rapidly develop and program systems for data acquisition, process control, automation, and numerous other applications. This technology soon was adopted by RC model hobbyists seeking to automate systems onboard small remote-controlled (RC) aircraft. By combining this technology with IMUs, hobbyists developed basic autonomous flight controllers. One of the notable implementations of early flight controllers was the MultiWii project [3]. MultiWii featured the integration of an IMU, taken from a Nintendo Wii MotionPlus controller, and an Arduino Pro Mini microcontroller in order to automatically stabilize a multirotor sUAS. Projects like these introduced a new level of complexity to RC aircraft and showed that the integration of small, low-cost sensors into commercial UAVs can greatly expand their functionality.

Since the introduction of GPS access to the civilian market in the 1980's, portable GPS receivers have also been developed into smaller, more efficient packaging. Small GPS receivers have found their way into automobiles, aircraft, watercraft, and cell phones. This technology was soon adopted to provide absolute position data for robotic control systems. Systems like the MultiWii utilize IMU's for state estimation, but a new level of autonomy was achieved with the integration of GPS receivers. Powered by readily-available microcontroller boards, open-source flight control systems such as ArduPilot, ArduCopter, and Pixhawk were created [4], [5]. These

projects combined attitude and control hardware into compact autopilot systems with robust software capable of synthesizing sensor data, operating control algorithms, and executing autonomous navigational tasks.

In the United States, the Federal Aviation Administration (FAA) issued 14 CFR Part 107 in August of 2016 as a regulation to more easily integrate sUAS into the National Airspace (NAS) [6]. This regulation replaced preceding laws which required commercial operators to acquire a Certificate of Authorization (COA) according to Section 333 of the FAA Modernization and Reform Act of 2012 [7]. The new Part 107 regulation requires UAS operators to pass an aeronautical knowledge examination and receive an sUAS license before flying drones for commercial or research purposes. UAS aircraft must then be registered and then operated and maintained according to the guidelines set forth by 14 CFR Part 107. This regulation enables more of a streamlined process for gaining certification as individuals or businesses to operate the growing number of commercial aircraft on the market currently. Thus 2017 marked a boom in UAS commercial operations as new UAS service companies were formed and existing larger businesses developed in-house UAV operations. In the last two years, the commercial drone industry has largely served the fields of construction, agriculture, and surveying using UAV platforms costing typically less than \$1,500 [8]. As the government continues to develop the legal infrastructure to support UAS integration into the NAS, the existing regulations and availability of technology has provided opportunity for growth in the field of UAV aviation.

1.2 Motivation

The motivation for this project primarily stemmed from a disaster-relief mission performed by the University of Alabama on October 6th, 2017. In partnership with the Alabama Center for Insurance Information and Research (ACIIR), the University of Alabama's College of

Engineering sent a team of researchers to Florida to assess structural damage of reinforced homes categorized as “FORTIFIED” according to national standards [9], [10]. The group, consisting of representatives from the departments of Civil Engineering and Aerospace Engineering, collected data in four cities of homes that were impacted by category-5 Hurricane Irma on September 10th, 2017 [11]. The objective of the mission was to compare the damage to FORTIFIED and non-FORTIFIED homes by collecting visible and thermal data. Visible imagery enabled large structural damage to be identified whereas thermal imaging provided insight into additional structural damage such as water infiltration and heat flux from roof damage. The mission provided an opportunity to test the feasibility of deploying sUAS for rapid multispectral data collection in a real-world application.

1.2.1 sUAS Remote Sensing in Hurricane Irma Aftermath

The University of Alabama’s Aircraft Rapid Prototyping Lab (ARPL) performed aerial imaging in residential areas of Florida impacted by Hurricane Irma. During the operation, three UAV’s were flown: a custom hexacopter built from a DJI S900 frame, a DJI Inspire 1 quadcopter, and a DJI Mavic Pro quadcopter. The S900 provided thermal imaging and radiometric data using a statically-mounted FLIR Vue Pro R infrared camera [12]. The Mavic Pro and Inspire 1 quadcopters provided high-definition visible light images for inspection and photogrammetry. The Inspire 1 was used for high altitude and large area mapping, while the Mavic Pro was suited for close-proximity inspection. The fleet, listed in Table 1.1 below, enabled a variety of data to be collected at each target location and demonstrated the compatibility of using off-the-shelf equipment alongside custom systems.

Table 1.1: ARPL UAV fleet

UAS	Weight (lb)	Flight Time (min)	Payload	Camera Mount	Spectrum	Video Downlink
S900	10.5	~12	FLIR Vue Pro R & GoPro Hero 5 Session	Static	IR & Visible Light	Yes
Inspire 1	6.75	22	DJI Zenmuse X-3	3-Axis Gimbal	Visible Light	Yes
Mavic Pro	1.62	27	Sony 4K Camera	3-Axis Gimbal	Visible Light	Yes

UAS Data Acquisition and Methodology

A custom multi-spectral payload was integrated on the S900 frame. The primary objective was to collect temperature data from damaged roofs to identify areas of potential water damage. This inspection also observed the thermal effects of missing roof tiles. The addition of a complimentary visible light camera (GoPro Hero 5 Session) enabled a direct comparison between visible damage and thermal effects for each location as illustrated in the undamaged home in Figure 1.1.



Figure 1.1: Thermal (left) and visible light (right) aerial images of undamaged home taken from S900

The Inspire 1 was used for high-altitude (up to 400 ft AGL) imaging over large affected areas. The aircraft's size and stability made it suitable for the coastal winds in excess of 15 knots.

These data can be combined using photo registration to create a comprehensive, high-resolution aerial map encompassing the area surveyed. This platform proved to be a viable method for obtaining data over a wide region with minimal time. From these data, affected areas were inspected for flooding as seen in Figure 1.2 below.



Figure 1.2: Aerial image of local flooding from Inspire 1 at 300 ft

The Mavic Pro was used for close inspection of targeted areas of structures. With the aircraft's small size and obstacle-avoidance capability, the Mavic Pro was suitable for approaching and tracking targets of interest upon a given structure. As seen in Figure 1.3 below, specific damaged regions or roof components can be observed closely.

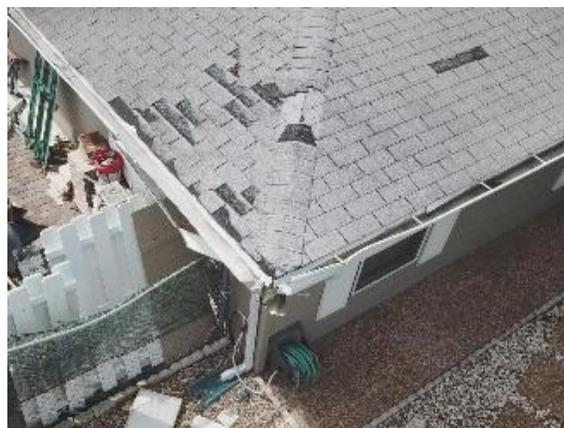


Figure 1.3: Close-proximity inspection image from Mavic Pro

Photogrammetry

Using the Inspire 1, a series of paths were flown around target structures while recording UHD video. From these videos, still frames were selected at regular intervals and processed using Autodesk ReMake 3D reconstruction software. This software used the image data to generate a 3D mesh, seen in Figure 1.4, with texture data for additional visual inspection, providing a comprehensive analysis of the structure's state.



Figure 1.4: 3D mesh of damaged house

The 3D meshes generated with this method possessed enough resolution to provide information about the material composition and state of damaged structures. Figure 1.5 below illustrates the spatial and textural depth of a portion of the same 3D mesh shown above.

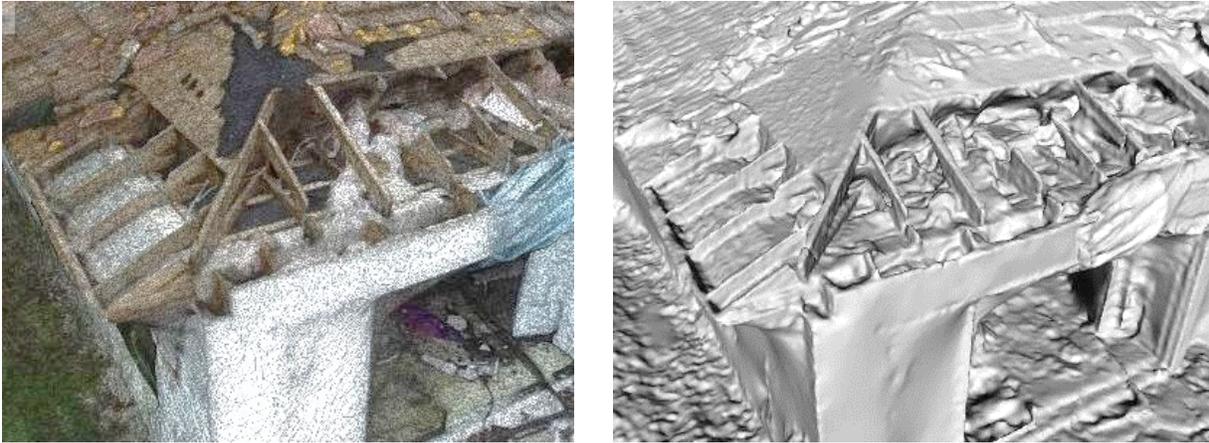


Figure 1.5: Textured and non-textured 3D meshes of structural damage

Results

The S900 UAV demonstrated capability of collecting thermal data with a FLIR camera payload. However, due to the narrow focal length and static mount on the thermal camera, many data were unusable due to misalignment and motion blur. To mitigate this, the FLIR camera was later integrated into a 3-axis brushless gimbal for additional stabilization and control. Data collected with the Inspire 1 and Mavic Pro UAVs yielded local aerial maps and 3D models which allowed quick analysis of flooding and structural damage. This deployment clearly illustrates the ability to rapidly provide basic aerial imaging data for use in disaster relief, structural damage assessment, and geographic information system (GIS) mapping. Using a combination of commercially-available and custom UAS, a diverse set of visible and thermal data can be collected quickly and efficiently.

1.2.2 Limitations of Existing sUAS Data Collection Method

The data collected during this mission reemphasized that sUAS allow a robust platform for rapidly collecting data over a large target area for structural damage assessment during active events. While the data from this deployment proved useful for general assessment and reinforced

the capability of remote inspection with sUAS, there remained several limitations regarding this technology.

The thermal data during this task were collected with an infrared camera which provides an estimate of surface temperature. This provides insight into regions of varying heat flux but is limited by purely measuring radiative flux [13]. It is desirable to detect air temperature in addition to surface temperature to provide a more accurate measure of heat flux on and around a building envelope. Collecting these data with a multimodal sensor could combine ambient air temperature with thermocouples, surface temperature with an IR camera, and fluid temperature with an ultrasonic thermometer. Augmenting basic imaging with novel sensors like these can provide a better representation of the thermal environment, allowing more comprehensive aerial data acquisition.

Another practical limitation was discovered in Florida during the deployment as aircraft were operated in urban residential areas. During the mission, data collection was significantly limited or deemed impossible due to the proximity of trees, telephone wires, buildings, cars, debris, and other obstacles near structures being inspected. These obstructions posed a challenge not only to the safe operation of the sUAS, but also hindered the quality of data collected. It is desired to have additional sensing capability to enable the aircraft to avoid thin obstructions such as poles and powerlines. By integrating additional avoidance sensors, it is possible to expand the UAV's ability to ingress and egress when operating at low altitudes in an urban environment.

While this mission demonstrated the rapid use of commercial sUAS for remote sensing, it also revealed the need for a more efficient data processing workflow. During this deployment, the aircraft were flown manually by a pilot. By implementing mission planning tools, the flight time can be further optimized. For mapping operations, it is desirable to have an automated flight

path to ensure consistent image alignment and efficient data capture compared to a manually-flown path. This can be achieved by utilizing ground control software (GCS) that supports this functionality. For more complex flights for 3D photogrammetric scanning, it may be necessary for a pilot to manually operate the aircraft. In these cases, it is also desired to still have a methodology to extract geographic information from videos captured during manual flight. By implementing custom tools with existing software packages, it is possible to develop an open-source data processing workflow for use with both custom and commercial UAV's such as DJI drones.

1.3 Objective

The objective of this project is to demonstrate technology supporting the development of an sUAS equipped with a multispectral, multimodal sensor suite for improved navigation and comprehensive data acquisition. The methods discussed compose a low-cost solution for the next generation of sUAS remote sensing and navigation in challenging flight environments. Such a platform would enable enhanced simultaneous visible and thermal remote inspection capability for application towards building inspection, agricultural mapping, and disaster relief while operating in low-light, GPS-denied environments.

A hypothetical UAV platform (illustrated in Figure 1.6) would likely consist of a multirotor platform equipped with a compact remote sensing suite comprised of multimodal sensors. For enhancing the UAV's spatial awareness, the aircraft would be fitted with laser projection and ultrasound rangefinders to provide altitude estimation and obstacle avoidance in little to no light and in indoor and complex flight environments. The sensor payload can then utilize thermocouples, ultrasonic thermometers, and an infrared (IR) camera to remotely collect comprehensive thermal data in and out of a nocturnal environment.

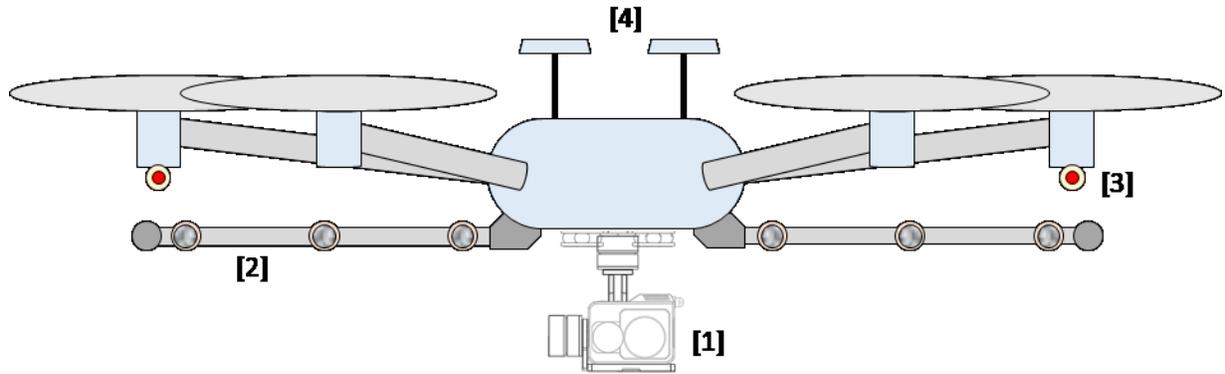


Figure 1.6: Illustration of multimodal thermal inspection UAV, featuring [1] visible/thermal imaging camera, [2] ultrasonic thermometer arrays (mounted on retractable landing gear), [3] laser pattern projectors, and [4] GPS/GNSS receivers

This thesis introduces common existing remote sensing and navigational techniques currently employed on UAV systems in Chapter 2. A series of low-cost sensing methods are then demonstrated, establishing fundamental concepts consisting of a multispectral surveying workflow, a novel ultrasonic thermometer and rangefinder, as well as a laser-projection obstacle avoidance system. The open-source surveying workflow discussed in Chapter 3 is designed to offer a no-cost solution to provide the rapid data collection of visible light and thermal orthomosaic and digital elevation models for multiple UAS platforms. Such a workflow should enable users to generate high resolution geographic surveys of comparable quality to expensive commercial software packages. In addition, the ultrasonic thermometer and laser feature tracking sensor prototypes seek to demonstrate low-cost alternatives to existing obstacle avoidance or remote sensing methods. The ultrasonic thermometer prototype in Chapter 4 seeks to determine if common ultrasonic transducers are capable of measuring air temperature to provide augmented thermal sensing capability to the multimodal UAS. Finally, the laser feature tracking sensor discussed in Chapter 5 prototype seeking to determine if basic laser diodes can provide high-contrast features to enable obstacle detection and ranging with basic RGB camera sensors.

Together, the methods of data collection and sensing shown in these experiments can be combined into a hypothetical next-generation sUAS platform for comprehensive multimodal surveying. The goal of the experiments set forth is to show that these methods provide sensing capability comparable to conventional sensing methods while expanding commercial UAS operation into complex, low-light environments at a low cost.

CHAPTER 2

OVERVIEW OF SUAS REMOTE SENSING AND NAVIGATIONAL TECHNIQUES

2.1 Passive Sensing

This section introduces basic data collection methods using passive sensors. Passive sensors in the broad sense collect information by detecting signals from the ambient environment such as sound waves and electromagnetic (EM) radiation. In contrast, active sensors—discussed in Section 2.2—require the transmission of a signal to gather information. Passive data are stored digitally onboard the UAV and can then be processed into useful data for inspection, surveying, and navigation.

2.1.1 Aerial Imaging and Mapping

Theory

Perhaps the most basic of remote sensing technologies is photography. Digital cameras operate by directing visible light rays onto an array of photodetectors. The array is typically a complementary metal-oxide semiconductor (CMOS) array which is used to detect the intensity of photons and convert them into digital signals. These signals are processed as a matrix of pixels and then stored for each channel or frequency of light captured—typically red, green, and blue at a depth of 8 bits per channel. Developments in small-scale manufacturing have resulted in the production of high-resolution digital image sensors and optical lenses. Ultra-high definition (12+ megapixels) cameras are now smaller than a US quarter. Improved sensor quality enables greater noise reduction and faster and cheaper microprocessors enable images and videos to be

captured quickly at high resolutions. These compact camera sensors are conducive to mounting onto aerial platforms such as UAVs to collect data from areas which otherwise cannot be reached.

Another crucial technology for aerial imaging is camera stabilization. Camera stabilization systems utilize moving optical lenses or a moving sensor to counteract motion blur when capturing digital images. The angular velocity, and thus motion blur, of the camera is measured with MEMS gyroscopes which provide signals to servo motors or Voice Coil Actuators (VCAs). This stabilization can eliminate small amplitude vibrations but cannot necessarily compensate for the motion and vibration of a UAV in flight. UAV's continuously rotate their bodies to maintain heading and position in the presence of wind and general disturbances. To counteract these low-frequency motions, a stabilized camera gimbal can be utilized. Similar to vibration reduction, a gimbal employs the use of a MEMS Inertial Measurement Unit (IMU) to detect the attitude of the camera which is free to rotate on a multi-axis gimbal. The gimbal is then fitted with motors to counteract the motion using a PID controller. Recent development in brushless motor and electronic speed control units allows cameras to be smoothly stabilized to compensate for nearly all motion in flight. This stability is essential for collecting consistent aerial imagery for mapping or general photography.

Once image data are collected, digital processing can be performed to reveal additional information about the region surveyed. Processing techniques such as gray-scale adjustment, edge enhancement, and spectral filtering can be used to detect and isolate desired features and perform analysis on certain frequencies of data for numerous surveying tasks [14]. Image registration—typically employing feature recognition algorithms—can also be used to stitch

images together to create orthomosaic maps. This essentially forms a synthetic aperture that allows large spatial areas to be surveyed from a single mobile UAV.

Applications

The most basic application of aerial surveying is imaging areas of interest for identifying and analyzing target objects. For instance, a UAV equipped a digital camera can provide the overhead inspection of industrial, commercial, or residential building structures for maintenance or damage assessment [9]. UAVs operating at moderately high altitudes can perform larger mapping operations by simply collecting image data with a downward facing camera. Using geolocated images enables surveying data to be collected and merged into larger maps constructed from smaller images. The resulting high-resolution map could often provide higher quality than a satellite image for a desired location. Such maps can be used for geographical information systems (GIS) in the form of geological surveys to collect data for larger existing geographical databases. Aerial mapping is useful for hydrological purposes to monitor the morphology of natural bodies of water such as river erosion [14]. This can be achieved using basic image processing techniques to isolate the color of water against land and thereby estimate surface area of water in the surveyed region. Monitoring these data over time can provide insight into the flow characteristics and erosion of rivers for civil planning or irrigation.

Aerial surveying can also provide crop health monitoring and vegetation counting for agricultural land. The rapid analysis of agricultural crops is desired as conditions can change frequently due to weather and seasonal effects [15]. Basic aerial imaging data can be manipulated to distinguish the types of crops in a survey area and thus estimate the crop acreage.

Additional processing such as motion tracking and object recognition yields more specialized data collection. This can allow the UAV to identify and track known targets for

commercial tasks such as counting industrial or agricultural inventory assets, locating human targets for disaster relief or security, and analyzing municipal infrastructure by observing street traffic. Advanced image processing techniques not only allow remote sensing tasks like these but can also function as obstacle avoidance or tracking for the UAV's navigational system [16]. Motion tracking and optical flow methods allow objects to be tracked from a stream of digital image data which can be processed onboard the UAV. Recent advancements have incorporated pairs of digital image sensors to allow stereoscopic image processing [17], [18]. Pairs of simultaneous images collected with the two cameras are compared and matched with a 3D reconstruction algorithm to estimate the range of detected features. This enables a low-fidelity depth measurement for estimating the aircraft's altitude or position relative to obstacles for enhanced navigation. Digital processing methods for these applications can be adapted for a wide range of remote sensing applications for image data collected and navigation using the visible light spectrum.

2.1.2 Multispectral Imaging

Theory

While standard digital cameras capture visible light, it can be valuable to collect invisible light in the Near Infrared (NIR) and Infrared (IR) frequencies. All matter contains thermal energy which can be emitted outward in the form of electromagnetic radiation. Thermal radiation manifests as radiative flux in the visible and IR bands. Although thermal radiation is a volumetric effect, most internal radiation is absorbed by adjacent matter except for the outer surface [13]. As such, thermal radiation is considered to be a surface effect, and detecting the intensity of IR radiation can allow the temperature of the emitting surface to be measured.

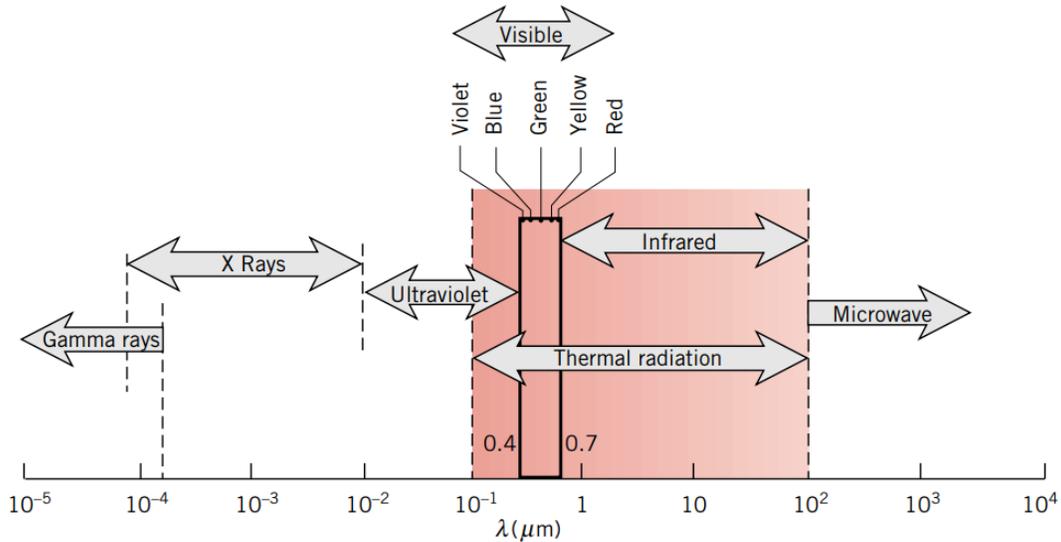


Figure 2.1 Spectrum of electromagnetic radiation [13]

Infrared cameras are similar in construction to visible light sensors but use an array of digital photoreceptors that are sensitive to NIR or IR frequencies. The system can then be calibrated to estimate the thermal emission of the scene and give a temperature estimate for each pixel. For most common IR cameras, the sensor's own temperature is monitored to maintain accuracy instead of requiring active cooling. Some traditional glass does not transmit infrared wavelengths, so crystals such as zinc selenide or zinc sulfide may be used for their wide range of transmission [19].

The primary advantage of these sensors is that the data include the information of radiometry which can be used to estimate heat flux or locate desired targets based off of their thermal signature. Many of the same post-processing techniques can be employed to enhance data analysis using similar image processing techniques described previously. These techniques include feature-based image registration for map generation, thresholding for analyzing desired frequency ranges, and motion tracking for target detection.

Applications

The applications of IR aerial imaging are similar to that of visible light but provides additional information in the form of temperature data. IR imaging can be used for detecting heat loss for industrial applications, detecting human subjects for disaster relief and law enforcement, and estimating the health of vegetation for agricultural applications [20]. While visible light sensors required ambient light for functionality, IR cameras can still be utilized in little to no visible light.

IR data can be rapidly collected by a UAV to observe the temperature of a building or industrial structure to provide an estimate of heat flux and thus provide insight on the thermal performance of the target. This can be used to assess a building's HVAC (Heating, Ventilation, and Air Conditioning) performance or search for structural damage. One study demonstrated the utility of a UAV-borne thermal imaging system to detect faulty cells on the solar panels of a photovoltaic plant [20]. UAVs equipped with these sensors can rapidly be deployed to assess damage that is not immediately visible to the human eye.

IR imaging can be used for not only examining known subjects, but can be utilized to detect hidden subjects. Thermal imaging is often used by the military for detecting and tracking targets demonstrating a thermal signature that is significantly different from the surrounding environment [22]. The detection and tracking of human or automotive heat signatures can provide a method of search and rescue for law enforcement or first responders to disasters that does not require daylight.

This method is also becoming useful in recent years for precision agriculture. In regards to plant health, thermal imaging can be used to measure the temperature of leaves which is related to the rate of the plant's transpiration [21]. Additionally, thermal data can quantify the

moisture content of the soil. Combining information of the available ground moisture and the plants' receptiveness can provide insight into the health and required irrigation for crops. Data collected in thermal imaging surveys can be combined with visible light or other data to create comprehensive multispectral map databases. These techniques demonstrate the versatile application of a UAV equipped with thermal imaging capability.

2.1.3 Photogrammetry

Theory

Photogrammetry is a process of generating 3D models from a series of two-dimensional images. 3D reconstruction is possible by matching features between images and estimating the camera's spatial position for each image. Common points between adjacent images are then located in 3D space, resulting in the formation of a 3D point cloud. The cloud can then be transformed into a mesh by connecting points with polygons textured with the visible light data collected from the 2D images. The result is a textured 3D model of the environment or subject. The addition of GPS data enables the 3D model to then be geolocated and overlaid on existing maps.

A notable advantage of this method is that photogrammetry does not require advanced ranging sensors but can generate detailed models from basic two-dimensional images. While photogrammetry can be achieved with simple image sensors, the process requires significant post-processing to extract 3D information. The computation must typically be performed post-flight on a computer capable of high-volume geometric processing. Nevertheless, this method enables a robust method for building comprehensive volumetric models of the surveyed environment for a wide range of 3-dimensional geometric analysis.

Applications

Photogrammetry can provide a comprehensive 3D scan of a target region for numerous applications. Photogrammetric data collected from a UAV platform can be used to generate high quality models of large building structures for damage assessment or historical preservation [22], [23]. The same method can be used to provide volumetric and thus product weight estimation for industrial mining operations or to monitor progress on construction sites.

Photogrammetry can also expand the aerial mapping techniques described previously and generate geographic information system (GIS) maps with terrain information. These 3-dimensional digital elevation models (DEM) can be used to provide high-quality local maps with a short turnaround. Such addition of 3D data can be utilized for hydrological analysis to further analyze the flow and erosive behavior of water bodies with higher precision [24]. For military operations, this method could also be employed to rapidly collect data from an area of interest and build a high-quality 3D map with minimal risk. The map could then be used strategically for simulating or planning battlefield scenarios in regions with little or no preexisting terrain data. The ease of image capturing makes photogrammetry a suitable method for rapidly collecting volumetric data with a low-cost UAV and sensor platform. The information can then be post-processed for numerous applications.

2.2 Active Sensing

This section describes sensing methods that involve the active transmission and reception of energy as a means of signal detection. These sensors actively transmit acoustic or EM waves that are then received after reflecting off of the local environment. Most commonly for UAS, this involves acoustic or radio range finding using time-of-flight or Doppler measurement to collect data.

2.2.1 Radar

Theory

Radar, or “radio detection and ranging,” utilizes radio waves generated from antennas to create radio pulses that propagate through space [25]. These signals reflect off of the local environment and are received by a similar antenna. By measuring the time delay, phase offsets, and frequency changes of the echoed signal, the position and velocity of targets can be detected using a wide variety of processing algorithms. The radio frequency of the signal and the pulsed waveform can be tailored specifically for the nature of the target and the type of data to be collected. Radar is widely used in ground-based and full-scale aircraft as well as space-born remote sensing systems. In recent years, the miniaturization of computer processors and electronic circuit fabrication has enabled the development of more compact standalone radar sensors whose formfactor can be integrated into sUAS payloads. These radars can be used to collect data for both remote sensing applications as well as providing state estimation and obstacle avoidance for the aircraft’s navigation. The lower EM frequencies of radar also enable waves to penetrate atmospheric moisture, foliage, and even solid earth to allow volumetric spatial measurement in certain applications.

Application

UAV’s equipped with radar systems can be used to collect data which in the past have traditionally been captured with full scale aircraft or satellites. Basic target detection and tracking can be utilized by sUAS for both sensing and navigation. Variants of radars could be used to detect the presence and range estimation of the ground and other obstacles to enhance the aircraft’s ability to navigate indoors or ion GPS-denied environments [26]. The same system could also be used to detect peer UAVs, full scale aircraft, and other types of vehicles. Tracking targets like these can not only provide advanced situational awareness for the drone’s

navigational system but also yield use for numerous remote sensing applications. Target tracking can be used for traffic analysis, security, counter-UAS defense, or other similar tasks.

By connecting small radar arrays to the UAV's inertial navigation system (INS), sUAS can be operated as a mobile transceiver for synthetic aperture radar (SAR) image collection. Similar to the aerial mapping methods described previously, SAR data can be compiled into maps of large surveyed areas. The advantage of this method over visible or thermal light sensors is that, with proper post-processing, it can produce high resolutions results through cloud cover and in the dark. Ground-penetrating radar (GPR) variants of SAR enable subterranean data collection for numerous agricultural and scientific applications [27]. Ground-penetrating SAR can also be used to estimate the biomass of subterranean crops over a large agricultural area. The ground-penetrating radars developed by the University of Alabama's Remote Sensing Center were used to successfully map several meters of snow in Grand Mesa, Colorado and 3 km of ice in Greenland [28]. Such methods can be employed for archaeological surveying, mine sweeping, and other applications. With radar systems becoming more compact, it is possible to have an sUAS equipped with radars that can provide both remote sensing data and advanced navigation.

2.2.2 LIDAR

Theory

LIDAR—"Light detection and ranging"—works similarly to radar but uses collimated light in the form of a laser. The laser beam is swept across the scene using an actuator and the time delay of the reflected laser pulses is measured at each position. The result is a collection of range estimates for each position of the laser projector. These data are then processed into a cloud of points which can be reconstructed to generate a 3D mesh. LIDAR systems are compact enough to be integrated as sUAS payloads which can enable the aircraft to generate high

resolution volumetric models of the local environment. Unlike radar, however, the EM frequencies used with LIDAR do not penetrate opaque objects but only provide 3D reconstruction of the surfaces in the sensor's line-of-sight (LOS).

Application

LIDAR has become a popular sensor for robotic navigation systems such as UAS and autonomous cars due to its high-resolution sensing [29]. LIDAR equipped on a UAS allows a rapid 3D view of the surrounding environment. This is particularly useful for detecting thin features such as trees or towers which are more difficult to detect with image-based navigation methods. When paired with image data, LIDAR can be used to generate high-resolution 3D meshes similar to photogrammetry. Volumetric data from LIDAR can thus be used for 3D mapping complex interior or exterior spaces for buildings and geological features. This can be useful not only for recreating the environment for analysis but also to allow the aircraft to navigate more efficiently in complex spaces.

2.2.3 Ultrasonic Ranging

Theory

Ultrasonic range finding is performed by emitting and receiving pulses of high frequency sound waves through the air with a transducer. Ultrasonic frequencies exceed the audible range of human hearing typically at 40kHz or beyond [30]. Similar to the time-of-flight methods employed by radar and LIDAR, ultrasonic sensors use a microcontroller to measure the delay time between the emitted pulse and its echo received by a similar transducer. While the waveform typically consists of a simple pulse, the pulse waveform and frequency can be tailored to suit the application. This method is common for robotic and industrial automation applications to estimate the position and velocity of nearby objects in air or other fluids. These compact sensors have also been integrated to assist multirotor UAV's in autonomous takeoff and landing.

Application

Ultrasonic sensors, as with other robotic applications, can enhance a UAV's state estimation. Ultrasonic rangefinders can be used on the base of the aircraft for high fidelity altitude estimation for takeoff and landing. Similar sensors on the sides can be used on multirotor UAS for obstacle avoidance between stationary objects and peer aircraft [18].

By adapting existing radar algorithms and hardware configurations, ultrasonic sensors could be evolved into arrays capable of remote damage inspection and fluid property analysis [31]. As demonstrated in Chapter 4, ultrasonic sensors prove to be capable of measuring air temperature which can have applications in HVAC inspection and weather observation. By further developing data processing algorithms, ultrasonic sensors onboard UAV's can be expanded from simple obstacle avoidance to advanced navigation and specialized data acquisition.

CHAPTER 3

AERIAL SURVEYING

3.1 Introduction

Aerial mapping with sUAS offers a low-cost, low-risk means of rapidly collecting data over a vast area. Cost is reduced from collecting the same imagery from the ground or by hiring a full-scale aircraft to perform the task. Such operations require extensive planning and pose a larger safety and financial risk than using small UAS with comparable sensing capability. Maps generated from UAV scanning can then provide a comprehensive view of the site in a matter of hours instead of days or weeks. Further expanding 2D mapping into 3D mapping by means of photogrammetry opens opportunities for additional applications. Applications of photogrammetry extend from geological surveying, mining operation monitoring, construction site or industrial inventory calculation, and structural damage assessment. Combining photogrammetry with mapping enables high-resolution mapping with an additional depth of detail to show terrain contours, buildings, and vegetation in the form of a digital elevation model (DEM).

According to the 2018 Drone Market Sector Report by Skylogic Research, the popular Chinese drone company, DJI, possessed approximately 74% of the commercial sUAS market share last year as illustrated in Figure 3.1 [32]. DJI's increasing popularity likely arose from the compact design of their aircraft with the ability to include ultra high-resolution stabilized cameras and stereoscopic optical sensors for obstacle avoidance on newer models.

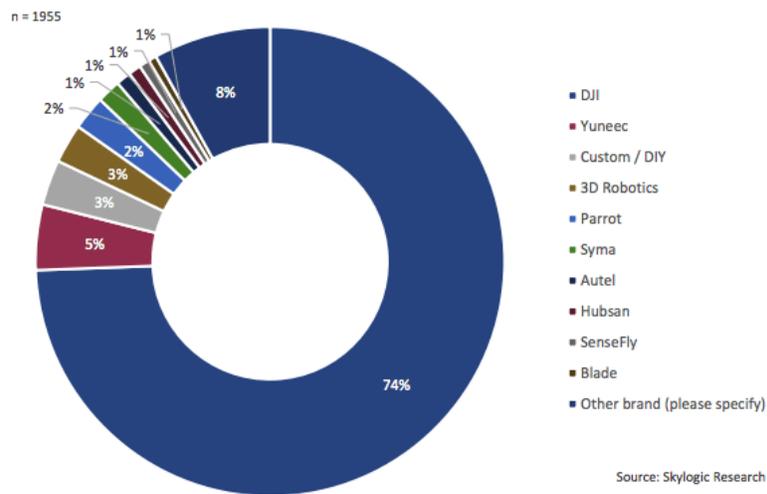


Figure 3.1: 2018 Drone brand market share [33]



Figure 3.2: Mavic Pro UAV [34]

A DJI Mavic Pro, shown in Figure 3.2, was used in the following experiments to collect high resolution aerial imagery to demonstrate its ability as a platform for rapid mapping and photogrammetry. The Mavic Pro is a 335 mm semi-autonomous electric quadcopter which is controlled with a radio transmitter operating on a proprietary DJI datalink protocol. The handheld controller can be connected to a smart phone or tablet which can then run ground control software for displaying telemetry and video information as well as sending additional commands to the aircraft. The default ground control software, DJI “Go” was compared to a 3rd party application, Litchi [35].

3.2 Data Collection

3.2.1 Autonomous Mapping

While the DJI Go application enables basic and advanced flight modes and camera controls for the aircraft, it does not feature the ability to generate and upload waypoints without manually recording a flight in its entirety. For large-scale surveying it is useful to have the ability to create missions prior to flight to collect adequate data while minimizing the time the UAV is airborne. With only the ability to manually record flights, it is time-consuming to execute automated surveying missions. Instead, an ideal GCS interface would enable the user to pre-plan a variety of mission configurations including grid patterns for mapping, orbits for 3D scanning, and circles for fixed-wing surveillance. For open-source autopilot systems, perhaps the most widely used GCS software packages are QGroundControl and MissionPlanner [36], [37]. As demonstrated in Figure 3.3, GCS software such as QGroundControl enables the user to draw flight patterns on a computer or tablet by programming waypoints and setting action for capturing images as desired. While these packages are widely available for open-source systems, they are not compatible with the communication protocols established within DJI systems.



Figure 3.3: Autonomous waypoint execution (orange points) in QGroundControl (red line indicates actual aircraft path)

Litchi is an alternative GCS application which provides similar functionality to the DJI Go app with expanded capability of autonomous flight pattern generation. The following flight path was drawn within Litchi's "Hub" application on a computer browser as seen in Figure 3.5. The mission consists of a grid pattern intended to capture high-resolution visible light images at each point for creating a map of the University of Alabama Arboretum in Tuscaloosa, AL. The mission shown here, as well as others, can be saved to cloud storage where it is downloaded to the GCS and uploaded to the aircraft. Upon execution, the aircraft traveled to each waypoint at an altitude of 100 m and captured a RAW image of the ground while facing north. The aircraft then returned to the home position after 12 minutes in flight (approximately 60% of the flight battery capacity). For still images taken with DJI drones, GPS data including latitude, longitude, GPS altitude, barometer altitude, and time are recorded to the EXIF metadata in each image file. This is later exploited to generate geographically referenced orthomosaics.



Figure 3.4: Reference image of UA Arboretum (Google Earth)



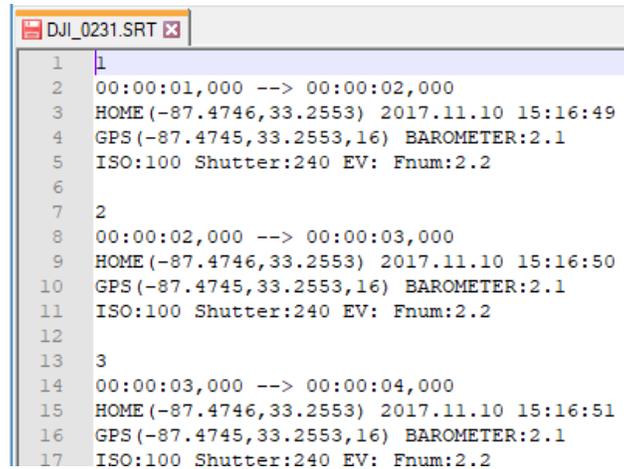
Figure 3.5: Litchi autonomous mapping grid as seen on mobile GCS

3.2.2 GPS Emulation for DJI Drones

While waypoint mission planning is an efficient method for capturing image data for large areas, it may not be optimal for certain conditions that require data collection in a challenging flight environment. For instance, imagery for 3D reconstruction using photogrammetry requires images of the subject at varying angles to enable the image processing algorithm to detect structure from motion. If scanning a subject such as a digging operation at a construction site or building structure scan in an urban setting, it is important for the aircraft to avoid any obstacles that may obstruct its path during flight. The autonomous waypoint missions described previously do not take terrain or obstructions into account. While some DJI drones—such as the Mavic Pro used in these experiments—are equipped with obstacle avoidance sensors, they only work in certain directions and with adequate ambient lighting. These sensors make the aircraft aware of obstacles to prevent hitting them, but additional development is required to have the aircraft dynamically plan alternate routes around the obstructions. Even if the obstruction is avoided, the alternate path may deviate from the planned scanning path, resulting in unexpected inconsistencies in the photogrammetric data.

To this end, it is desired to develop a method of rapidly capturing images during a manual flight for the purpose of 3D reconstruction. As mentioned previously, DJI drones tag still images with GPS information, but when flying in manual mode, each image must be captured by the pilot by pressing the shutter button at each point of interest. Alternatively, the UAV could be set to record a UHD video for the duration of the flight, thus allowing the pilot to maneuver the aircraft safely and efficiently. While recording videos, DJI drones can export a log file in tandem with GPS data for each video captured. The data, formatted as seen in Figure 3.6, are saved as a

proprietary *.SRT format which lists the GPS time and position as well as camera settings at a sample rate of 1 Hz for the duration of the corresponding video file.



```
DJI_0231.SRT
1 1
2 00:00:01,000 --> 00:00:02,000
3 HOME (-87.4746,33.2553) 2017.11.10 15:16:49
4 GPS (-87.4745,33.2553,16) BAROMETER:2.1
5 ISO:100 Shutter:240 EV: Fnum:2.2
6
7 2
8 00:00:02,000 --> 00:00:03,000
9 HOME (-87.4746,33.2553) 2017.11.10 15:16:50
10 GPS (-87.4745,33.2553,16) BAROMETER:2.1
11 ISO:100 Shutter:240 EV: Fnum:2.2
12
13 3
14 00:00:03,000 --> 00:00:04,000
15 HOME (-87.4746,33.2553) 2017.11.10 15:16:51
16 GPS (-87.4745,33.2553,16) BAROMETER:2.1
17 ISO:100 Shutter:240 EV: Fnum:2.2
```

Figure 3.6: Example of DJI geospatial data log

In order to process DJI videos into reconstruction software, the videos were split into frames and tagged with the GPS information contained in the SRT log files. This was achieved using a custom Python script utilizing OpenCV and GPSPHOTO libraries. The script collects the videos and their corresponding SRT logs for the directory selected by the user. It then converts the SRT files to text files and reads them into the workspace. The stream of GPS data is then parsed into a list of dictionaries each containing the latitude, longitude, altitude, barometer value, and GPS time for each data point. Using OpenCV, each video file is exported as a sequence of JPEG images at full resolution at a framerate of 1 Hz to match that of the GPS log [38]. The output images are stored in new directories for each video file. Using GPSPHOTO, the GPS coordinates and time are embedded in each of the output JPEGs' EXIF metadata to emulate being geotagged during capture. Additional information could be embedded at this stage but is not mandatory for the next steps in this workflow. The resulting images are tagged JPEGs which can be used to create georeferenced meshes and orthomosaics using other software.

With this method, the location accuracy is clearly limited to the possibility of error between the timing of the SRT log points and the frames being extracted. For multirotor aircraft, this error should be negligible but for higher-speed fixed wing aircraft it may result in higher positional deviations. Assuming a multirotor cruise speed of 5 m/s during video capture and a worst-case phase offset of 1 second between SRT logging and frame capture, the tagged images could possess up to a 5 m positional error. This falls close to the average GPS accuracy of 4.9 m [39]. If, however a fixed wing UAS possessed a similar phase offset while cruising at 15 m/s, the error would increase to 15 m which could cause more significant alignment issues for the image reconstruction. With additional development, an algorithm could be developed to interpolate between SRT log points and a higher frame count can be extracted from videos and tagged.

3.3 Post-Processing

3.3.1 Orthomosaic Generation

Image data from the arboretum survey above were processed into an orthomosaic map using a series of software packages for comparison. The data set used consists of 55 geotagged 12-megapixel JPEG images captured on February 10, 2019 using the Mavic Pro camera. For each of these tests, they were performed on an MSI desktop computer operating with Windows 10. Tasks were executed on an Intel i5-9600K processor with 32GB of RAM. Conversely, the ReCap Photo test was performed on an Autodesk cloud computer.

Adobe Photoshop CC 2019

The map data was processed using Adobe Creative Cloud's "PhotoMerge" script for Photoshop¹. This script imports the images and registers them using Adobe's proprietary image

¹ A single commercial license of Adobe Photoshop CC 2019 costs approximately \$408 per year as of December 2019 [73]

processing algorithms to register, align, and blend the images into a seamless mosaic. The quality of merged imaging is typically high with this method, but the output is a purely two dimensional image. Because of this, terrain elevation and object height are not captured in the output orthomosaic. In addition, PhotoMerge does not regard the GPS data within the images. Since this information is discarded, the output image does not have absolute ground control points which can often result in warping or barreling of the output image during reconstruction.

As seen in Figure 3.7 below, the basic “Reposition” method—consisting of simply translational transforms—generally arranges the images into a map with equal spacing but significant misalignment between images. The “Automatic” method appears to apply projective transforms to the images, resulting in better alignment between images but introducing significant overall distortion. While the Reposition method merely translates the images to their appropriate locations, the projective transformations in the Automatic method warp the four corners of the individual images to account for some slight variation in camera angle. (See reference image above for comparison). The final image shows the Automatic method repeated with the addition of vignette removal and geometric distortion correction. This method demonstrates the best alignment and blending of images but does still possess some overall warping. The spatial resolution of these orthomosaics were approximately 33 pixels per meter (px/m).



Figure 3.7: Aerial maps generated using Photomerge

Autodesk ReCap Photo v.19.1.1.2

The map data were also processed using Autodesk's ReCap Photo photogrammetry software². The same 55 image data set was uploaded to Autodesk ReCap's cloud server where the images were processed remotely. The reconstructed mesh was then downloaded when complete. The resulting mesh is shown below in Figure 3.8, oriented as a map view with a resolution of 3.5 px/m. The mesh successfully merged and blended images together for a majority of the surface but exhibits holes in the mesh near vertical features such as trees. Compared to the reference map, the proportions of the map appear accurate and successfully aligned. The algorithm used, however, appears to have automatically cropped noticeable portions of imagery from the west and east sides of the field.

² A single Autodesk ReCap Pro license costs approximately \$310 per year as of December 2019 [74]



Runtime = 60 min

Figure 3.8: Aerial map generated with ReCap Photo



Runtime = 16 min

Figure 3.9: Aerial map generated with WebODM and imported to Google Earth

OpenDroneMap

To explore alternative software to Photoshop and ReCap Photo, aerial map data from the Arboretum flight were processed using the open-source program WebODM [40]. To run the software, a Linux virtual environment was built using Docker. The virtual machine operates with all 6 cores of the i5-9600K CPU and with 28GB of dedicated RAM. Once running, the program creates a virtual node which can be accessed through the WebODM browser interface. From this GUI, batches of image data can be uploaded to the node and reconstructed into georeferenced 3D meshes and orthomosaics. The task results in both a 3D mesh and a geoTIFF with a resolution of 9.8 px/m. The geoTIFF image can then be imported and overlaid on existing map databases such as Google Earth as shown in Figure 3.9 above.

DroneMapper

A second open-source program, DroneMapper, was tested with the arboretum data set. The results were processed in the program's GUI, resulting first in a grayscale preview orthomosaic with rough alignment. DroneMapper then constructs a digital elevation model as shown in Figures 3.10 and 3.11. The DEM is displayed in the DroneMapper GUI with an elevation heatmap drawn as an overlay. A geoTIFF (18.8 px/m resolution) is exported upon DEM execution which can also be imported to existing maps. Finally, a 3D mesh is generated in the form of a PLY point cloud. As seen in the images below, the map and mesh qualities are relatively high with accurate alignment and reasonable elevation modeling. The elevation heatmap and 3D point cloud illustrates reconstructed vegetation from tall grass to trees. However, the reconstruction process clearly demonstrates bands of missing imagery along the edges of the models. These artifacts could be due to inadequate matching criteria for the edge images since the center appears to be well populated.



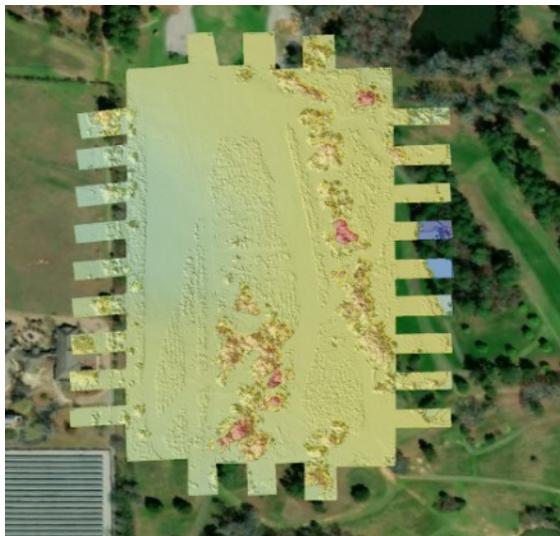
Runtime = 8 min (4 min low quality)

Preview



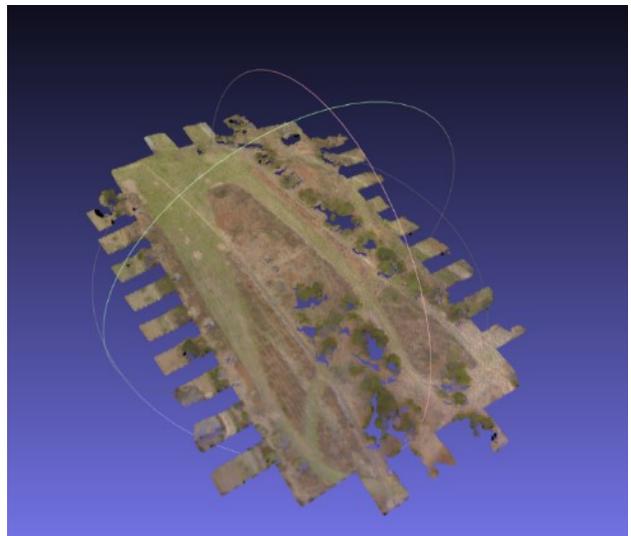
GeoTIFF (in Google Earth)

Figure 3.10: Aerial maps generated with DroneMapper



Runtime = 11 min (4 min sparse)

DEM heatmap



Runtime = 7 - 9 min

DEM point cloud (in Meshlab)

Figure 3.11: DEM's generated with DroneMapper

Mapping Software Comparison

As seen in Table 3.1 below, the open-source programs yield comparable if not better results than the proprietary software packages tested when using the same data set. Both WebODM and DroneMapper had substantially lower runtimes and maintained high output resolutions and geospatial data. The spatial resolution of the original JPEG images was approximately 33 px/m and was used to calculate the relative resolution for each run. These results show that these open source tools can be a valuable asset for affordable aerial surveying.

Table 3.1: Performance comparison of mapping software

Software	Est. Runtime (min)	Resolution (px/m)	Relative Quality	Geo-referenced	Open Source	Notes
Photoshop	30+	33.0	100%	No	No	Warped
ReCap Photo	60	3.5	15.9%	Yes	No	Holes
WebODM	16	9.8	48.5%	Yes	Yes	
DroneMapper	11	18.8	33.3%	Yes	Yes	Clipped edges

3.3.2 Photogrammetric Mesh Generation

To demonstrate the expansion of 2D mapping into higher depth, a collection of photogrammetry programs was compared to observe their quality and reliability of reconstructing 3D models and DEM's from still images. While some of these packages do not enable geotagging, the primary objective is to compare the speed and quality of 3D reconstruction and mesh generation. For each software package used, images were processed from a manmade construction mound located in a park in Bartlett, TN. The image data was collected as a 4K video by the Mavic Pro in a dynamic orbit around the terrain feature. The flight path in this case was executed by utilizing DJI Go's "Point of Interest" flight mode which enables a concentric orbit about the target feature at a constant altitude. Using the Python program described previously, the aircraft SRT logs were used to generate a group of 76 geotagged JPEG images from the flight.

Autodesk ReCap Photo v.19.1.1.2

The data were first processed by again using Autodesk ReCap photo. The images were uploaded and processed on the Autodesk servers, resulting in the mesh shown below in Figure 3.12. The model—most clearly seen in the untextured screenshot—clearly shows a distinct elevation change. Minor features are visible such as depressions where a ground vehicle left tracks on the mound. This algorithm again appears to automatically crop out regions beyond the center of the model, but the primary subject is shown in remarkable detail for the small number of images used.

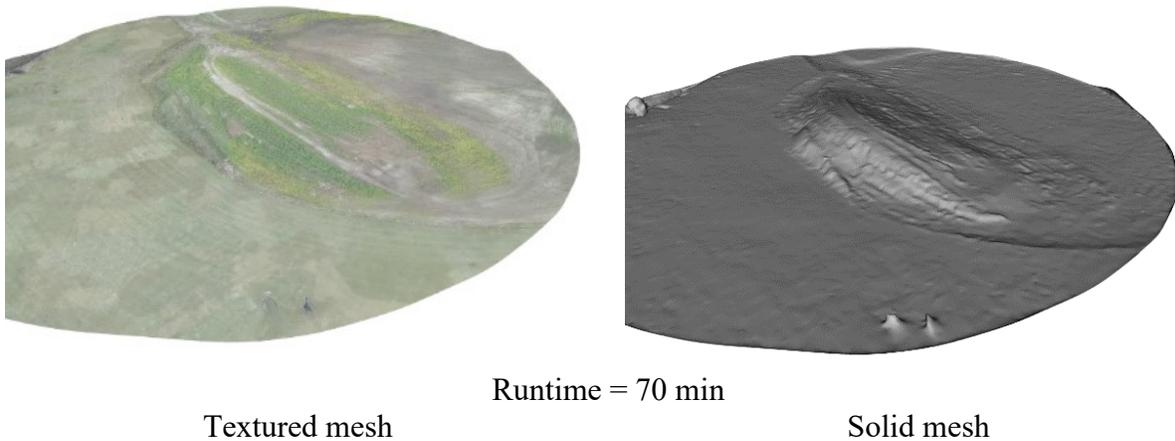


Figure 3.12: 3D Mesh generated with ReCap Photo

VisualSFM

Using the same set of data, a 3D pointcloud was generated using another open-source program, VisualSFM. Developed by Changchang Wu and based partially on Yasutaka Furukawa's Patch-based Multi-View Stereo (PVMS) software, VisualSFM is a GUI application that interactively performs 3D reconstruction using structure from motion (SFM) [41], [42].

The images were initially loaded into the workspace and matches between images were calculated. Once matches were achieved, the software generated a sparse point cloud by

estimating camera locations for each image and then reconstructing the matched points in 3D space. Once a successful sparse cloud was built, the software then filled in the model with additional detail by generating a dense reconstruction as seen in Figure 3.13.

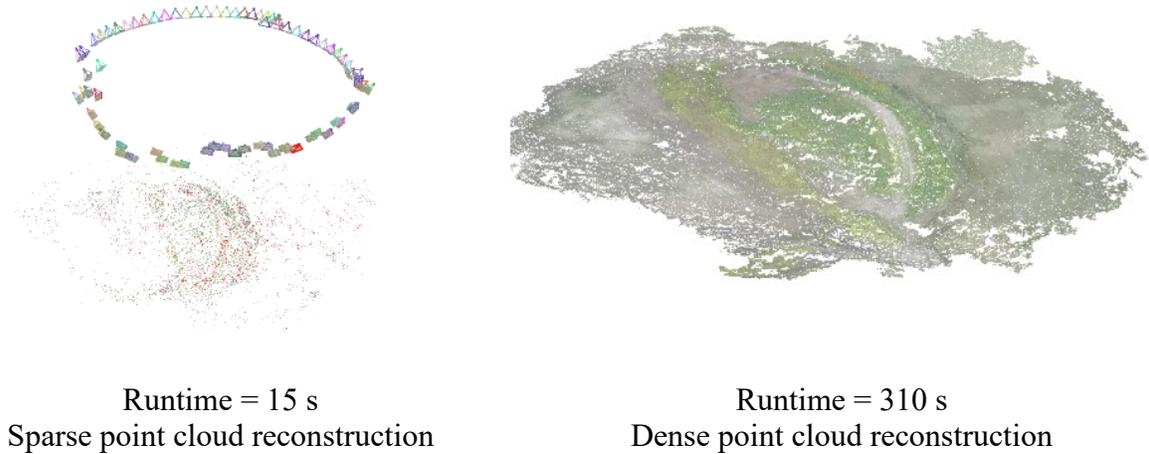


Figure 3.13: Point-cloud generation using VisualSFM

VisualSFM generates a point cloud but does not create surfaces to make a sealed mesh. To achieve this, the point cloud was exported to Meshlab where surfaces were stitched using screened Poisson surface reconstruction [43]. The resulting mesh (Figure 3.14) shows changes in elevation consistent with the actual feature but includes a significant amount of noise and false features along the mesh. False features are likely due to outlier cloud points which are not properly filtered by the screened Poisson surface reconstruction algorithm. Curling fringes occur along the edge of the model as well.

Additional tuning can be performed by varying the input parameters on the mesh reconstruction but appears to be highly dependent upon the amount of noise from the point cloud. In addition to this, adding image data to VisualSFM has frequently resulted in failed sparse reconstruction. The seemingly open-loop portion of this algorithm branches the reconstruction results into multiple models upon failure, making it difficult to merge additional

data into the system for higher quality photogrammetry. VisualSFM does not take GPS coordinates into account so results can often diverge. The processing time for this method is listed in Table 3.2.

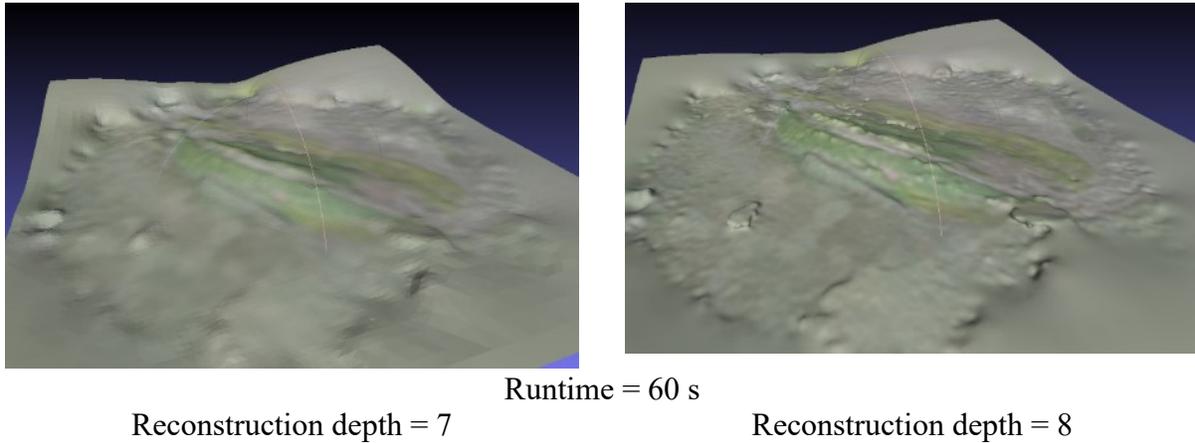


Figure 3.14: Mesh generation of VisualSFM point-cloud using Meshlab

Table 3.2: Runtime statistics for VisualSFM workflow

Task	Duration (s)
Compute matches	67
Sparse point cloud reconstruction	15
Dense point cloud reconstruction	310
Mesh generation (Meshlab)	60
Total	452

OpenDroneMap

This dataset was also processed by again using WebODM running on an Intel i5-9600K processor. The results below (Figure 3.15), rendered in Blender, demonstrate high quality mesh reconstruction similar to that generate by ReCap photo. The mesh covers a slightly wider area than ReCap photo, revealing additional terrain features such as the ditch and trees shown on the left side of the images below. By examining the untextured model it can also be observed that some faint radial ribbing occurs on the ground surface from the edges of the individual images. These periodic artifacts could possibly be due to vignetting or lens distortion from the still

images, slight errors in alignment, or slight phase offset from the emulated GPS data. By preprocessing the images to remove vignetting and distortion and including a larger data set, these artifacts may be attenuated or removed. Due to the retention of GPS information, this processing method also yields a georeferenced 2D orthomosaic which can be imported to existing maps (Figure 3.16).

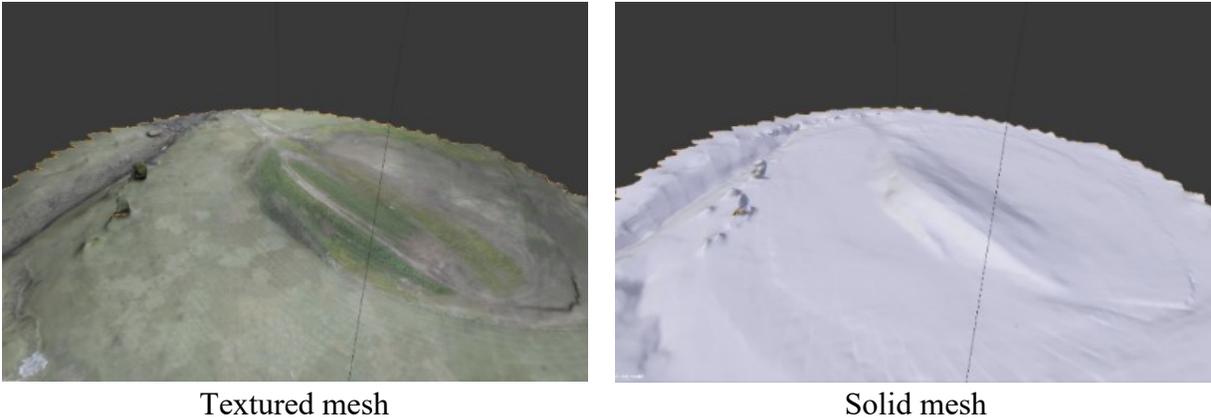


Figure 3.15: Mesh generation using WebODM



Figure 3.16: Georeferenced orthomosaic from WebODM overlaid on Google Earth

Photogrammetry Software Comparison

Table 3.3 below shows statistics of the 3D meshes generated from these software packages using the same data set. As previously stated, VisualSFM had the shortest runtime but provided unsatisfactory results due to its noisy open-loop reconstruction algorithm. While ReCap Photo produces a relatively high quality mesh, it is outperformed by WebODM which uses less than half of the runtime for a more dense reconstruction. In addition, WebODM is the only package tested which utilizes and maintains geospatial data during reconstruction. This further emphasizes the utility of open-source photogrammetry software as an alternative to commercial packages.

Table 3.3: Performance comparison of 3D reconstruction software

Software	Est. Runtime (min)	Vertices	Faces	Georeferenced	Notes
ReCap Photo	70	69.8k	139.4k	No	Automatically cropped
VisualSFM	8	447.9k	0 [†] /100.6k	No	Noisy reconstruction
WebODM	27	100.0k	184.7k	Yes	

[†]Initial reconstruction was only a point cloud, external meshing was required (Meshlab)

3.4 Thermal Mapping

While visible light surveying can prove valuable for visual inspection and basic image processing, it is desirable to collect mapping data in the infrared band to observe the thermal distribution of regions of interest. As previously described in Chapter 2, thermal imaging can prove useful for agricultural and geological applications such as crop health monitoring and industrial energy auditing. To demonstrate the implementation of an open-source thermal mapping UAS platform, a custom workflow was designed and tested to rapidly capture and process large amounts of imagery from a UAV equipped with an infrared camera.

3.4.1 Methods

For this experiment, the same DJI S900 hexacopter discussed in Chapter 1 was upgraded by integrating a Pixhawk 2 flight control unit and Here 2+ GPS receiver. A FLIR Vue Pro R

radiometric infrared camera was attached to the aircraft on a brushless 3-axis gimbal. This payload was powered from a programmable battery elimination circuit (BEC) which allowed the payload to be turned on and off remotely. The camera's tilt axis was controlled by an auxiliary pulse-width modulation (PWM) signal from the aircraft's receiver, enabling the operator to control the angle in flight—although the camera is typically locked downward for mapping operations. The FLIR Vue Pro R supports the MAVLink communication protocol [44] used within Pixhawk and was connected to receive a telemetry stream from the flight controller's secondary telemetry port. This enabled the aircraft's position, attitude, and time data to be automatically embedded within the thermal image files.

On June 17, 2019, the S900 thermal imaging system was flown over the University of Alabama Arboretum at an altitude of 120 m. The flight was an autonomous survey grid generated within the Mission Planner ground control software (GCS). This automation is trivial to program with existing open-source GCS software and optimizes the amount of data collected during a given flight. Throughout the 14-minute flight, the FLIR Vue Pro R was set to sequence mode, capturing a thermal image every second. As seen in the left image of Figure 3.17 below, the aircraft navigated along the planned grid pattern between the light blue waypoints stored in the aircraft's memory. The actual recorded flight path closely follows the desired path as illustrated by the red lines on the left image and the light blue rendering on the right image. The darker blue path on the right image of Figure 3.17 denotes the aircraft's recovery in return-to-launch (RTL) mode.

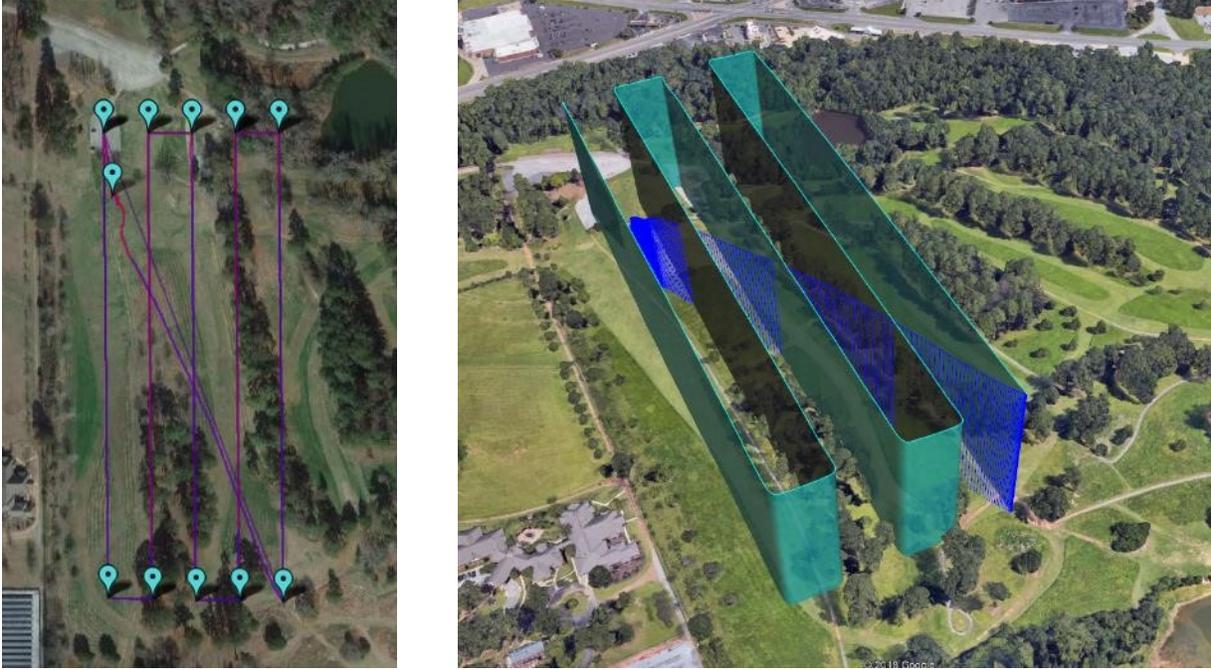


Figure 3.17: Autonomous flight path as seen in Mission Planner (Left) and imported to Google Earth Pro (Right)

3.4.2 Data Processing

To utilize the functionality of the 3D reconstruction used previously in this chapter, the radiometric data captured from this experiment had to be converted into linearized grayscale JPEG images. This was achieved with a custom MATLAB script which extracts the radiometric data from each image using a FLIR developer tool [45] and then maps the recorded pixel temperatures between the overall maximum and minimum values captured. The data were then exported as a series of grayscale JPEGs. The MAVLink telemetry data captured from the flight controlled were stored as EXIF metadata within each radiometric JPEG. This information was transferred to the processed images using ExifTool [46].

The resulting 481 linearized images—and preview images—from this experiment were then imported into WebODM and processed using the method described in Section 3.3. The result generated a georeferenced orthomosaic (Figure 3.18) with a resolution of 8.8 px/m as

illustrated below. This resolution of the orthomosaic has negligible loss from the original JPEGs after reconstruction. The 3D reconstruction task had a runtime of 3.5 hours using an Intel i5-9600K CPU and 32GB of RAM.



Figure 3.18: Heatmap orthomosaic overlaid on Google Earth Pro (left) and linearized orthomosaic (right) of thermal aerial survey generated from WebODM

3.4.3 Thermal Mapping Results

The resulting thermal map can then be analyzed using basic image processing techniques. With a known maximum and minimum temperature, the gray values can be mapped or interpolated for each pixel to display the corresponding measured pixel temperature. While a

more advanced analysis tool could certainly be developed, a custom MATLAB function³ was written to illustrate basic temperature inspection on the map generated as illustrated in Figure 3.19.

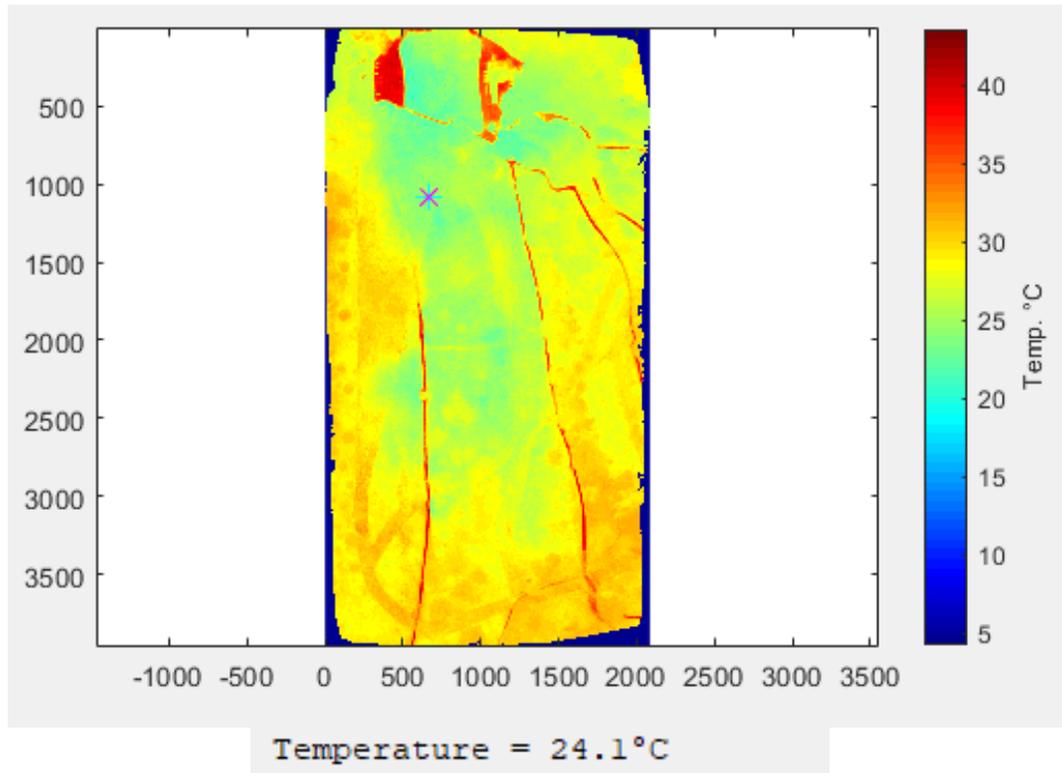


Figure 3.19: Example of thermal map analysis tool

To verify the accuracy of this thermal analysis workflow, a simple experiment was conducted using a thermal image with features of known temperatures. Figure 3.20 below—captured with the same FLIR Vue Pro R camera used during the aerial survey—was taken of two styrene cups: one filled with ice water and the other with hot water. The temperature of these cups as well as the ambient table temperature were recorded with a thermocouple and overlaid on this image for illustration.

³ See Appendix D for code

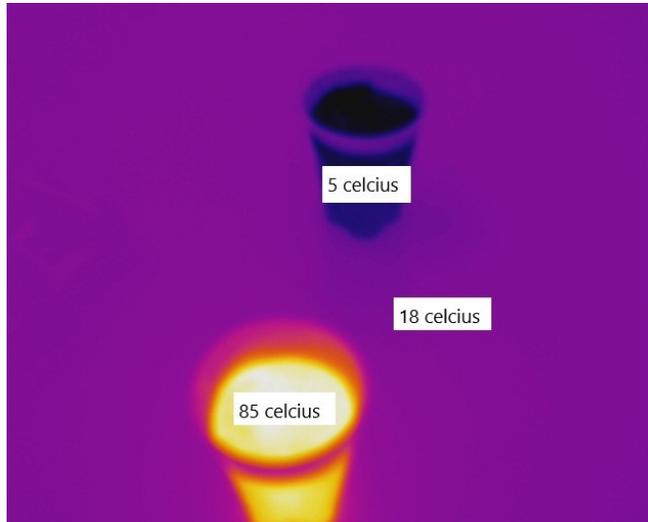


Figure 3.20: Calibration image with ice water container (top) and hot water container (bottom)

Using an abbreviated version of the thermal mapping procedure, this image was analyzed by first converting the image into a linearized grayscale bitmap image using the custom MATLAB script previously described. The resulting maximum and minimum pixel temperatures were found to be 80.96°C and 0.01°C , respectively. The image was converted from JPEG to TIF format and loaded into the viewing tool (Figure 3.21) as a surrogate thermal orthomosaic. Points on the hot and cold cups as well as the table were selected, resulting in the following measurements. As seen in Table 3.4, the measurements were accurate within approximately 6°C . The primary source of error in this experiment is believed to be a discrepancy between the fluid temperature recorded by the thermocouple and the surface emission of the objects seen through the infrared camera. The estimated feature temperatures show similar error when comparing the custom thermal analysis results to the camera manufacturer's own analysis tool. Because this error is found using both tools, it can be assumed that processing image data with the aerial mapping workflow does not significantly affect the quality of thermal data taken from the IR

camera. Additional improvements can be made by ensuring the settings within the FLIR ensure proper calibration to the environment being surveyed.

Table 3.4: Thermal Analysis Validation

Feature	Measured Temp. (Thermocouple)	Est. Temp. (FLIR Tools)	Error	Est. Temp. (Custom Tool)	Error
Cold Water Cup	5 °C	3.1 °C	1.9 °C	3.5 °C	1.5 °C
Hot Water Cup	85 °C	80.0 °C	5.0 °C	80.0 °C	5.0 °C
Background	18 °C	24.5 °C	6.5 °C	23.8 °C	5.8 °C

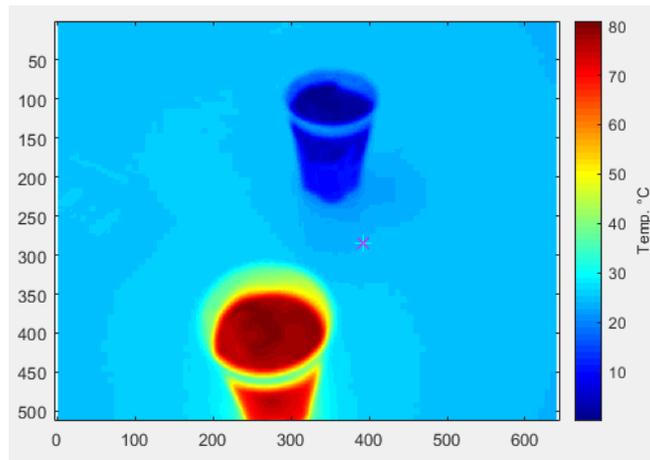


Figure 3.21: Hot and cold water calibration image as seen in thermal analysis tool

This surveying experiment demonstrates the ability of a small UAS to rapidly collect IR image data over a significant geographical area and synthesize the data into thermal maps using custom tools. Such a workflow could enable individuals and businesses to gather radiometric data for analysis without requiring proprietary software packages or external services.

3.5 Conclusions

The hardware and methods introduced in this section enable a low-cost workflow (summarized in Figure 3.22) to gather data from proprietary image sensors and process the results into useful maps or models with a quick turnaround. The workflow enables the utility of popular commercial products such as DJI drones and FLIR cameras to be further exploited for

enhanced data collection. The workflow demonstrated in this experiment can be used to rapidly and autonomously collect visible or thermal image data with a multicopter or fixed wing UAV and reconstruct a comprehensive orthomosaic which can then be inspected using image analysis tools. Such a process can provide an alternative method to costly software packages or UAS services thus giving businesses and individuals an efficient solution for rapid aerial surveying.

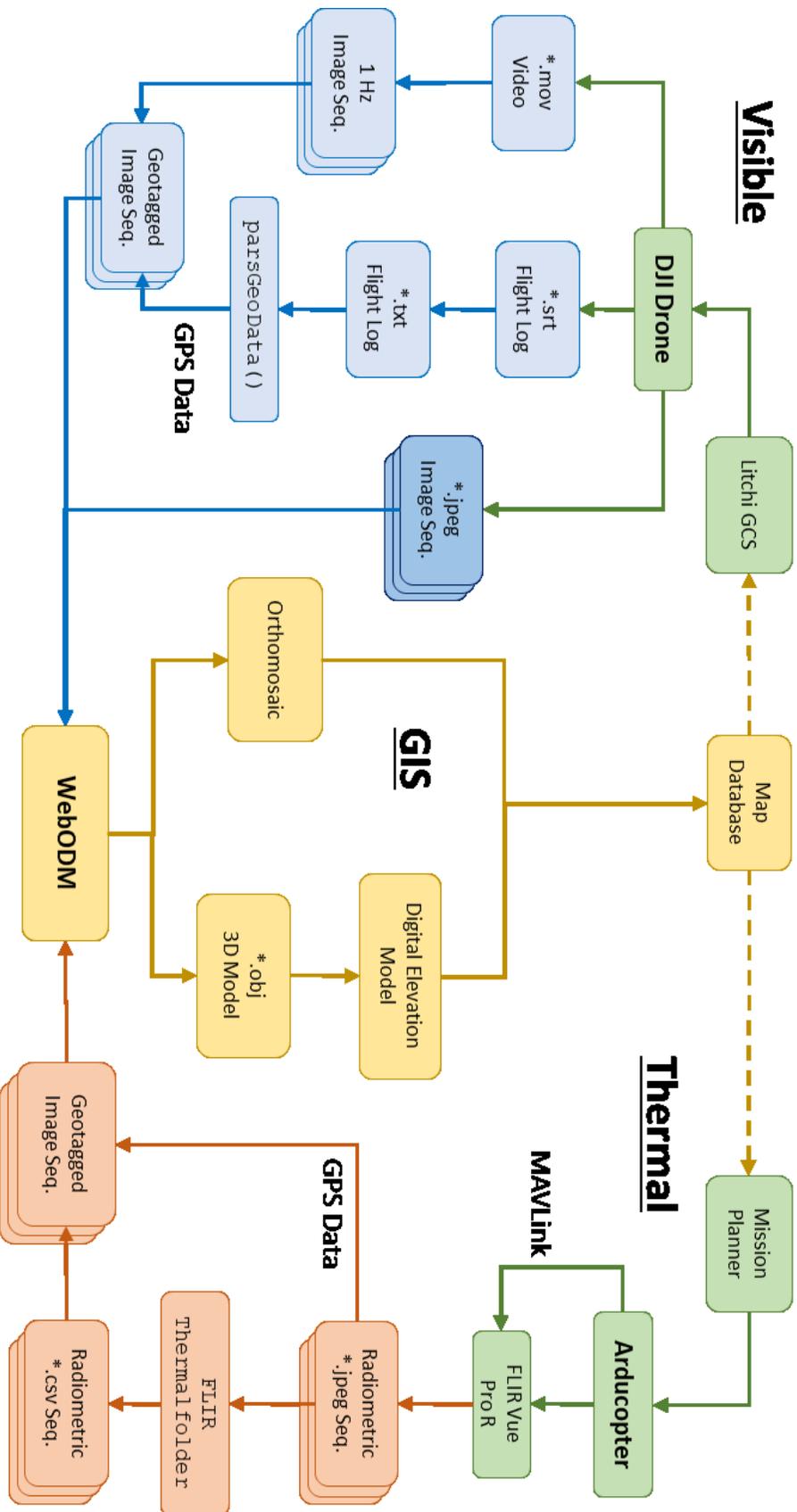


Figure 3.22: Multispectral Aerial Mapping Workflow Overview

CHAPTER 4

ULTRASONIC THERMOMETRY AND NAVIGATION

This chapter demonstrates the design and testing of an ultrasonic thermometry prototype for a novel building energy infiltration estimation system. The novel system demonstrates the possibility of a dual-purpose sensor that can provide thermal remote sensing as well as spatial navigation. The device utilizes continuous ultrasonic waves to measure the echo phase difference of a pair of 40 kHz transducers. This phase difference is used to estimate the surface temperature and thermal boundary layer thickness of an isothermal heated vertical wall in natural convection. The experiment successfully provides a model for estimating wall surface temperature from data gathered from the prototype sensor. While intended for use as a thermometric sensor, this system also demonstrates the ability to detect changes in target range, yielding the opportunity of an ultrasonic ranging sensor for navigation.

4.1 Introduction

In 2010, the buildings sector represented over 40% of the United States' primary energy consumption [47]. This indicates that buildings contribute a significant portion of energy usage and improving the efficiency of these buildings can reduce overall consumption. To estimate the energy efficiency of a building, it is important to understand the heat flux surrounding the structure's surface. By quantifying the heat flux entering and exiting the building surfaces, the net gain or loss in thermal energy can be estimated [48]. Reliably measuring this energy flow can

gain insight into the building's HVAC efficiency and its ability to retain heat in cold environments and insulate against heat in hot environments.

Existing technology for measuring building infiltration is by using blower door testing and tracer gas methods [51], [52]. Both methods, however can require extensive preparation and cannot be used while the building is occupied. Blower door methods are suitable primarily for larger structures and tracer gas methods do not enable individual leaks to be located. It is desirable to have an alternative method that can be used more conveniently—with or without occupants—and can detect the specific locations of infiltrations. The proposed solution is the development of a mobile ultrasonic thermometry device which can detect air leaks and estimate the heat flux in and out of a structure. This device would utilize a series of ultrasonic transducers to measure the difference in echo phase and time-of-flight using a sonic waveform to estimate fluid properties. By collecting air temperature and estimating thermal boundary layer thickness, the infiltration at any region on the building's surface can be measured. Once collected, the data will enable the estimation of overall heat flux to quantify the building's energy efficiency. Such a device could be mounted on an sUAS for enhanced thermal remote sensing for use in building inspection and other operations.

The objective of these experiments is to characterize the performance of a pair of 40 kHz ultrasonic transducers and demonstrate the feasibility of using them in a prototype for ultrasonic thermometry. The performance of two types of transducers were measured with varying distance, angle and frequency. A prototype ultrasonic thermometry (UST) device was constructed using two 40 kHz transducers and a custom amplifier circuit. An experiment was then conducted with a heated wall to observe the correlation between the sensor's echo phase offset and wall temperature.

4.2 Theory

The thickness of a thermal boundary layer in laminar natural flow on a vertical heated wall is given as [49]

$$\delta = 3.936 \left(\frac{0.952 + Pr}{Pr^2} \right)^{1/4} \left(\frac{1}{Gr_x^{1/4}} \right) x \quad (4.1)$$

The Grashof number (Gr) as a function of vertical distance from the wall base (x) is given as

$$Gr_x = \frac{g\beta\Delta T}{\nu^2} x^3 \quad (4.2)$$

where $\Delta T = T_s - T_\infty$ is the difference between the wall surface temperature (T_s) and ambient air temperature (T_∞) as measured by the UST. Assuming the thermal expansion coefficient for an ideal gas is

$$\beta = \frac{1}{T_\infty} \quad (4.3)$$

Equation 4.3 above simplifies to

$$\delta = 3.936 \left(\frac{(0.952 + Pr)T_\infty \nu^2}{g\Delta T Pr^2} x \right)^{1/4} \quad (4.4)$$

If it is then assumed that the Prandtl number is $Pr = 0.7$, Equation 4.4 becomes

$$\delta(x, T_\infty, \Delta T) = 3.936 \left(0.333 \frac{T_\infty \nu^2}{\Delta T} x \right)^{1/4} \quad (4.5)$$

Assuming a kinematic viscosity of approximately $\nu \cong 12.7e - 6 \text{ m}^2/\text{s}$, Equation 4.5 reduces to

$$\delta(x, T_\infty, \Delta T) = 0.011 \left(\frac{T_\infty}{\Delta T} x \right)^{1/4} \quad (4.6)$$

Thus, given the estimated vertical position from the wall base (x), the ambient air temperature (T_∞), and the differential temperature (ΔT), the thickness of the isothermal wall's boundary layer can be estimated. The ambient air temperature can be measured using a thermocouple mounted

on the UST device. Differential temperature is the primary goal of the UST device and is estimated by analyzing the phase offset of the echoed sonic waveform. Given a continuous sinusoidal waveform, the angular phase offset of the return signal is computed numerically in MATLAB by implementing a fast Fourier transform (FFT) on each signal. The angular difference between the peaks of these spectra are used to measure the phase difference (ϕ). Once data is collected from experimentation, a model will be generated to then compute the differential temperature given the ambient air temperature and phase difference.

4.3 Methods

4.3.1 Ultrasonic Transducer Characterization

In order to observe the performance of off-the-shelf ultrasonic transducers, two pairs of commercial 40 kHz transducers were analyzed. One set was taken from an HC-SR04 ultrasonic ranging module (Figure 4.1, left) and another set were PUI Audio UT/R-1640K-TT-2 transducers. Each transducer was mounted horizontally on static mounts as shown in the right images of Figure 4.1 below. The faces of the mounts were coated with foam rubber around the transducers to reduce undesired echoing.

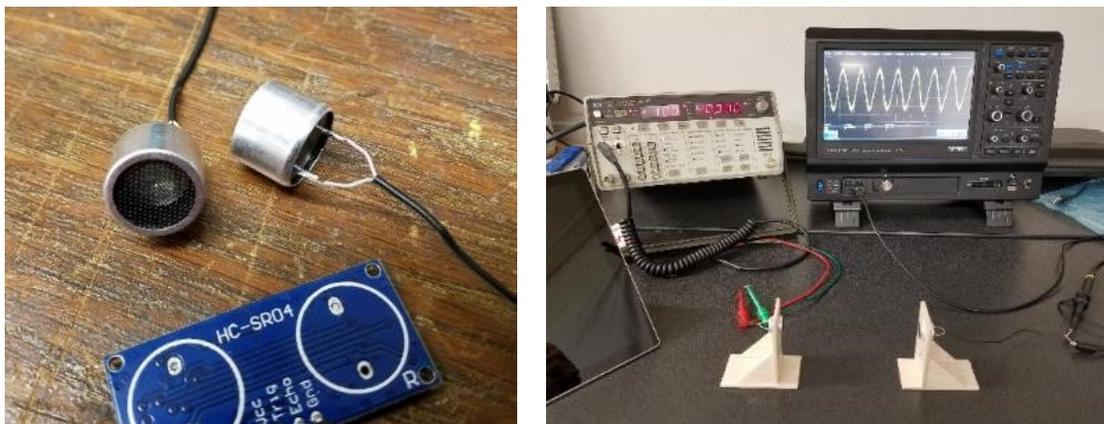


Figure 4.1: 40 kHz ultrasonic transducers (left) and setup for preliminary ultrasonic transducer analysis (right)

For each test, the transmitter was connected to an HP 4935A Transmission Test Stand which was used to generate a continuous sinusoidal waveform at the desired frequency and power. The transmitter and receiver were then connected to separate channels on a Teledyne Wavesurfer 3022 oscilloscope.

The spatial power decay of a 5 dBm (3.16 mW) 40kHz continuous sinusoidal signal was measured from the receivers at distances varying from 2.5 to 100 cm. The resulting performance for this and the following experiments were calculated using the power ratio equation below:

$$\frac{P_{rx}}{P_{tx}} = \frac{V_{rx}^2(\text{RMS})}{V_{tx}^2(\text{RMS})} \quad (4.7)$$

The next experiment observed the response of the receivers at a distance of 10 cm listening to a 5 dBm signal. The signal was swept from 37 kHz to 44 kHz. This enabled estimation in the resonant frequencies and bandwidth of the sensors. The transducers' directional responses were analyzed by measuring the response gain as a function of angle. The receivers were placed at a distance of 20 cm from the transmitters and sequentially rotated $\pm 90^\circ$ from center. The waveform used was a 5 dBm 40 kHz sinusoid.

4.3.2 UST Prototype Design

A prototype UST device was then constructed to enable consistent data collection during thermometry testing. The mount was designed to hold one UT-1640K-TT-2 transmitter and one UR-1640K-TT-2 receiver at a 12 cm spacing. The mount was fabricated using 3D printing and holds a circuit board for the UST amplifier circuit (Figure 4.2). The front face of the device was covered in foam rubber (not shown) during experiments to attenuate undesired echoes.

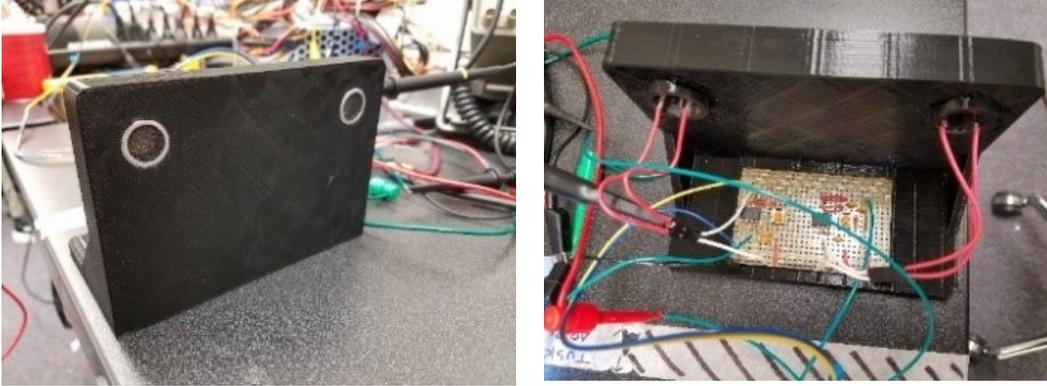


Figure 4.2: Ultrasonic thermometry (UST) prototype

For each transducer, a filtered 10x gain amplifier was constructed according to the schematics shown in the Appendix [50], [51]. The transmitter was connected to the HP 4935A Transmission Test Stand with an input power of -1.5 dBm (0.71 mW) which was then boosted to an amplitude of approximately 10V. The amplified output of the transmitter and receiver circuits were then connected to separate channels on the Wavesurfer 3022 oscilloscope.

4.3.3 Heated Wall Ultrasonic Thermometry

An experiment was conducted to test the UST prototype's ability to detect fluid temperature change from a smooth wall [52], [53]. The UST prototype (Figure 4.3) was mounted perpendicular to a ceramic electric wall-mounted room heater at a distance of 1.5 m. Two thermocouples were used to record both the ambient air temperature halfway to the wall and the surface temperature on the wall-mounted heater. Foam rubber was used to enclose the test area to eliminate echoing and clutter from the surrounding environment.



Figure 4.3: Heated wall ultrasonic thermometry experiment construction

The transmitter was connected to the transmission test stand set to a continuous sinusoidal waveform at 40 kHz and a power of -1.5 dBm. The amplified outputs of the transducers were both monitored on separate channels on the oscilloscope.

The wall heater was then turned on and data were collected every minute while the wall heated and then cooled. The heater was turned on for 20 minutes and then turned off to cool for 55 minutes. For each test, the ambient air temperature and surface temperature were recorded from the thermocouples. Additionally, the average surface temperature of the wall was captured using a Fluke handheld infrared camera. A sample waveform of each transmit and receive signals were recorded to a CSV file for postprocessing. The signal data were processed using a custom MATLAB script⁴ which measures the phase offset angle between the transmitted and received signals for each data point.

⁴ See Appendix E for code

4.4 Results

4.4.1 Ultrasonic Transducer Characterization

The range drop-off for each of the two 40 kHz ultrasonic transducers was recorded and the results are shown in Figure 4.4.

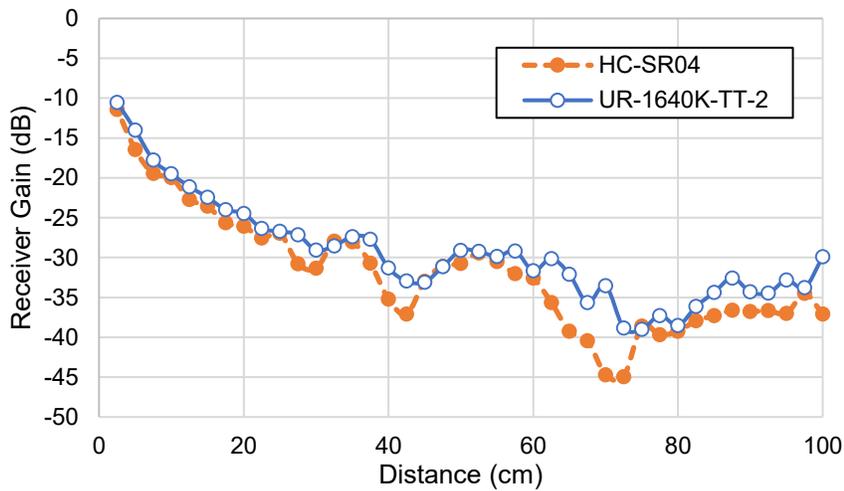


Figure 4.4: Acoustic intensity ultrasonic transducers

As seen in the plots above, the signal power for both transducers decreases with range at a similar rate.

The results of the frequency sweep comparison are shown in Figure 4.5 below.

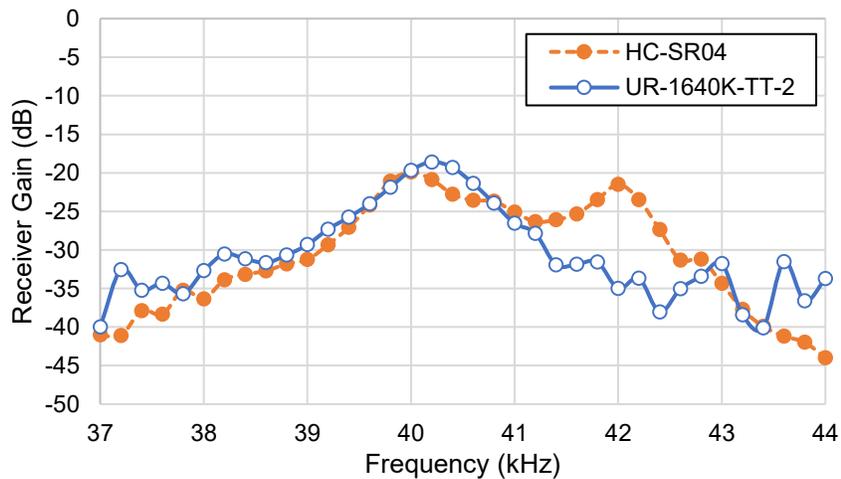


Figure 4.5: Frequency response of ultrasonic transducers

The HC-SR04 response indicates a pair of resonant frequencies at which the receiver response is maximized. The peak response for the HC-SR04 occurs at 40.0 kHz and 42.0 kHz. This transducer's 10 dB bandwidth is only 3.9 kHz, ranging from 38.9 to 42.8 kHz. The UR-1640K-TT-2 transducer features a single peak response at 40.2 kHz. Its 10 dB bandwidth is 3.1 kHz, ranging from 38.5 to 41.6 kHz.

As seen in Figure 4.6 below, the directional nature of the transducers is evident as gain decreases with rotation. However, the UR-1640K-TT-2 transducer appears to have a more radially constant gain compared to the HC-SR04. Because of this property and its singular peak resonant frequency, the UR/T-1640K-TT-2 transducer was selected for use in the thermometry test.

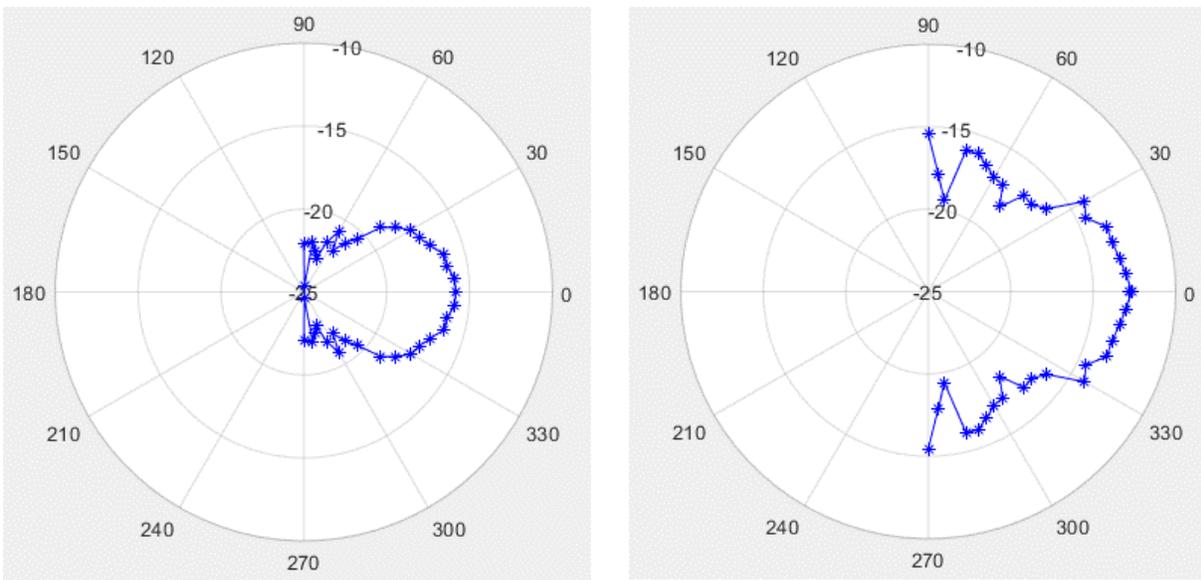


Figure 4.6: Directional responses of HC-SR04 (left) and UR-1640-TT-2 (right)

4.4.2 Heated Wall Ultrasonic Thermometry Experiment

During this experiment, the surface temperature of the heated wall and the ambient air temperature were recorded. The plot in Figure 4.7 shows the ambient air temperature and wall surface temperature profiles during heating and cooling of the wall heater during the

thermometry experiment. After 20 minutes of heating, the wall surface temperature reached a peak of 62 °C and dropped to 26 °C after 55 minutes of cooling. The ambient air temperature at the midpoint gradually increased from 22.5 °C to 24 °C during the experiment due to proximity.

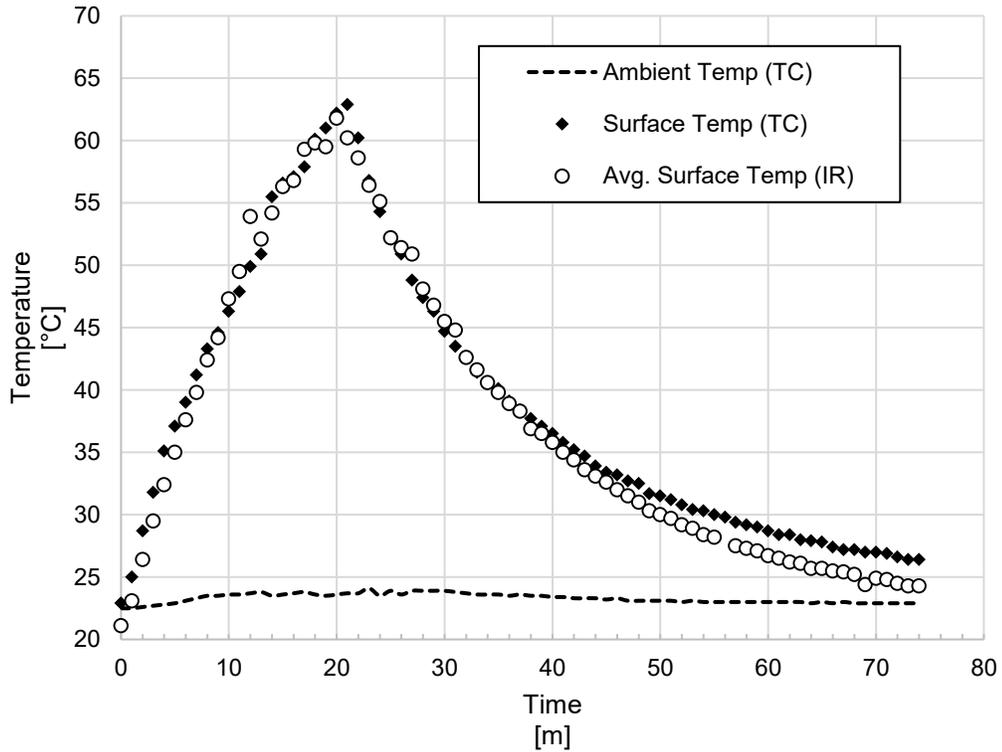


Figure 4.7: Heated wall transient temperature profile

The phase offset for each sample point was calculated using the MATLAB script, resulting in the plot in Figure 4.10. This plot suggests that the measured phase offset of the return signal can be used to represent the average fluid temperature difference between the sensor and the heat source. This illustrates the correlation between the measured echo phase difference and the relative temperature denoted as wall surface temperature minus ambient air temperature. The correlation appears linear⁵ but begins to decrease and diverge toward higher temperatures.

⁵ Residuals are plotted in Figure A.2 in the Appendix

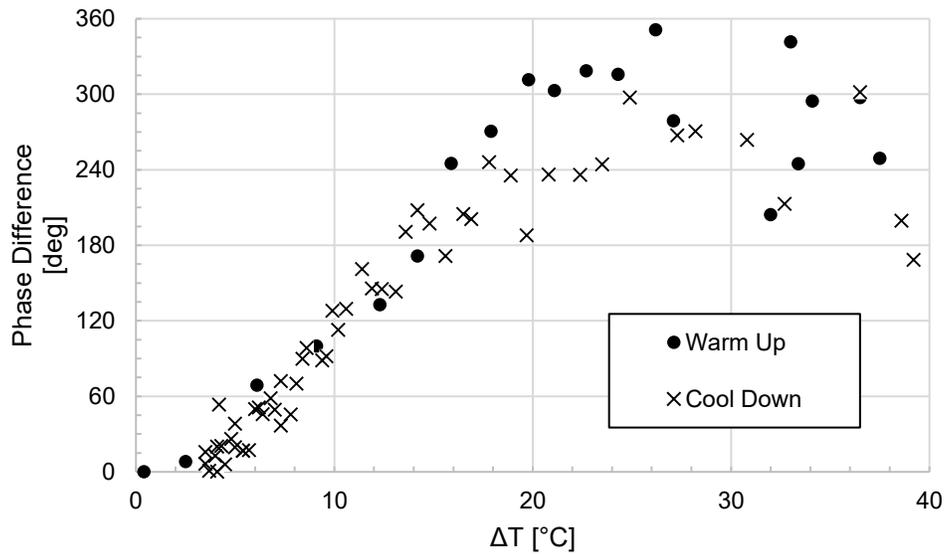


Figure 4.8: Correlation of echo phase difference and differential temperature

The phase response of the sensor can be approximated as a linear function shown in Figure 4.11 below. The linear fit is an acceptable approximation from room temperature (0 °C/K relative) to 42 °C (20 °C/K relative). Beyond this temperature, the variation of the phase difference increases and exceeds the 95% confidence interval. The cause of this diversion is believed to be due to the fluctuation in phase and amplitude of the return signal caused by turbulent convection currents near the heated wall. This phenomenon is visible on the oscilloscope during operation. More accurate readings can be taken if additional averaging or continuous phase calculation is performed during data acquisition in this more turbulent region.

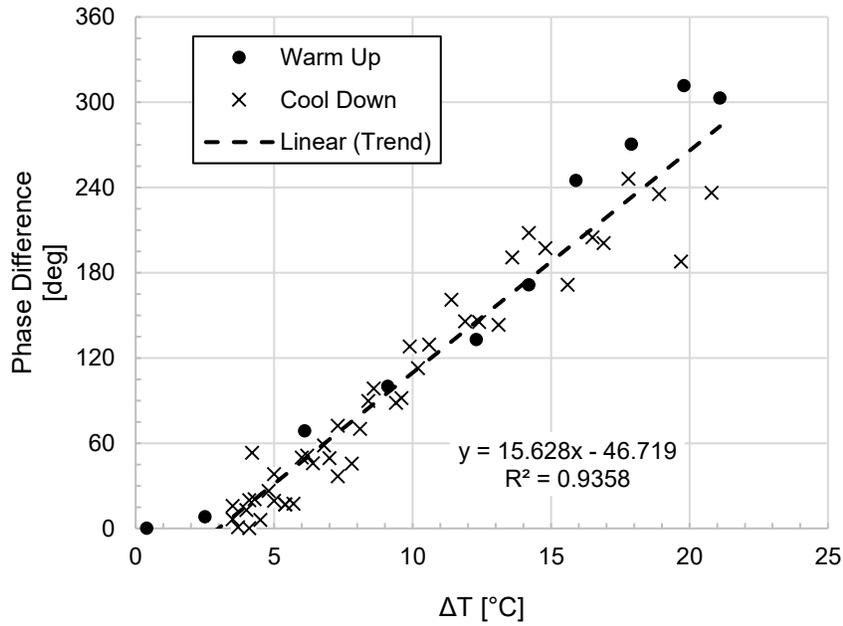


Figure 4.9: Linear approximation of phase-temperature correlation

Using the linear trend line from the data collected, an equation for the differential temperature is found to be:

$$\Delta T(\phi) = \frac{\phi + 46.72}{15.63} = 0.064\phi + 2.99 \quad (4.8)$$

where ΔT is in $^{\circ}\text{C}$ or K and ϕ is in degrees. Substituting into equation 4 yields a new equation for the boundary layer thickness:

$$\delta(x, T_{\infty}, \phi) = 0.011 \left(\frac{T_{\infty}}{0.064\phi + 2.99} x \right)^{1/4} \quad (4.9)$$

For the experiment performed, the vertical distance (x) from the wall base was 7.5 cm or 0.075 m. Using this value and using the temperature and phase difference from the heated wall experiment, the thermal boundary layer thickness profile is plotted in Figure 4.12.

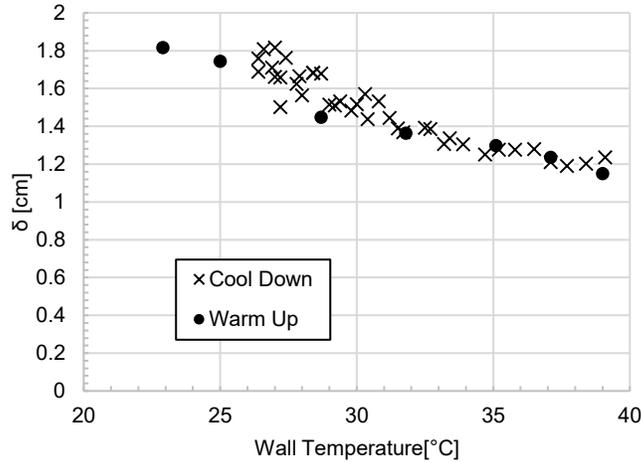


Figure 4.10: Thermal boundary layer thickness estimation

4.5 Application for Ultrasonic Navigation

Standard 40kHz ultrasonic transducers have often been used for robotic applications and have recently been incorporated into sensor suites onboard small UAS. The small sensors typically operate using time-of-flight measurement that estimates the range between a pair of transducers and a target object up to a few meters away using a simple pulse acoustic waveform [55]. The time delay between transmitting and receiving the pulse between a pair of transducers or a single transducer is used to estimate range. This capability is particularly suitable for estimating altitude upon takeoff and landing for multirotor sUAS. Already implemented on many commercial UAVs, ultrasonic rangefinders can augment the drone’s ability to takeoff and approach for landing with higher precision than solely using the flight control system’s IMU and GPS. Similarly, this ranging technique can be employed to aid in detecting obstacles during horizontal movement for multirotor UAS. Since ultrasonic sensing can be used in light and GPS-denied environments, these abilities are especially useful in the progression of nocturnal UAS navigation.

As demonstrated in the ultrasonic thermometry experiment previously described, even low-cost ultrasonic transducers can provide adequate transmission and reception required for advanced signal detection. By utilizing a digital receiver, precise measurements could be collected with low-cost ultrasonic transducers rather than the simple pulse time-of-flight method. Adapting existing radar signal processing methods to acoustic waveforms could enable these ultrasonic transducers to detect range, velocity, and possibly direction of targets.

Matched filtering

By implementing existing signal processing techniques, the quality of detected acoustic signals can be improved by optimizing the signal to noise ratio (SNR) of the ultrasonic waves. One common method is by implementing a matched filter [25]. This method essentially correlates the return signal with a known template signal to reduce noise. The received echo is convolved with a conjugation of the time-reversed input signal to greatly increase the SNR in the presence of noise. As illustrated in the examples in Figures 4.13 and 4.14, this is an effective method for reducing noise in simple pulse and other waveforms. As the velocity of the target increases, it may be necessary for the matched filter to be adjusted to prevent Doppler mismatch. Alternatively, the sensor could apply a number of preset matched filters in a series to ensure moving targets are detected more readily.

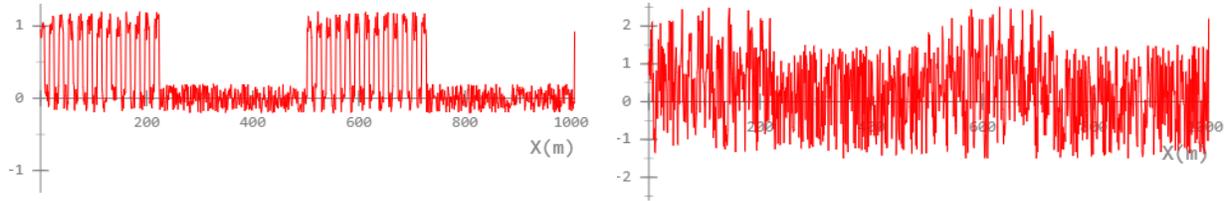


Figure 4.11: Simulated simple pulse signals with low noise (left) and high noise (right) [56]

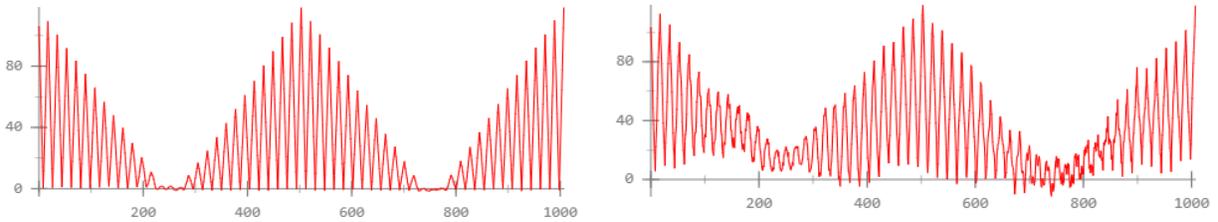


Figure 4.12: Simulated matched filters of corresponding simple pulses with low noise (left) and high noise (right)

Pulse Compression

The simple pulse waveform that is typically used in ultrasonic rangefinders is suitable for estimating range to targets as the range resolution is directly related to the duration of the pulse. However, simple pulses do not have much Doppler resolution and thus are not suitable for measuring the velocity of moving targets.

To combine range and velocity measurement, a digital ultrasonic system could generate a more complex waveform such as a linear frequency modulation (LFM) chirp. This type of waveform would enable the transducer to estimate both the range and velocity of target objects concurrently.

Phased Array

By incorporated a suite of multiple ultrasonic transmitters and receivers onboard the aircraft, the overall power of the acoustic signal can be boosted, thus increasing the effective range of the system. With multiple transducers, the system could also incorporate phased array techniques by slightly varying the time delay of each pulse. While this would require advanced

digital circuitry for control, it could enable beam steering for collecting multiple range estimates along a spatial line or a plane. Such a method could be used for generating low-fidelity acoustic spatial mapping or thermal boundary layer mapping of target surfaces (Figure 4.15). When used concurrently on a single platform, this would enable a UAS to navigate as well as collect spatial and thermal remote sensing data.

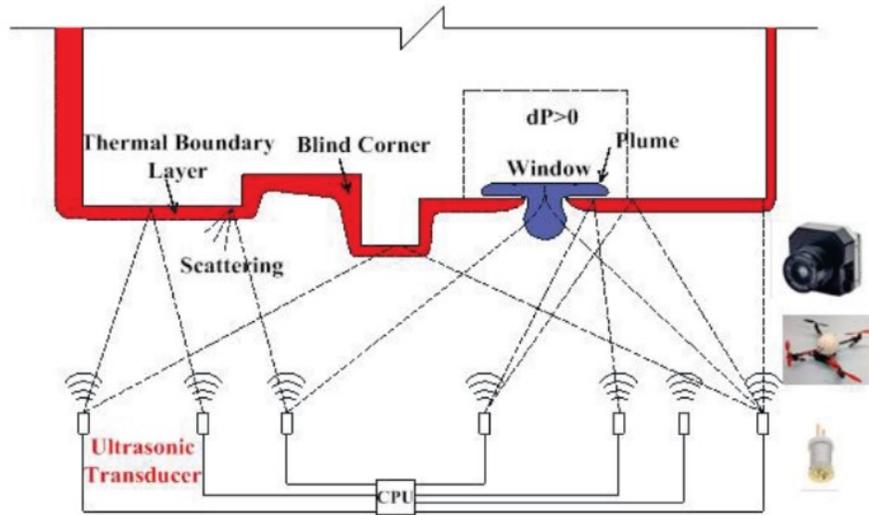


Figure 4.13: Concept for ultrasonic thermometry array [47]

Sample Calculations

Assuming a UR-1640K-TT-2 ultrasonic transducer was used for a digital ranging sensor, the range and Doppler resolution can be estimated for a hypothetical ultrasonic ranging sensor.

The range resolution (ΔR) of a simple pulse waveform is [57]:

$$\Delta R = \frac{c\tau}{2} \quad (4.10)$$

Where c is the speed of sound in the ultrasonic medium and τ is the pulse width in seconds.

Assuming operation in dry air at sea level and a temperature⁶ (T) of 25 °C, the speed of sound is

⁶ The ambient air temperature can be calibrated continuously by utilizing a thermocouple onboard the aircraft

assumed to be approximately 345 m/s. This scenario assumes an ideal gas constant (R) to be 286 J/kgK and a specific heat ratio (γ) of 1.4 for air.

$$c = \sqrt{\gamma RT} = \sqrt{(1.4)(286)(298.15)} \cong 345 \text{ m/s} \quad (4.11)$$

For use as a navigational and remote sensing device onboard a small UAV, it is assumed that the desired minimum ambiguous range is approximately 0.2 m. This distance represents closing distance for a small UAV during landing approach or takeoff. Using this minimum range, the pulse width can be designed:

$$\tau = \frac{2R}{c} = \frac{2 * 0.2}{345} = 1.16 \times 10^{-3} \text{ s}$$

The range resolution of an LFM chirp is calculated as [58]:

$$\Delta R = \frac{c}{2\beta} \quad (4.12)$$

Where β is the transducer's bandwidth—previously calculated to be 3.1 kHz. Thus the range resolution of a hypothetical ultrasonic LFM waveform would be approximately 6 cm.

$$\Delta R = \frac{345}{2 * 3.1 \times 10^3} = 0.056 \text{ m} \cong 6 \text{ cm}$$

The Doppler resolution (ΔF_D) of this LFM chirp is calculated as the inverse of the pulse width [25]:

$$\Delta F_D = \frac{1}{\tau} = \frac{1}{1.16 \times 10^{-3}} = 862.5 \text{ Hz} \quad (4.13)$$

The wavelength (λ) of the acoustic signal is computed as:

$$\lambda = \frac{c}{f} = \frac{345}{40.2 \times 10^3} = 8.6 \times 10^{-3} \text{ m} \quad (4.14)$$

The corresponding measurable velocity resolution (Δv) can then be estimated as:

$$\Delta v = \Delta F_D \cdot \lambda = (862.5)(8.6 \times 10^{-3}) = \pm 7.4 \text{ m/s} \quad (4.15)$$

While the velocity resolution in this example is not ideal, it could be used to augment existing attitude control algorithms such as the popular extended Kalman filter (EKF) that operate using sensor fusion. The waveform could certainly be adapted to suit the operational requirements of the UAV and could even be adapted for different portions of the flight as needed.

Additional processing methods can be implemented to further increase the signal to noise ratio of this system and address issues such as range-Doppler coupling. These sample calculations demonstrate that common 40 kHz ultrasonic transducers could be utilized to provide range and velocity measurement by using more advanced signal processing techniques than simply time-of-flight.

4.6 Conclusion

The experiments in this project demonstrate that standard off-the-shelf 40 kHz transducers can be used for detecting fluid temperature surrounding a smooth. The experiment showed that there exists a linear relationship between return signal phase difference and average fluid temperature of air for temperatures ranging from room temperature to 40 °C. By incorporating additional transducers in an array and further developing data collection and analysis, this prototype can certainly become a viable option for using ultrasonic thermometry for estimating building energy infiltration.

As shown in the sample calculations, these off-the-shelf ultrasonic sensors could be used to help UAVs navigate in nocturnal and GPS-denied environments by adapting existing radar signal processing methods. A hypothetical ultrasonic array could enable an aircraft to simultaneously navigate complex environments in darkness while also collecting useful information of the thermal environment.

CHAPTER 5

LASER-AUGMENTED FEATURE TRACKING (LAFTR)

The project addressed in this chapter seeks to demonstrate the feasibility of a compact Laser Augmented Feature TRacking (LAFTR) system for integration on UAV platforms. The goal of LAFTR is to provide an inexpensive and computationally efficient LIDAR alternative by augmenting existing optical feature tracking with laser light. This will be achieved by painting a scene with a laser projection pattern. The laser features are then captured using a filtered CMOS camera offset from the projector. The laser features can be generated using passive optics tailored to optimize the detection of desired target objects. Using an image processing algorithm, the parallax, scale, and movement of the features will enable the UAV to detect position and estimate the depth of the surrounding objects and navigate accordingly. While not as high fidelity as LIDAR, this method is a simple solution to enhance a standard RGB camera's ability to detect objects in a nighttime environment without needing to store and process large volumes of spatial data. This document describes the methods, application, and preliminary experiments of this novel system.

5.1 Introduction

Advances in the miniaturization and reliability of inertial sensors have contributed to small UAS' robust flight control. By augmenting these advanced control systems with additional sensors such as GPS, ultrasound, or LIDAR, UAV's can autonomously navigate in increasingly complex environments. However, these costly sensors have significant disadvantages for sUAS use. While GPS provides absolute position data, such navigation is often unusable indoors

without signal repeaters or in GPS denied areas. LIDAR provides high spatial detail but is expensive, bulky, and requires advanced signal processing and moving optical hardware. Another alternative is the implementation of dual cameras for stereographic image processing and generating 3D structure from motion [59]. Such methods require multiple cameras and rely on 2D feature recognition using ambient light. Recent developments in optical sensing allow complex navigational features such as target tracking and obstacle avoidance to be done with a single “monocular” digital camera sensor [60]. LAFTR seeks to utilize the simplicity of the monocular camera sensor (commonly present on current commercial sUAS) but expands its capability with an active collimated light source for improving feature tracking.

It is hypothesized that illuminating the observed scene with laser dots of a predetermined wavelength will enable rapid feature tracking comparable to SURF or SIFT methods. While traditional feature-based methods may yield higher resolution 3D models, they rely on sufficient ambient light and a significant amount of continuous image data in order to properly register the 3D environment. It is hypothesized that LAFTR can provide a rapid low-fidelity obstacle avoidance solution operable in little to no ambient light. If developed further, LAFTR could be integrated onto the UAS to enable basic obstacle detection without necessitating complex hardware or processing capability (Figure 5.1) [61], [62].

For the efficient use of UAS, and to safely expand their operational extent, it is desired that the aircraft can detect small and thin obstacles. Thin structures such as power lines, radio antennas, and poles pose significant limitation as to where UAVs can be flown. Feature-based 3D reconstruction techniques often fail to detect thin features as such algorithms track consistent features and often discard thin, independent features with high parallax. To ensure thin features are detectable by the LAFTR system’s monocular camera, a variety of laser projection patterns

can be utilized to detect desired geometry [63]. The laser projectors can be set to passively generate specified patterns by using a combination of lenses and diffraction films unlike LIDAR which requires a rotating optical sensor [64]. For instance, a vertical stripe pattern can be projected to track power lines and a horizontal stripe pattern can be projected to track poles and antennas [65]. Emerging technology for compact laser emitters could be utilized such as vertical-cavity surface-emitting lasers (VCSELs) to maintain a low profile for integration on small UAS [64]. Utilizing laser projection patterns can eliminate the need of high-power flood lights for feature tracking. Additionally, if the laser projector is used in the IR or other invisible band, UAS could be used at night without disrupting civilian activity on the ground or manned aircraft in the air.

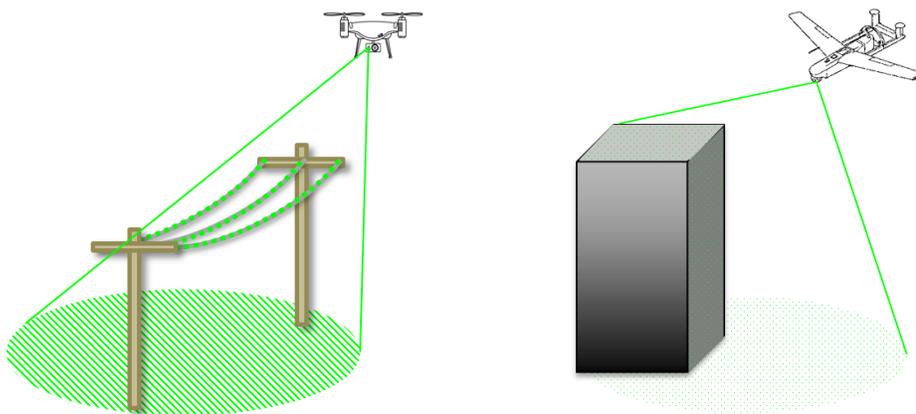


Figure 5.1: CONOPS of laser projector illumination on aerial platforms

The goal of this experiment is to demonstrate the feasibility of a LAFTR prototype system. A single stationary laser dot projector and camera will be used to identify and track targets along the horizontal plane using two projection patterns and their corresponding algorithms. The first method will utilize a periodic dot grid pattern for general target tracking and ranging. The second method will use a periodic vertical stripe pattern for detecting and ranging

thin horizontal features. It is assumed that the vertical stripe method could easily be adapted for detecting vertical features using horizontal stripes.

5.2 Theory

Concept

The general concept of LAFTR is to use a single digital camera for capturing and processing images with one or more laser pattern projectors. The laser projectors are offset from the camera position to enable a degree of range detection. This setup can be operated in a basic “free” mode for motion tracking outdoors, or an “indoor” mode for tracking targets and estimating their range by using reflections from the background.

Free Target Detection

In its most basic configuration, LAFTR simply projects a dot matrix pattern and tracks the reflected dots in the view of the RGB camera. In this configuration, it is assumed that the background is at infinity and any reflected dots are tracked as targets (Figure 5.2). The system is calibrated to measure the dot spacing at varying reflection depths. When a target is detected, the spacing of the dots reflected is mapped to judge an estimated range.

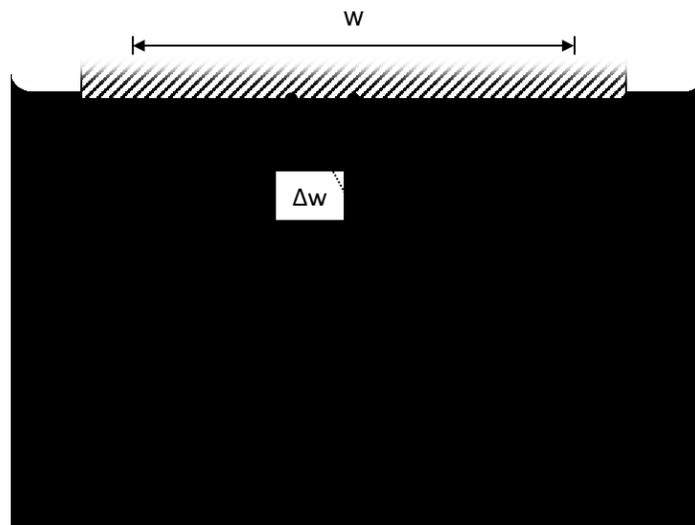


Figure 5.2: LAFTR operation for free target detection.

For free mode operation, the estimated range (r) to the target is calculated using Equation 5.1 below:

$$r = \frac{d}{\tan(\delta + \Delta\theta) - 2 \tan(\phi) \left(\frac{\Delta w_{px}}{w_{px}} \right)} \quad (5.1)$$

where:

d is the horizontal distance from the camera center to the laser projector center,

δ is the horizontal inclination of the laser projection from center calibration,

$\Delta\theta$ is the measured angular distance between marker grid points,

ϕ is the horizontal camera field-of-view,

Δw_{px} is the measured horizontal pixel width between detected marker points, and

w_{px} is the captured image width in pixels.

Close-Proximity Target Detection and Ranging

For operation in an enclosed environment, LAFTR can utilize information from dots reflected on the scene's background as illustrated in Figure 5.3. In this mode, the sensor detects moving targets and measures their shadow width to determine the target's distance. For the experiment performed in this section, a single projector is used to measure a horizontal shadow using this mode. Further development can explore the benefit of adding multiple projectors at horizontal and vertical arrangements for additional depth perception.

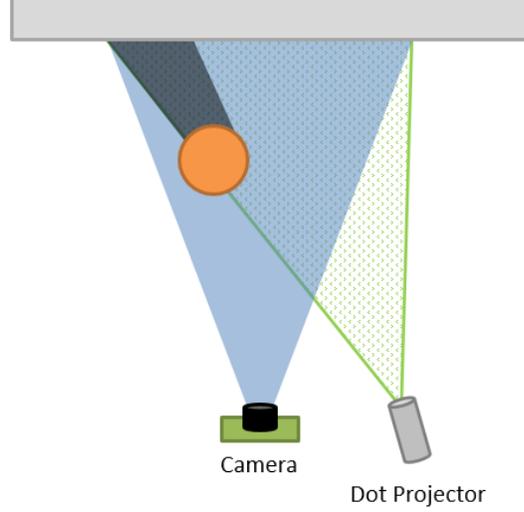


Figure 5.3: LAFTR operation for indoor detection and ranging.

For operation against a calibrated background scene, the distance from the camera to an occlusion can be estimated if the sensor geometry and position are known relative to a calibration plane. The distance (r) to the target can be found as:

$$r = \frac{d \sin(\delta - \Delta\theta)}{\cos(\delta - \Delta\theta) + \left(\frac{\sin(\delta - \Delta\theta)}{\tan \delta}\right)} \quad (5.2)$$

The expression for the angle shift ($\Delta\theta$) is:

$$\Delta\theta = \text{atan} \left(\frac{2\Delta w_{\text{px}} \tan(\phi/2)}{w_{\text{px}}} \right) \quad (5.3)$$

where:

Δw_{px} is the horizontal shift of the occluded point in pixels, and

ϕ is the camera's horizontal field-of-view.

The primary goal of this experiment is to explore the feasibility of the close-proximity indoor target detection. This is achieved using first a dot grid pattern for generic target tracking and then a vertical stripe pattern for thin horizontal target tracking.

5.3 Methods

5.3.1 Dot Grid Image Processing Algorithm

Image Capture

A single-laser LAFTR consists of a Raspberry Pi 3 computer with a NOIR Raspberry Pi V2 camera. The camera does not possess an integrated IR filter to enable future experimentation with IR band lasers. For dot projection, a 5mW 532 nm handheld laser pointer is used with two integrated diffraction films. A custom Python program controls the camera and projector as well as the image processing algorithm [66]. The Pi3 captures successive frames from a video stream while triggering the laser dot projector during each exposure. Once captured, images are processed using two operations that (1) locate and track the laser dots and (2) track moving targets. These two processes are, in the end, synthesized to give a range estimate. An overview of the algorithm can be seen in Figure 5.4.

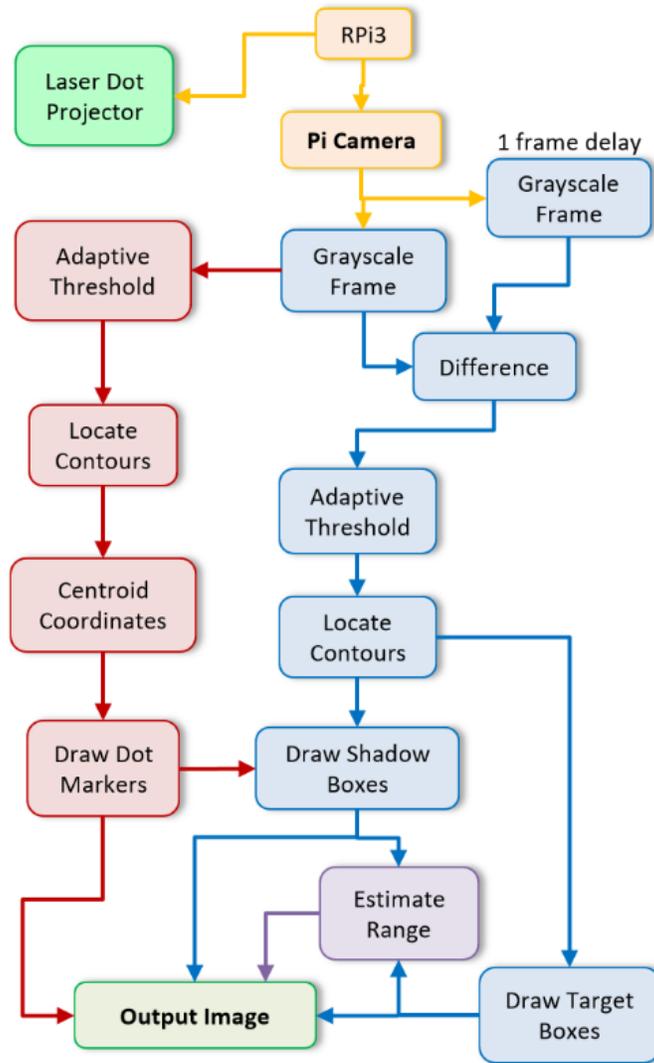


Figure 5.4: LAFTR dot grid image processing algorithm flowchart

Laser Dot Tracking

Each frame is converted to a binary image using an adaptive averaging threshold. The image is dilated, and each blob's centroid is calculated [67]. The dot locations are tracked (Figure 5.5) such that, when occluded, they indicate target shadows.

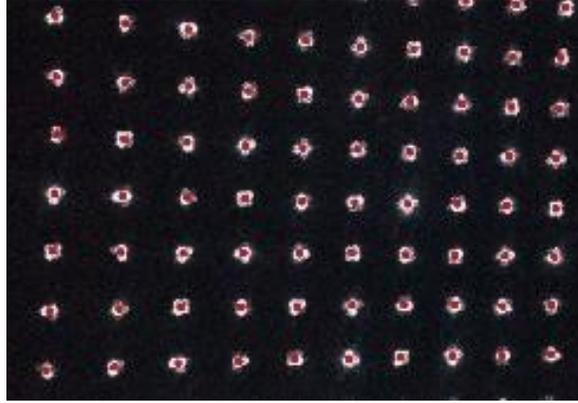


Figure 5.5: Section of captured image with red points marking tracked dot centroids

Motion tracking is performed by measuring the difference of two successive frames. The differential frame is converted to a binary image and adaptively thresholded. The detected blobs are dilated and then tracked and highlighted with bounding boxes (Figure 5.6) to indicate portions of detected targets [68].



Figure 5.6: Sample of tracked target with occlusion region (shadow)

Target Range Estimation

If a bounding box contains a tracked dot, it is considered a foreground object, but if occluded it is considered a shadow. Once categorized, the horizontal locations of these regions

are averaged. The difference between these regions is then used to compute the estimated target range using Equation 5.2.

5.3.2 Vertical Stripe Image Processing Algorithm

Image Capture

A similar configuration to the Dot Grid experiment is utilized for this second iteration of the LAFTR project. Again the Raspberry Pi 3 computer is used with the NOIR Raspberry Pi V2 camera offset from the 5mW green laser. However, in this configuration, the laser pattern was manifested as a series of vertical stripes. Image data are captured in a similar manner to the Dot Grid experiment but using a separate Python code. This code is used to (1) identify a thin horizontal occlusion and mark its position and then (2) provide a range estimate. The overall algorithm is illustrated in Figure 5.7 on the following page.

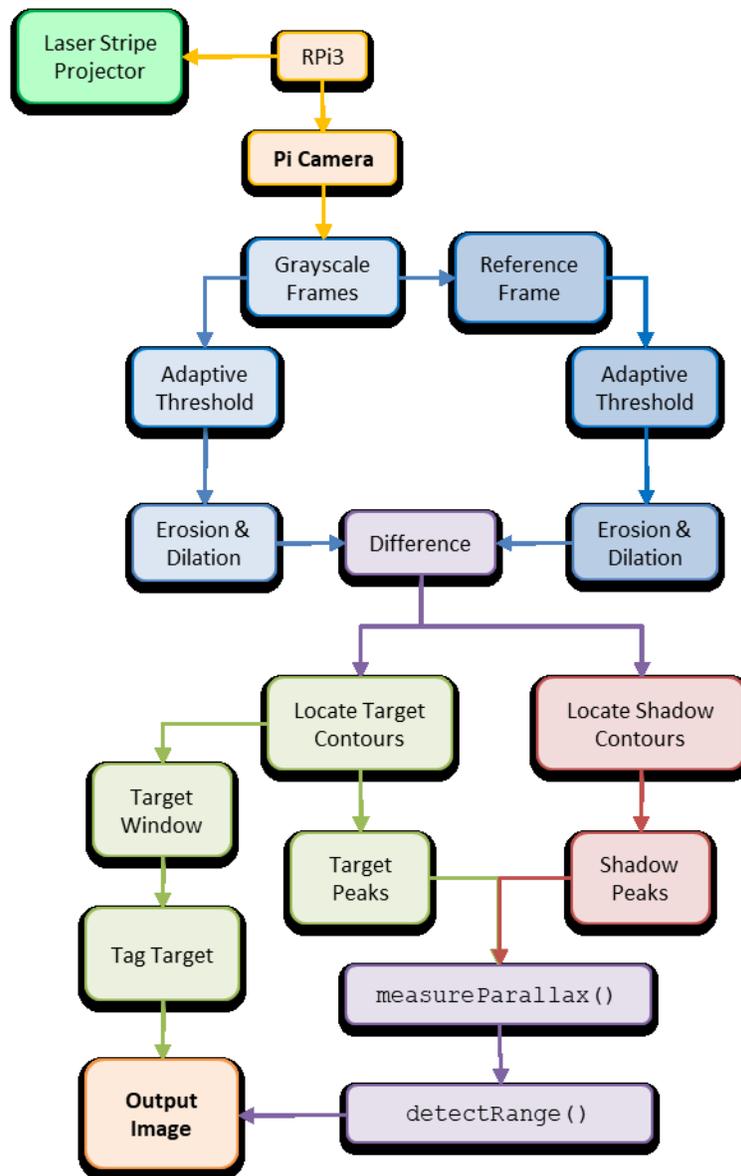


Figure 5.7: LAFTR vertical stripe image processing algorithm flowchart

Laser Pattern Tracking

For this configuration, an initial image is captured of the background environment to establish a reference of the reflected vertical stripes with no occlusions. When a thin horizontal occlusion is introduced in front of the sensor, the result is a series of short vertical line segments or essentially points (see Figure 5.8). Consecutive images are captured, converted to grayscale,

and conditioned with a slight Gaussian blur for noise reduction. An adaptive threshold is then applied to isolate the dots generated on the target object. To further reduce noise, an asymmetric erosion-dilation operation is performed. Once conditioned, the dots are located in a similar method to the Dot Grid algorithm, using blob detection of a defined pixel area.

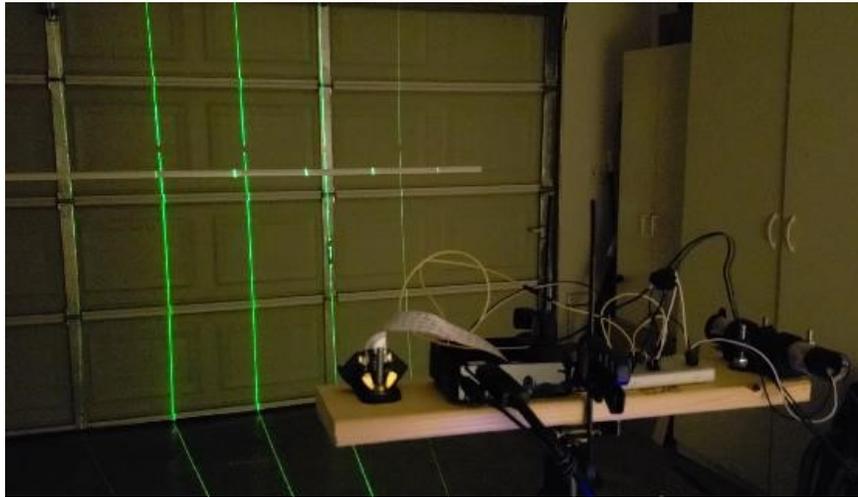


Figure 5.8: LAFTR in operation with thin horizontal occlusion placed at 200 cm

Range Estimation from Parallax

Upon locating the series of horizontal target dots, the algorithm generates a window focusing on the target region. From there, consecutive images are compared to the background image to detect positions where portions of the pattern are both displaced or occluded by the thin object. The peaks of these features (Figure 5.10) are then categorized as target or shadow dots. Within the window, the parallax is estimated by measuring the average horizontal gap width between target (reflected) and shadow (occluded) laser points. This gap width is then fed into the function describing Equation 5.2 to provide a range estimate.

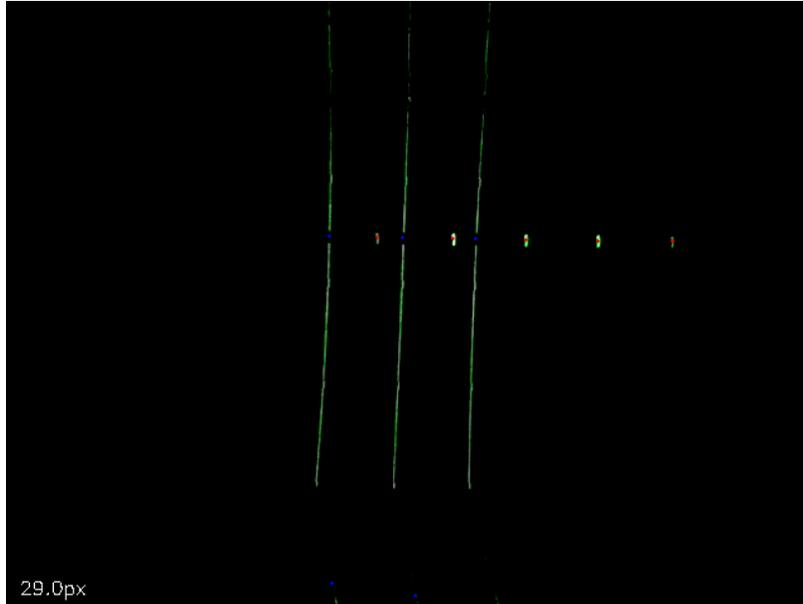


Figure 5.9: Occlusion target point detection and categorization

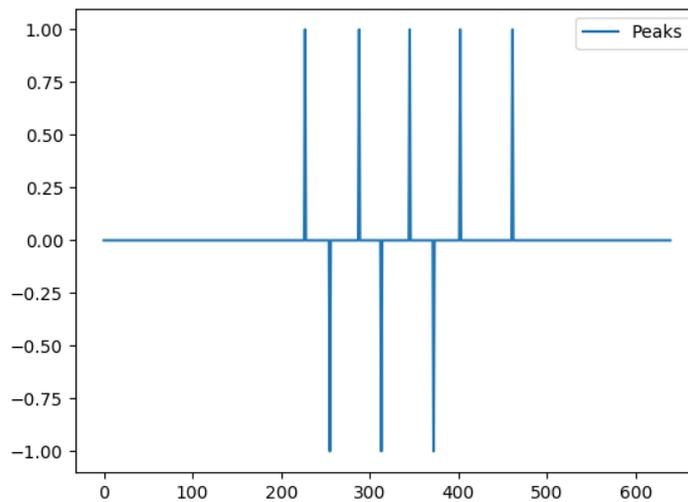


Figure 5.10: Target (positive) and shadow (negative) peak detection

5.4 Experimentation

5.4.1 Dot Grid LAFTR Experiment

Setup

For the preliminary LAFTR experiment, a single-laser LAFTR system was constructed for gathering preliminary data using a dot grid pattern. The setup consists of a Raspberry Pi 3

computer with a NOIR Raspberry Pi V2 camera. For dot projection, a 5mW 532nm handheld laser pointer is used with two integrated diffraction films. The camera and laser are held approximately 12 in apart and focused upon a smooth wall. Preliminary testing confirmed that the NOIR camera picked up residual IR band light from the green laser. It was found that because the laser's IR component was diffracted at a wider angle than the green band, the grid pattern was disrupted. The laser's spectrum was analyzed using an Ocean Optics Red Tide USB650 fiber optic spectrometer, demonstrating peak intensities at 532 nm and 810 nm as shown in Figure 5.11. To isolate the green band, a Melles Griot 550 nm band pass filter was integrated on the Raspberry Pi V2 camera. In the future, an IR band laser can be used to eliminate the need for a band pass filter.

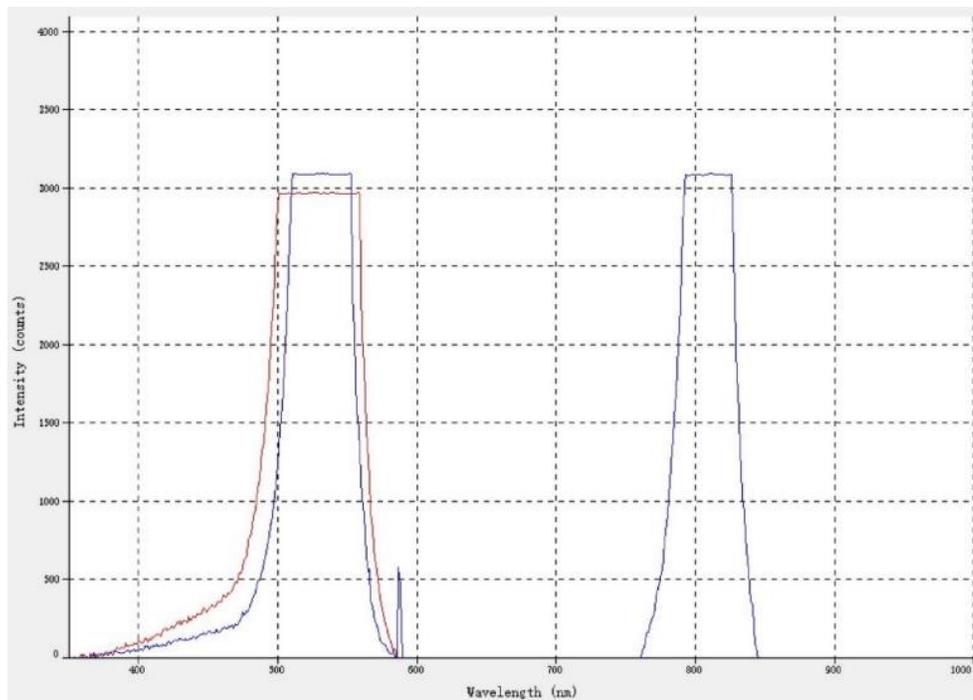


Figure 5.11: Unfiltered (blue) and bandpass filtered (red) spectra of 532 nm laser

The laser diffraction films enable the generation of three sizes of equally-spaced grid patterns based on the outer grating's rotation. This results in the narrow, medium, and wide grid spacings shown in Figure 5.12.

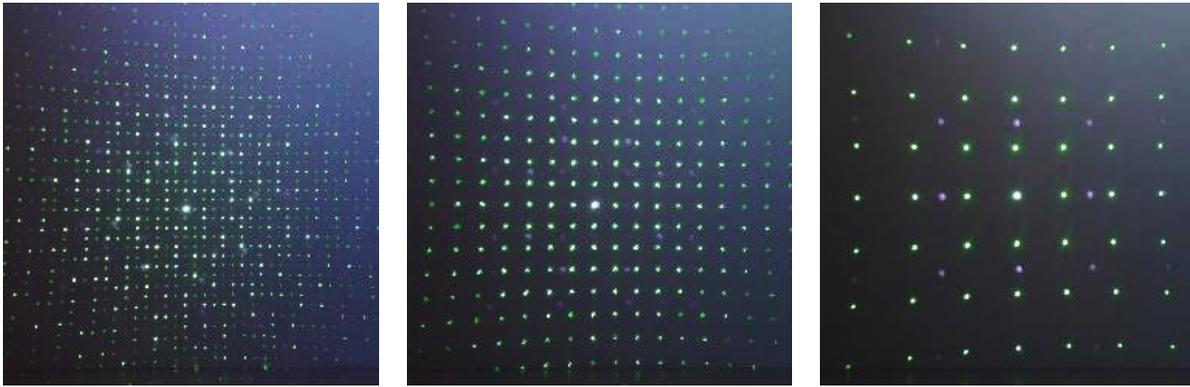


Figure 5.12: Narrow (left), medium (middle), and wide (right) laser dot spacing

The angular spacing of each grid size was experimentally measured and can be seen in Table 5.1.

Table 5.1: Laser dot projector characterization

Diffraction Size	Azimuthal Width	Angular Period
Narrow	60.5°	1.5°
Medium	61.8°	2.8°
Wide	84.4°	7.0°

Generic Target Tracking

An experiment was conducted with this LAFTR system placed perpendicular to a smooth concrete wall at a distance of 12 ft and a height of 4.5 ft. The system was then run such that the user could display the target tracking in real-time. The ambient light environment was dim (8 - 10 lux) and the sample rate was approximately 4 Hz. Test images were collected with a lateral moving human target at varying distance from camera as well as radial movement. Samples were also taken of a pair of lateral moving targets to test multiple target tracking. The system was then tested in complete darkness, using only the laser projector and no ambient lighting, while tracking a lateral moving human target.

Dot Grid Range Estimation

Following the results of the initial target tracking experiment, adaptive thresholding was implemented, and a range-detection algorithm was developed. The system was then tested against a smooth wall at a distance of 4.5 ft and a height of 4.5 ft. This test was performed in moderate ambient lighting (40 – 45 lux). Samples were taken of a smooth paper book moving radially from the wall toward the camera.

5.4.2 Vertical Stripe LAFTR Experiment

Setup

A similar setup to the Dot Grid system was constructed (Figure 5.13) to demonstrate the feasibility of detecting thin features with LAFTR. The same setup previously described was built but, due to hardware availability, the Raspberry Pi camera was covered with an IR filter taken from an old digital camera to eliminate extraneous wavelengths. The dot grid laser optics were also replaced with a Fresnel lens from a laser level which was then divided using a sheet of Flinn C-Spectra 200 line/mm diffraction film. The result produced a series of vertical stripes within the camera's field of view with an angular period of 6.0° . The LAFTR sensor was fixed 4.5 ft above the ground and pointed perpendicular to a metal garage door at a distance of 12 ft. The laser was controlled from one of the Raspberry Pi's GPIO pins through a reed relay. A push button was added to enable the user to manually collect images during operation.

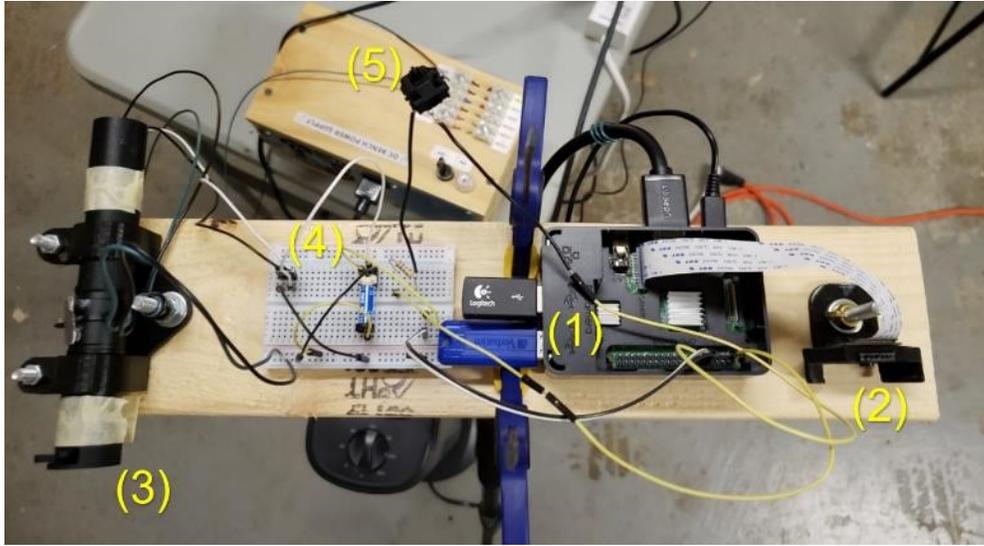


Figure 5.13: LAFTR test setup with (1) Raspberry Pi 3 computer, (2) Raspberry Pi Camera V2 with IR filter, (3) 5mW laser with vertical line optics, (4) relay circuit, and (5) manual image capture push button

Horizontal Target Detection

This system was tested using a single horizontal target at the same elevation of the sensor's camera. The target was a $\frac{3}{4}$ " strip of aluminum coated in white paint. Data were collected with the target at incremental distances beginning at 12 ft and proceeding down to 3 ft from the sensor. This experiment was conducted in darkness (0 lux) with no additional ambient light besides the laser. Upon detecting the target, a preview window displayed the sensor view, overlaying a marker over the horizontal target and a range estimation when available.

5.5 Results

Dot Grid LAFTR Results

Images during the Dot Grid experiments were recorded with a manual trigger and some samples can be seen in Figures 5.14 and 5.15 below. Red markers indicate tracked dot centers and blue boxes indicate portions of detected targets. The system successfully detected and tracked the moving subject in dim ambient light moving at varying distance from the camera.

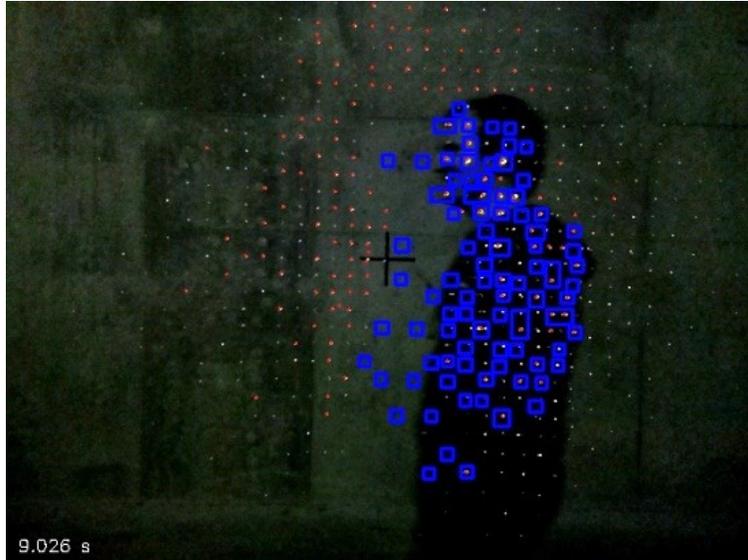


Figure 5.14: Lateral moving human target at 5 ft



Figure 5.15: Multiple lateral moving human targets at 6 ft and 10 ft

The sensor also successfully tracked a moving target in the absence of ambient light, relying only on the laser dot projector (Figure 5.16). At the time of data collection in the dark environment, numerous false positive targets were detected due to thresholding inconsistencies. This issue was addressed by implementing adaptive thresholding for future tests.

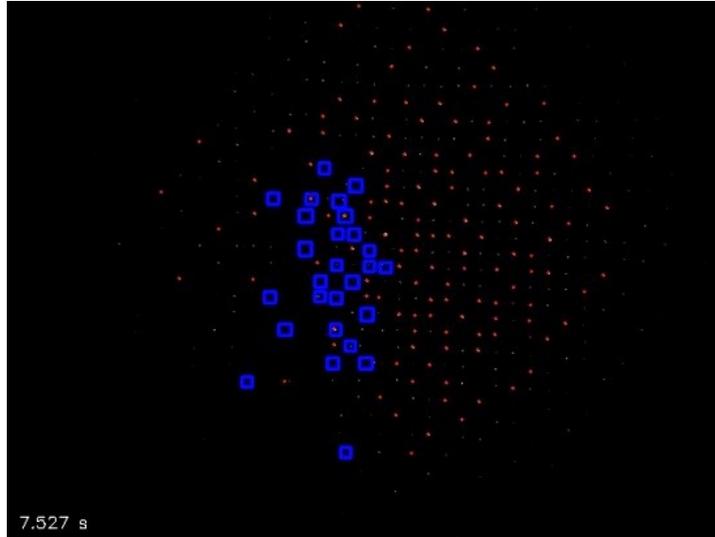


Figure 5.16: Lateral moving target at 6 ft with no ambient light

After integrating the range estimation algorithm, a second data set was collected with a hand-held target moving radially with respect to the camera (Figure 5.17). The images below are sampled from a sequence of data taken in the lab. Green boxes indicate detected motion regions categorized as occlusions. The resulting range estimates were consistently accurate within roughly 10 cm (4 in). This accuracy can be improved with calibration or additional adjustment to the laser pattern and processing algorithm. It was observed that closer targets took significantly longer processing time due to the increased quantity of tracked regions. This issue can be optimized by grouping or filtering tracked features. While the range estimation for this configuration is achieved by averaging blob position, it can be assumed that such a method would encounter issues resolving multiple objects if they were to be introduced. However the algorithm's proof-of-concept demonstrated that the function created for this sensor can provide basic range estimation. The results from this experiment fed into the development of the horizontal feature detector.

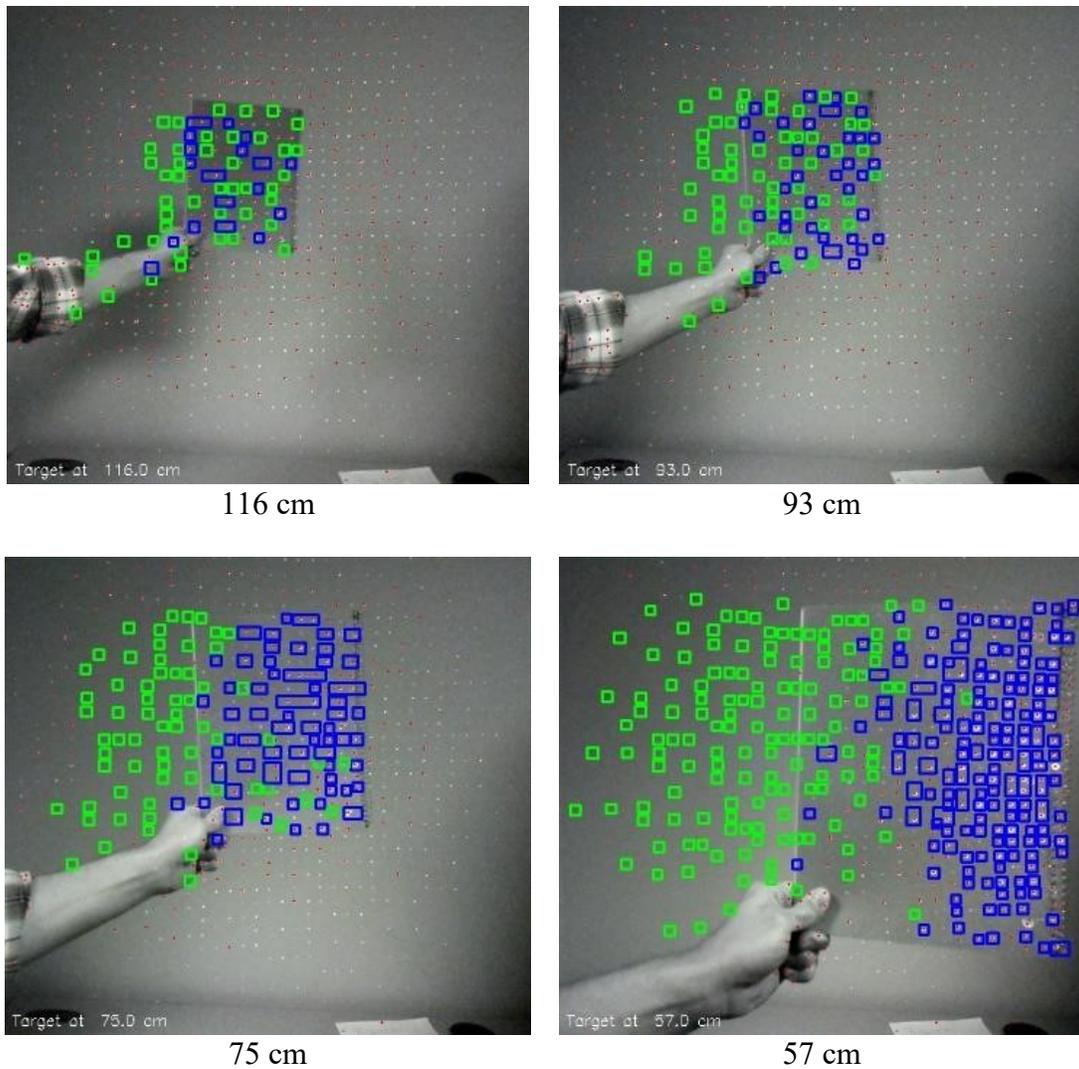


Figure 5.17: Sequence of target tracking and ranging with estimated ranges listed (green indicates occluded points [shadows] and blue indicates target surface points)

Vertical Stripe LAFTR Results

When introducing the horizontal target during the stripe pattern experiment, the LAFTR sensor was able to successfully identify and mark the occlusion's position in real time (Figure 5.18). While the sample rate was low—approximately 1.2 Hz—the sensor demonstrated an ability to detect the target continuously.

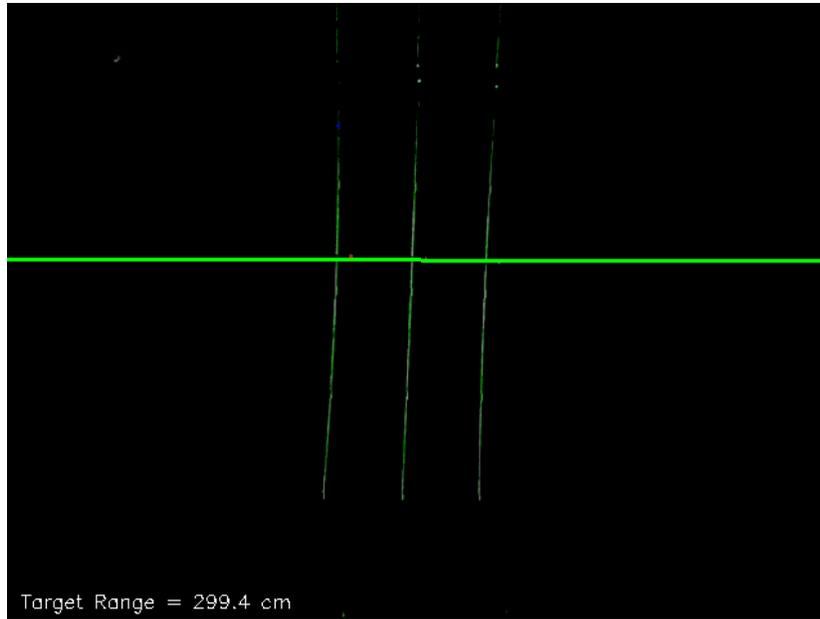


Figure 5.18: Sample image of LAFTR viewer with target overlay and range estimation

The range estimation algorithm was somewhat sporadic due to errors resolving the occluded dot patterns. Nevertheless, the range estimations captured between 2.0 and 3.0 m were accurate within approximately 20 cm (8 in) as seen in Figure 5.19. It was found that the range estimation inverted between 1.7 m and 2.0 m. The parallax detection algorithm does not account for the direction of offset from the reference image as it simply searches for the lowest average gap width between shadow and target peaks. If these could be resolved, it is believed that the range estimation between 1.7 m and 2.0 m would be consistent along the trend shown between 2.0 and 3.0 m. It can also be seen that the range estimations between 1.0 and 1.7 m display a discontinuity but still follows a similar trend from the 2.0 – 3.0 m zone. This is due to the ambiguity associated with the detection method and the periodic nature of the laser pattern. At 1.7 m, range estimation failed because the displaced target dots were in phase with the background and were thus indistinguishable from the default background. As such the range estimation resets within 1.7 m and does not account for the additional offset due to periodic ambiguity. This is the fundamental

limitation of this sensor and may need to be controlled from a secondary ranging method to remove error propagation.

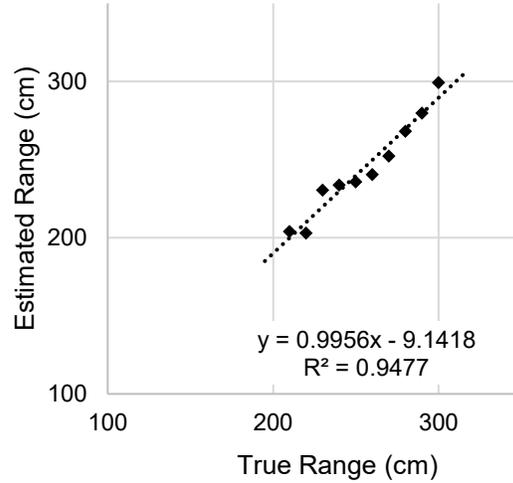
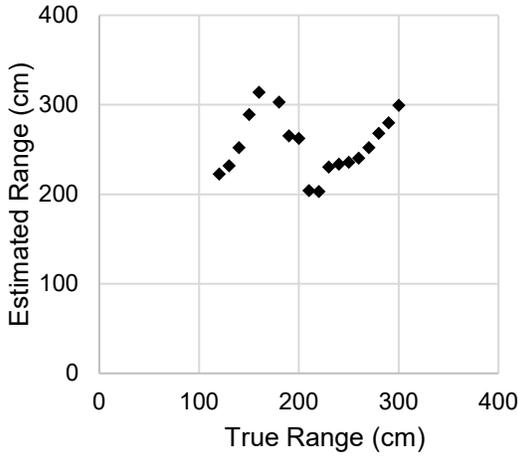


Figure 5.19: Estimated range relationship

Figure 5.20: Linear region of estimated range

Despite its limitations, this LAFTR configuration successfully demonstrates the ability to identify and track a thin, horizontal feature with no ambient light. As seen in Figure 5.20 below, the range estimation proves reasonable with for ranges between 2.0 and 3.0 m. This can be further improved by establishing and implementing a calibration procedure to further improve accuracy and remove the 9 cm bias seen in the charts below.

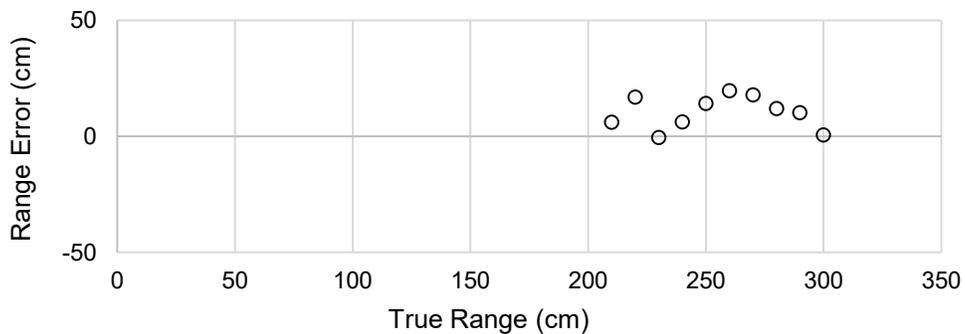


Figure 5.21: Associated error of range estimation

5.6 Conclusion

The experiments performed with this prototype system indicates that LAFTR is a viable method for tracking motion and possibly estimating the range of moving targets in a dark indoor environment. The initial dot grid system consistently detected a variety of moving targets in ambient light and in darkness up to 12 ft. Initial results indicate that low-fidelity range estimation can be done by measuring the width of occluded laser features. This capability was further explored with the vertical stripe system that proved to be capable of identifying and ranging thin horizontal obstacles within an accuracy of 20 cm. The performance of the vertical stripe system was hindered due to its inability to distinguish parallax direction and was subject to ambiguity due to the periodicity of the projection pattern. To ensure better data is collected, a combination of laser patterns such as alternating vertical stripes, horizontal stripes, and dots could reduce uncertainty. The periodic ambiguity is a critical issue with the stripe sensor and may only be prevented by implementing differential range propagation from tracked targets or by introducing feedback from a secondary sensor for error reduction. LAFTR can further be improved by experimenting with a variety of laser projector patterns to determine optimal feature tracking. Further developing the image processing algorithm can enable more accurate range measurement, distinct target labeling, and speed. Once refined and expanded to have multiple laser projectors, LAFTR can be used to determine rough azimuth, altitude, and range detection for use as an obstacle avoidance suite onboard a small UAS.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

This thesis introduces a collection of alternative technologies to support the development of a hypothetical next-generation sUAS for collecting comprehensive environmental data in little to no light. Such technology was desired in response to the restricted performance discovered during ARPL's Hurricane Irma aerial surveying deployment. Combining this motivation with the growing need to develop better and cheaper remote-sensing and navigation methods, a series of experiments were performed to explore low-cost solutions to the existing limitations. The experiments successfully demonstrate software processes and novel hardware to affordably augment commercial UAS's ability to collect data and navigate in complex flight environments.

The aerial mapping workflow from Chapter 3 compared open-sourced GIS software to commercial software and showed suitable performance from WebODM to generate high resolution maps and DEM's with little to no user cost. Using the GPS emulator and IR image conversion processes developed for this research, the aerial mapping workflow was expanded to support the use of DJI drones and FLIR thermal cameras and provide a comprehensive means of rapidly collecting and processing surveying data for individuals and businesses.

To support the development of advanced sensing and navigation, the novel ultrasonic and laser sensors from Chapters 4 and 5 represent a potential solution to existing UAS navigational limitations. The ultrasonic thermometry experiment successfully demonstrated the feasibility of

measuring air fluid temperature using an acoustic sensor for numerous remote sensing applications. This experiment also showed the possibility of applying radar signal processing algorithms to further develop ultrasonic sensing toward more precise ranging for data acquisition or obstacle avoidance. The LAFTR system also demonstrated a novel sensing method to enable low-fidelity target detection for thin features that typically require advanced and expensive hardware to detect.

Together, the experiments in this project establish the groundwork for a small UAS capable of navigating through a complex environment without the need for expensive sensors or within a GPS denied area. Using basic ultrasonic transducers and laser projectors could enable the UAS to not only navigate a dark environment and avoid hard-to-detect obstacles but also collect comprehensive thermal data along its way. By then employing the aerial surveying workflow developed in this research, the hypothetical UAS could be used to rapidly collect and process thermal and visible light data autonomously for numerous applications as UAS begin to provide solutions to everyday problems.

6.2 Future Work

While the concepts explained in these chapters show successful results, each concept requires significant development to become a fieldable solution. Additional steps can certainly be taken to further progress these concepts into actual tools for integration on existing or theoretical UAS platforms.

Aerial Mapping

While it was concluded that WebODM showed the best 3D reconstruction quality among software tested in this research, the process could possibly be further optimized. The method used to run WebODM in these experiments were performed in a Linux virtual machine running

within a Microsoft Windows operating system. The Windows machine used possessed only 6 CPU cores and 32 GB of RAM which were used to capacity. During reconstruction, it is presumed that the program's memory spilled over onto swap, limiting the operation to the speed of the hard disc. If the reconstruction process could be distributed across CPU's on a more powerful Linux computer or server with more memory, the overall reconstruction time could likely be reduced.

The DEM's produced from the aerial mapping technique used in this research produced polygon files in common formats such as .OBJ or .PLY 3D meshes. It is possible that a process could be established to convert the 3D mesh data into a topographical map for importing into GCS software for in situ mission planning. Using an open-source tool such as Blender [69], a script could be developed to automatically slice the DEM mesh into layers of desired elevation and produce a georeferenced topographical image. The image could then be uploaded to a web map service (WMS) and then used to build autonomous flight missions in GCS software such as Mission Planner. Such a workflow could be used for several realtime surveying applications—especially for tactical military operations. For regions with inadequate satellite map information, a scout UAS could autonomously map the region and build a topographical map in a matter of hours to then enable more precise mission planning with additional UAS in the fleet.

To further enhance the precision of the GPS emulator program for DJI mapping, it is desired to use more image data than the 1 Hz sample rate restricted by the SRT log rate within the aircraft. To work around this, an algorithm could be developed to interpolate between the available GPS points from the logs and fill the missing gap with artificial points to estimate the path in between log points. This would enable additional images to be “georeferenced” to enable better photogrammetric data collection when operating at higher flight speeds.

Ultrasonic Thermometry

While the ultrasonic thermometer prototype successfully demonstrated thermal measurement in a controlled environment, it is desirable to test the sensor with more varying conditions. The experiment from Section 4.4.2 could be repeated with the sensor placed at varying range to observe any potential inconsistencies or loss when detecting fluid temperature. Additionally, a variety of building materials could be tested to compare the acoustic response of not just drywall but also brick, concrete, wood, or other materials.

To further explore the feasibility of ultrasonic thermometry, a phased array could be constructed to boost the overall gain of the acoustic signal and possibly enable beam steering. If adequate data can be collected from such a system, it may be possible to use specialized SAR algorithms to generate low-fidelity thermal or spatial maps of the target surface using ultrasound. This could be optimized by testing and selecting the best acoustic waveform for detecting the obstacles in the environment.

Laser-Augmented Feature Tracking

The prototype built for LAFTR successfully demonstrated the basic concept of estimating range by measuring parallax from reflected laser features. However, the quality of this measurement could be greatly improved by using a higher power laser and better passive optics to create more laser stripes within the camera field of view. Additionally, if an IR laser were implemented, the camera could filter out visible light for operation at night or in proximity to humans. An advanced system could incorporate multiple laser projectors to produce vertical stripes, horizontal stripes, and dots to detect and categorize a variety of targets. This should also be tested against a variety of obstacle orientations, shape, and reflectivity to quantify the sensor's performance.

To fully reach its potential, the LAFTR system must have an algorithm capable of separating detected targets and dynamically adjusting for movement of the sensor itself. The preliminary experiment in this research shows basic range estimation of a single target but is not capable of distinguishing multiple targets. The experiment in Section 5.4.2 also is limited to a stationary sensor that relies on a static background. If the sensor were mounted on a mobile platform (ie. UAS), the reference image for parallax calculation must be continuously refreshed.

The Python code within LAFTR can certainly be optimized further. The image processing time could be reduced with better image data from quality camera and laser optics and faster computer vision algorithms. The range calculation in the existing code is performed using actual trigonometry functions, whereas a fielded sensor could utilize a lookup table stored in memory for more rapid processing.

REFERENCES

- [1] D. News, "MEMS Price to Fall As Production Rises," *Automotive, Automation & Motion Control*, 22 September 2008.
- [2] H. Barragán, "The Untold History of Arduino," 2016. [Online]. Available: <https://arduinohistory.github.io/>. [Accessed March 2019].
- [3] J. Vasquez, "Multicopters and Their MultiWii Beginnings," 19 April 2015. [Online]. Available: <https://hackaday.com/2015/04/19/multicopters-and-their-multiwii-beginnings/>.
- [4] ArduPilot, "ArduPilot Open Source Autopilot," 2016. [Online]. Available: <http://ardupilot.org/>. [Accessed March 2019].
- [5] 'Pixhawk: Home Page,' [Online]. Available: <http://pixhawk.org/>. [Accessed March 2019].
- [6] F. A. Administration, "14 CFR Part 107 Small Unmanned Aircraft Systems," 1 January 2017. [Online]. Available: <https://www.govinfo.gov/content/pkg/CFR-2017-title14-vol2/pdf/CFR-2017-title14-vol2-part107.pdf>. [Accessed June 2019].
- [7] F. A. Administration, "FAA Modernization and Reform Act of 2012 - Section 333," 1 February 2012. [Online]. Available: https://www.faa.gov/uas/advanced_operations/section_333/media/Sec_331_336_UAS.pdf. [Accessed June 2019].
- [8] DroneDeploy, "2018 Commercial Drone Industry Trends Report," May 2018. [Online]. Available: https://dronedeploy-www.cdn.prismic.io/dronedeploy-www%2Fae535fda-dfc9-4bcf-9743-292df714e9fe_dd__2018_trends_report-f.pdf. [Accessed June 2019].
- [9] The University of Alabama News Center, "University of Alabama Team Assesses Hurricane Irma Aftermath," 15 November 2017. [Online]. Available: <https://www.ua.edu/news/2017/11/university-of-alabama-team-assesses-hurricane-irma-aftermath/>. [Accessed March 2019].
- [10] I. I. f. B. & H. Safety, "FORTIFIED Home - The National Standards for Resilient Construction," 2019. [Online]. Available: <https://fortified2018.wpengine.com/fortified/>. [Accessed 2019].

- [11] Tallahassee Weather Forecast Office, "Detailed Meteorological Summary of Hurricane Irma," 14 August 2018. [Online]. Available: https://www.weather.gov/tae/Irma_technical_summary. [Accessed March 2019].
- [12] FLIR, "FLIR Vue Pro R Radiometric Drone Thermal Camera," 2019. [Online]. Available: <https://www.flir.com/products/vue-pro-r/>. [Accessed May 2019].
- [13] F. Incropera, D. Dewitt, T. Bergman and A. Lavine, Fundamentals of Heat and Mass Transfer, Hoboken: John Wiley & Sons, 2007.
- [14] M. Hord, Digital Image Processing of Remotely Sensed Data, New York: Academic Press, Inc, 1982.
- [15] R. MacDonald and F. Hall, "Global Crop Forecasting," *Science*, vol. 208, no. 4445, pp. 670-679, 1980.
- [16] B. Zang, "Systems and Methods for Target Tracking". United States Patent 9,164,506 B1, 20 October 2015.
- [17] D. Eynard, P. Vasseur, C. Demonceaux and V. Frémont, "UAV altitude estimation by mixed stereoscopic vision," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Teipei, 2010.
- [18] DJI, "Guidance," 2019. [Online]. Available: <https://www.dji.com/guidance/features>. [Accessed May 2019].
- [19] R. Kingslake, Lens Design Fundamentals, New York: Academic Press, Inc., 1978.
- [20] A. Fox, "The Latest Drones Tests for Public Safety," 18 October 2017. [Online]. Available: <https://efficientgov.com/blog/2017/10/18/7-public-safety-drone-tests/>. [Accessed December 2019].
- [21] S. Gallardo-Saavedra, L. Hernández-Callejo and O. Duque-Perez, "Technological review of the instrumentation used in aerial thermographic," *Renewable and Sustainable Energy Reviews*, vol. 93, pp. 566-579, 2018.
- [22] N. Negied, E. Hemayed and M. Fayek, "Pedestrians' detection in thermal bands - Critical Survey," *Journal of Electrical Systems and Information Technology*, vol. 2, no. 2, pp. 141-148, 2015.
- [23] G. Ezenne, L. Jupp, S. Mantel and J. Tanner, "Current and potential capabilities of UAS for crop water productivity in," *Agricultural Water Management*, vol. 218, pp. 158-164, 2019.

- [24] U. o. A. A. R. P. Laboratory, "Merrill Hall 3D Model," October 2018. [Online]. Available: <https://sketchfab.com/3d-models/merrill-hall-7daae01da9494f74b91cb0703533cc67>. [Accessed May 2019].
- [25] T. U. D. Team, "Somerville Hall 3D Scan," 13 February 2017. [Online]. Available: <https://www.youtube.com/watch?v=S7-wa2JiwhU>. [Accessed May 2019].
- [26] S. Hemmelder, W. Marra, H. Markies and S. Jong, "Monitoring river morphology & bank erosion using UAV imagery – A case study of the river Buëch, Hautes-Alpes, France," *International Journal of Applied Earth Observation and Geoinformation*, vol. 73, pp. 428-437, 2018.
- [27] M. Richards, *Fundamentals of Radar Signal Processing*, New York: McGraw-Hill, 2005.
- [28] A. Scannapieco, A. Renga and A. Moccia, "Compact Millimeter Wave FMCW InSAR for UAS," in *IEEE Metrology for Aerospace*, Benevento, 2015.
- [29] M. e. a. Schartel, "A Multicopter-Based Focusing Method for Ground Penetrating Synthetic Aperture Radars," in *IEEE International Geoscience and Remote Sensing Symposium*, Valencia, 2018.
- [30] S. Yan and S. Gogeneni, "Airborne UWB Radar for Alpine Snow Measurements," 2018. [Online]. Available: https://rsc.ua.edu/wp-content/uploads/2019/02/NWC_posterPDF.pdf. [Accessed October 2019].
- [31] C. Ippolito, K. Krishnakumar and S. Hening, "Preliminary results of powerline reconstruction from airborne LiDAR for safe autonomous low-altitude urban operations of small UAS," in *IEEE Sensors*, Orlando, 2016.
- [32] T. Rossing, *Springer Handbook of Acoustics*, New York: Springer, 2007.
- [33] D. Zhang, R. Watson, G. Dobie, C. MacLeod and G. Pierce, "Autonomous Ultrasonic Inspection Using Unmanned Aerial Vehicle," in *IEEE International Ultrasonics Symposium*, Kobe, 2018.
- [34] S. French, "DJI Market Share," 18 September 2018. [Online]. Available: <http://thedronegirl.com/2018/09/18/dji-market-share/>. [Accessed March 2019].
- [35] Skylogic Research, "2018 Drone Market Sector Report," 14 January 2019. [Online]. Available: <http://droneanalyst.com/research/research-studies/2018-drone-market-sector-report-purchase>. [Accessed March 2019].
- [36] DJI, "Mavic Pro," 2019. [Online]. Available: <https://www.dji.com/mavic>. [Accessed March 2019].

- [37] Litchi, "Litchi Mission Hub," [Online]. Available: <https://flylitchi.com/hub>. [Accessed March 2019].
- [38] QGroundControl, "QGroundControl," 2017. [Online]. Available: <http://qgroundcontrol.com/>. [Accessed March 2019].
- [39] Ardupilot, "Mission Planner Home," 2019. [Online]. Available: <http://ardupilot.org/planner/>. [Accessed March 2019].
- [40] OpenCV Team, "OpenCV Library," 2019. [Online]. Available: <https://opencv.org/>. [Accessed March 2019].
- [41] National Coordination Office for Space-Based Positioning, Navigation, and Timing, "GPS Accuracy," 5 December 2017. [Online]. Available: <https://www.gps.gov/systems/gps/performance/accuracy/>. [Accessed March 2019].
- [42] OpenDroneMap, "WebODM Drone Mapping Software," [Online]. Available: <https://www.opendronemap.org/webodm/>. [Accessed March 2019].
- [43] E. a. W. C. Zheng, "Structure from Motion Using Structure-Less Resection," in *IEEE International Conference on Computer Vision*, Santiago, Chile, 2015.
- [44] Y. a. P. J. Furukawa, "Accurate, Dense, and Robust Multiview Stereopsis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1362-1376, 2010.
- [45] M. a. H. H. Kazhdan, "Screened Poisson Surface Reconstruction," *ACM Transactions on Graphics*, vol. 32, no. 3, 2013.
- [46] Dronecode, "MAVLink Developer Guide," [Online]. Available: <https://mavlink.io/en/>. [Accessed July 2019].
- [47] FLIR, "Batch Operations on IR Images," [Online]. Available: https://flir.custhelp.com/app/answers/detail/a_id/1694/track/AvP9~wqCDv8S~a~3Gpoe~yJ7afMqYy75Mv8f~zj~PP~W. [Accessed March 2019].
- [48] P. Harvey, "ExifTool: Read, Write and Edit Meta Information," [Online]. Available: <https://www.sno.phy.queensu.ca/~phil/exiftool/>. [Accessed June 2019].
- [49] Z. O'Neill, *IRG 030-1718: Building Infiltration Estimator with Ultrasonic Thermometry (BLAST)*, Tuscaloosa, AL: American Society of Heating, Refrigeration, and Air-Conditioning Engineers, 2017.
- [50] Z. Hao and Z. O'Neill, "Model-Based Estimation of Building Infiltration," in *ASHRAE 2018 Building Performance Analysis Conference and SimBuild*, Chicago, IL, 2018.

- [51] U. D. o. Energy, "Blower Door Tests," [Online]. Available: <https://www.energy.gov/energysaver/blower-door-tests>. [Accessed December 2019].
- [52] N. R. E. Laboratory, "Field Test Best Practices: Tracer Gas Testing," [Online]. Available: https://buildingsfieldtest.nrel.gov/tracer_gas_testing. [Accessed December 2019].
- [53] J. Lienhard, *A Heat Transfer Textbook*, Cambridge, Massachusetts: J.H. Lienhard V, 2000.
- [54] G. Tobey and J. Graeme, *Operational Amplifiers: Design and Applications*, New York City, NY: McGraw-Hill, 1971.
- [55] C. Wosjlaw and E. Moustakas, *Operation Amplifiers: The Devices and Their Applications*, Hoboken, NJ: Wiley, 1986.
- [56] K. Huang, C. Huang and Y. Y. M. Li, "High precision, fast ultrasonic thermometer based on measurement of the speed of sound in air," *Review of Scientific Instruments*, vol. 73, p. 4022, 25 October 2002.
- [57] R. Jia, Q. Xiong, L. Wang, X. Shen, S. Liang and X. Shi, "Study of ultrasonic thermometry based on ultrasonic time-of-flight measurement," *AIP Advances*, vol. 6, no. 3, 2016.
- [58] ElecFreaks, "HC-SR04 Datasheet," [Online]. Available: <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>. [Accessed July 2019].
- [59] G. GmbH, "Matched Filter Simulator," [Online]. Available: <http://www.grauonline.de/alexwww/ardumower/filter/filter.html>. [Accessed July 2019].
- [60] S. Gurbuz, *ECE 693 Advanced Radar Signal Processing Lecture 3: Radar Cross Section and Clutter*, Tuscaloosa: The University of Alabama, 2018.
- [61] S. Gurbuz, *ECE 693 Advanced Radar Signal Processing Lecture 6: Waveform Design*, Tuscaloosa: The University of Alabama, 2018.
- [62] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, pp. 7-42, 2002.
- [63] G. Zhou, S. Yang and A. Liu, "Methods and systems for determining a state of an unmanned aerial vehicle". United States of America Patent 10,060,746 B2, 28 August 2018.
- [64] J. Posdamer and M. Altschuler, "Surface measurement by space-encoded projected beam systems," *Computer Graphics and Image Processing*, vol. 18, pp. 1-17, Mar 1982.

- [65] S. Ramasamy, A. Gardi and R. Sabatini, "A laser obstacle detection and avoidance system for manned and unmanned aircraft applications," in *International Conference on Unmanned Aircraft Systems*, Denver, CO, 2015.
- [66] J. e. a. Chen, "Fast 3-D shape measurement using blink-dot projection," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan, 2013.
- [67] B. Pesach and Z. Mor, "Projector of structured light". United States of America Patent 2018/0231373 A1, 16 August 2018.
- [68] B. Denkena, H. Ahlers, F. Berg and T. Wolf, "Fast inspection of larger sized curved surfaces by stripe projection," *CIRP Annals*, vol. 51, no. 1, pp. 499-502, 2002.
- [69] R. Gonzalez and R. Woods, *Digital Image Processing*, 2nd Edition, Upper Saddle River, NJ: Prentice Hall, 1992.
- [70] R. Szeliski, *Computer Vision: Algorithms and Applications*, London, England: Springer, 2011.
- [71] A. Rosebrock, "Basic motion detection and tracking with Python and OpenCV," 25 May 2015. [Online]. Available: <https://www.pyimagesearch.com/2015/05/25/basic-motion-detection-and-tracking-with-python-and-opencv/>. [Accessed December 2018].
- [72] Blender, "About Blender," [Online]. Available: <https://www.blender.org/>. [Accessed October 2019].
- [73] D. Halliday, R. Resnick and J. Walker, *Fundamentals of Physics*, Hoboken, NJ: John Wiley & Sons, Inc., 2010.
- [74] G. Pálmason, J. Friedman, R. Williams Jr., J. Jónsson and K. Saemundsson, "Aerial infrared surveys of Reykjanes and Torfajökull thermal areas, Oceland, with a section on cost of exploration surveys," *Geothermics*, vol. 2, pp. 399-412, 1970.
- [75] Adobe, "Creative Cloud Plans for Teams," 2019. [Online]. Available: <https://www.adobe.com/creativecloud/business-plans.html>. [Accessed December 2019].
- [76] Autodesk, "ReCap Pro Subscription," 2019. [Online]. Available: <https://www.autodesk.com/products/recap/subscribe?plc=RECAP&term=1-YEAR&support=ADVANCED&quantity=1>. [Accessed December 2019].

APPENDIX A
ADDITIONAL FIGURES

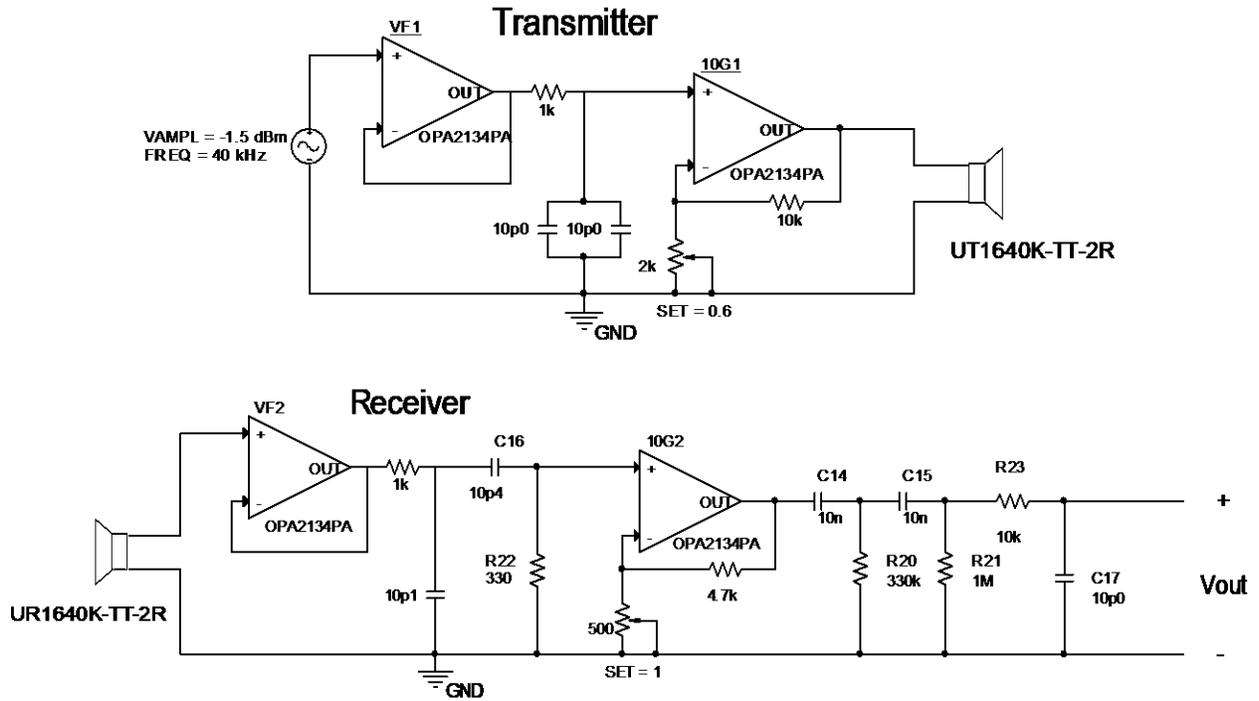


Figure A.1: Ultrasonic thermometer custom circuit diagrams

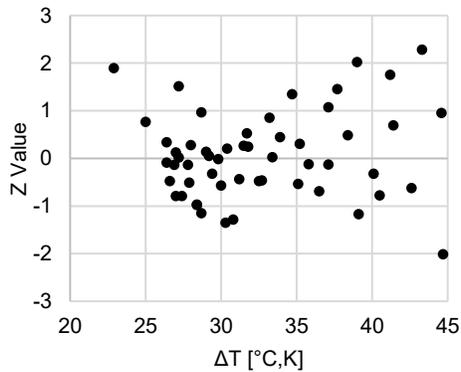


Figure A.2: Residuals for the linear modeling of phase difference correlation for ultrasonic thermometer

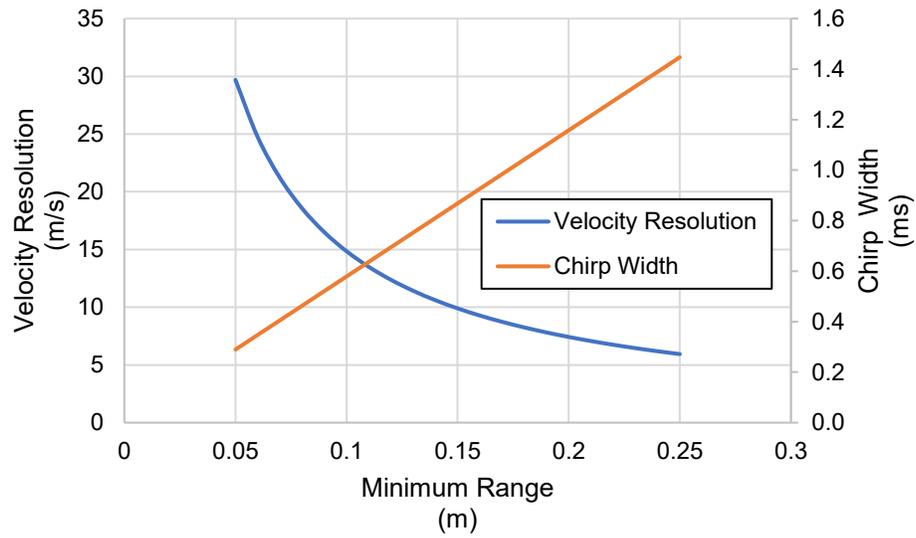


Figure A.3: Velocity resolution of ultrasonic LFM chirp as a function of desired minimum range



Figure A.4: LAFTR Laser stripe optics, featuring Fresnel lens and adjustable diffraction film

APPENDIX B

LAFTR: FREE TARGET DETECTION

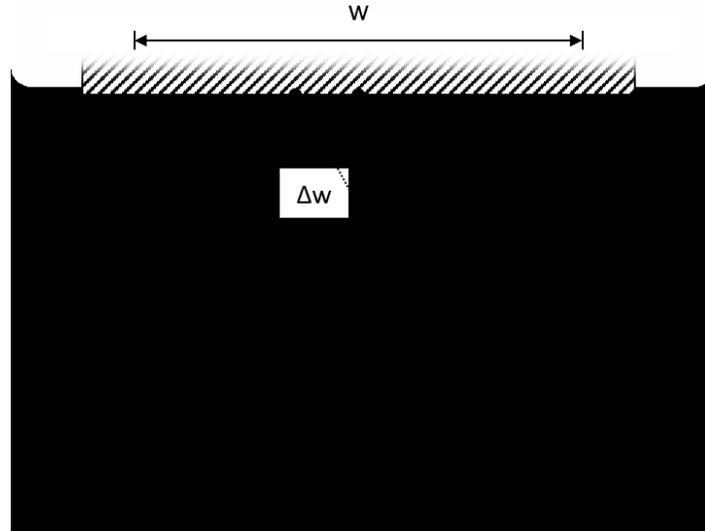


Figure B.1: Schematic of LAFTR free target detection method

Point C represents the center of a camera with a horizontal field of view, ϕ . The camera is positioned normal to a flat wall at an unknown distance, r . A laser projector light source is positioned parallel to the camera with an offset of d . Assuming the laser pattern consists of a periodic feature such as a line or dots, the angular period of the pattern is denoted as $\Delta\theta$. Assuming the laser pattern is centered with the camera, point A denotes the locational of the central laser feature. The angle associated with this alignment is denoted as the angle δ . The first adjacent feature then appears at a horizontal offset of Δw located at point B on the surface. Using geometric analysis, the relation below can be established:

$$\tan(\delta + \Delta\theta) = \frac{\delta + \Delta w}{r} \quad (\text{B.1})$$

The spatial measurements w and Δw have corresponding digital values perceived by the camera in pixels, denoted as w_{px} and Δw_{px} , respectively. Thus the relationship exists:

$$\Delta w = w \left(\frac{\Delta w_{px}}{w_{px}} \right) \quad (\text{B.2})$$

The spatial width of the camera's field of view is shown to be a function of the distance to the wall:

$$w = 2r \tan \phi \quad (\text{B.3})$$

Such that Equation A.2 can be expanded into

$$\Delta w = 2r \tan \phi \left(\frac{\Delta w_{px}}{w_{px}} \right) \quad (\text{B.4})$$

Thus Equation A.1 can be expanded to

$$\tan(\delta + \Delta\theta) - d = 2r \tan \phi \left(\frac{\Delta w_{px}}{w_{px}} \right) \quad (\text{B.5})$$

Solving for r yields an equation to describe the estimated target range given the laser feature disparity measured within the camera view:

$$r = \frac{d}{\tan(\delta + \Delta\theta) - 2 \tan \phi \left(\frac{\Delta w_{px}}{w_{px}} \right)} \quad (\text{5.1})$$

APPENDIX C

LAFTR: CLOSE-PROXIMITY TARGET DETECTION AND RANGING

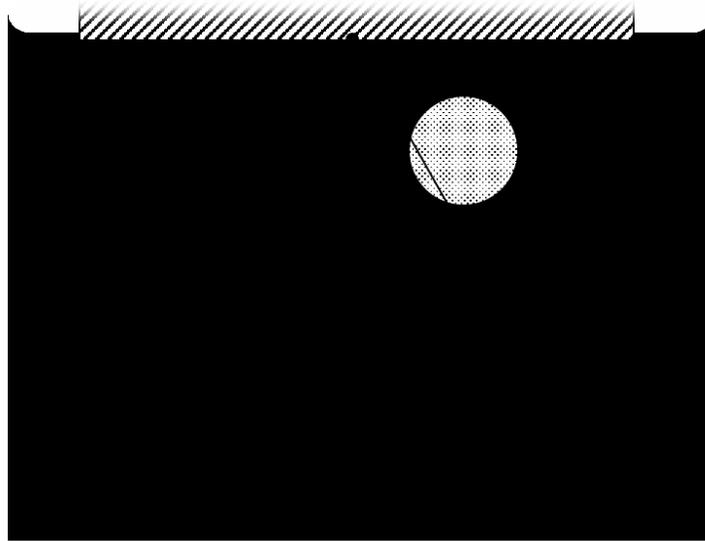


Figure C.1: Schematic of LAFTR close-proximity target detection and ranging

Given a similar setup to the free target detection method previously described, this method utilizes a background of known distance L to estimate the range to a target. As the target obstructs the camera's (C) view, the laser features emitted from the projector (P) appear to shift from R on the background to R' on the target surface. The pixel measurement of this offset feature can be related to the spatial geometry using the relation:

$$\frac{w_{px}}{2 \tan\left(\frac{\Phi}{2}\right)} = \frac{\Delta w_{px}}{\tan(\Delta\theta)} \quad (C.1)$$

The position vector of the laser marker on the target (R') relative to the camera (C) is found to be:

$$\overrightarrow{r_{R'/P}} = -n \cos(\delta) \hat{i} + n \sin(\delta) \hat{j} \quad (C.2)$$

The position vector of the target marker relative to the camera is:

$$\overrightarrow{r_{R'/C}} = \overrightarrow{r_{P/C}} + \overrightarrow{r_{R'/P}} = (d - n \cos(\delta)) \hat{i} + n \sin(\delta) \hat{j} \quad (C.3)$$

This position vector can also be defined as:

$$\overrightarrow{r_{R'/C}} = m \cos(\delta - \Delta\theta) \hat{i} + m \sin(\delta - \Delta\theta) \hat{j} \quad (C.4)$$

Relating the components of these vectors yields the equations:

$$\begin{aligned} d - n \cos(\delta) &= m \cos(\delta - \Delta\theta) \\ n \sin(\delta) &= m \sin(\delta - \Delta\theta) \end{aligned} \quad (C.5a, C.5b)$$

Written in matrix form, these equations are:

$$\begin{bmatrix} \cos(\delta - \Delta\theta) & \cos(\delta) \\ \sin(\delta - \Delta\theta) & -\sin(\delta) \end{bmatrix} \begin{bmatrix} m \\ n \end{bmatrix} = \begin{bmatrix} d \\ 0 \end{bmatrix} \quad (C.6)$$

Solving simultaneously for m yields:

$$m = \frac{d}{\cos(\delta - \Delta\theta) + \frac{\sin(\delta - \Delta\theta)}{\tan(\delta)}} \quad (C.7)$$

Thus, using

$$r = m \sin(\delta - \Delta\theta) \quad (C.8)$$

The equation for the range estimation becomes

$$r = \frac{d \sin(\delta - \Delta\theta)}{\cos(\delta - \Delta\theta) + \frac{\sin(\delta - \Delta\theta)}{\tan(\delta)}} \quad (5.2)$$

Where the angular shift is expressed with the geometric relationship:

$$\Delta\theta = \text{atan} \left(\frac{2\Delta w_{px} \tan\left(\frac{\phi}{2}\right)}{w_{px}} \right) \quad (5.3)$$

APPENDIX D

THERMAL ORTHOMOSAIC VIEWER CODE

```
function geotiffViewer(T_max,T_min)
    % Displays linearized grayscale result image and allows temp reading
    % Created 7/5/19 by Jamie Moon (jmmoon@crimson.ua.edu)
    % Revised 7/22/19
    % (C)2019 Jamie Moon, All Rights Reserved
    disp('Select Resulting GeoTiff from reconstruction');

    [file,path] = uigetfile({'*.tif','GeoTIFF Files (*.tif)'},...
        'Select Reconstructed Geotiff'); % Navigate to thermal data
    clc;
    if size(path,2) == 1
        error('Operation Terminated.');
```

```
    end
    filename = [path file];
    metadata = imfinfo(filename); % Get metadata
    ortho = imread(filename);
    ortho = ortho(:,:,1); % Get grayscale layer of image
    delta_temp = T_max - T_min;
    ortho2 = delta_temp.*(double(ortho)./255) + T_min; % Map radiometric
data to grayscale

    while 1 % NOTE: Needs a fix to allow user to easily escape loop
        disp('Select pixel. Press ENTER when finished.');
```

```
        imagesc(ortho2); % Display image as classic heat map (linear)
        axis equal; colormap jet;
        colorBar = colorbar;
        ylabel(colorBar, 'Temp. °C');
```

```
        [x,y] = getpts; % User selects pixel of interest
        close all; clc;
        x = round(x);
        y = round(y);
        temp = ortho2(y,x);
        disp(['Temperature = ' sprintf('%.1f',temp) '°C']); % Display
temperature
    end
end
```

APPENDIX E

PHASE OFFSET CALCULATOR CODE FOR UST

```
%% UST PHASE CALCULATOR
% Script for measuring phase shift of echoed 40kHz ultrasonic continuous
% wave
% The University of Alabama
% Department of Mechanical Engineering, Tuscaloosa, AL
%
% Requires "phdiffmeasure.m" script from Hristo Zhivomirov (see license)

% Created 12/1/18 by Jamie Moon (jmmoon@crimson.ua.edu)
% Revised 12/5/18

clc; clear all; close all; disp('Importing data...');
dataDir = 'E:\School Materials\BLAST\TBL Test 12-3-18\Oscope Data\';
tempData_fileName = 'E:\School Materials\BLAST\UST Data.xlsx';

xlsdata = xlsread(tempData_fileName,2,'A7:F81');
time = xlsdata(:,2);
tc_wall_temp = xlsdata(:,4);
tc_amb_temp = xlsdata(:,3);
ir_avg_temp = xlsdata(:,5);

first_run = 1;
mid_run = 20;
last_run = 75;

N_runs = last_run - first_run;
exportTrue = 0;

if exportTrue == 1 % Export plots as animation
    if exist('output.avi')
        delete output.avi;
    end
    aviObj = VideoWriter([dataDir 'output.avi'],'Motion JPEG AVI');
    aviObj.FrameRate = 2; % Output framerate [fps]
    aviObj.Quality = 95;
    n_images = N_runs;
    open(aviObj);
end

clc; disp('Processing...');
%figure;
for run = first_run:last_run
    [fileName1,fileName2] = nameFiles(run);
    data = dataImport(dataDir,fileName1, fileName2); % Import data
    deltaPhase(run) = phdiffmeasure(data(:,2), data(:,3));
end
```

```

        if deltaPhase(run) < 0
            deltaPhase(run) = 2*pi + deltaPhase(run);    % Wraparound negative
phase angles
        end
end
if exportTrue == 1
    close(aviObj);
end

% CORRECT DATA WRAPAROUND FOR WARMUP
dPhi_wu = deltaPhase(1:mid_run);
dPhi_wu = dPhi_wu - dPhi_wu(1); % Offset/zero data by first sample
for n = 1:length(dPhi_wu)
    if dPhi_wu(n) < 0
        dPhi_wu(n) = dPhi_wu(n) + 2*pi; % Wraparound negatives
    end
    if n > length(dPhi_wu)/4 && dPhi_wu(n) < pi/2
        dPhi_wu(n) = 2*pi - dPhi_wu(n); % Wraparound outliers
    end
end

% CORRECT DATA WRAPAROUND FOR COOLDOWN
dPhi_cd = deltaPhase(mid_run+1:last_run);
dPhi_cd = dPhi_cd - min(dPhi_cd(end-5:end)); % Offset/zero data by first
minimum sample
for n = 1:length(dPhi_cd)
    if dPhi_cd(n) < 0
        dPhi_cd(n) = dPhi_cd(n) + 2*pi; % Wraparound negatives
    end
end

%figure;
%polarplot(deltaPhase,tc_wall_temp,'o');

figure;    % Plot phase correlation
plot(tc_wall_temp(1:mid_run),radtodeg(dPhi_wu),'ok'); hold on;
plot(tc_wall_temp(mid_run+1:75),radtodeg(dPhi_cd),'*k'); hold off;
grid on; title('Correlation of Phase Angle and Surface Temperature');
legend('Warm Up','Cool Down');
xlabel('Wall Temperature [°C]'); ylabel('Phase Difference [°]');
axis([20 70 0 360]);
clc;

figure;
plot(time,tc_amb_temp,'-.k'); hold on;
plot(time,tc_wall_temp,'ok');
plot(time,ir_avg_temp,'*k');
title('Transient Target Temperature'); grid on;
xlabel('Time [min]'); ylabel('Temperature [°C]');
legend('Ambient Temp. (TC)','Wall Temp. (TC)','Wall Temp. (IR)');

%% FUNCTIONS

function [file1,file2] = nameFiles(n)
    % RETURN *.CSV FILENAME FOR nth RUN
    file1 = 'C1Run';
    file2 = 'C2Run';

```

```

if n >= 10
    file1(6:7) = num2str(n);
    file2(6:7) = num2str(n);
else
    file1(6:7) = ['0' num2str(n)];
    file2(6:7) = ['0' num2str(n)];
end
file1 = [file1 '.csv'];
file2 = [file2 '.csv'];
end

function data = dataImport(dir,file1,file2)
% RETURN DATA MATRIX WITH TIME(1) CH1(2) and CH2(3)
csv_data1 = csvread([dir,file1],5);
csv_data2 = csvread([dir,file2],5);
N_samples = length(csv_data1);
data = zeros(N_samples,3);
data(:,1) = csv_data1(:,1); % Time data
data(:,2) = csv_data1(:,2); % Channel 1 voltage
data(:,3) = csv_data2(:,2); % Channel 2 voltage
end

```

APPENDIX F

LAFTR RASPBERRY PI CODE FOR GRID METHOD

```
# LAFTR Motion Tracker
# For tracking moving objects using Pi camera and laser grid projector
# Created 11/26/18 by Jamie Moon (jmmoon@crimson.ua.edu)
# Revised 12/4/18 by Jamie Moon
# (C)2019 Jamie Moon, All rights reserved

#from imutils.video import VideoStream
#import imutils
import RPi.GPIO as GPIO
from picamera.array import PiRGBArray
from picamera import PiCamera
import argparse
import datetime
import time
from time import sleep
import cv2
import numpy as np
import csv

camera = PiCamera()
rawCapture = PiRGBArray(camera)

### INPUTS ### ### ### ### ### ###

blurSize = 5           # Radius for Gaussian blur [px]
#deltaThresh = 0.8     # Differential image intensity [0-1]
#dotThresh = 0.5       # Binary threshold for dot tracking [0-1]
dotBlockSize = 7      # Local Avg block size for adaptive dot thresholding
[px]
dotOffset = -10       # Threshold offset for adaptive dot thresholding [0-
255]
blobBlockSize = 5     # Local Avg block size for adaptive blob thresholding
[px]
blobOffset = -30      # Threshold offset for adaptive blob thresholding [0-
255]
shutterSpeed = 40     # Exposure time [s^-1]
resolution = 480      # Vertical resolution [px] [240,480,720,1080]
ledPin = 17           # Connect Laser switch to BCM17 on GPIO
buttonPin = 27        # Connect pushbutton to BCM27 on GPIO
ledWarmup = 0.015    # Time delay to ensure laser warmup [s]
dotArea_min = 4       # Minimum dot blob area for tagging
targetArea_min = 20   # Minimum target blob area
w = 5                 # Structural element width for dilation [px]
captureMode = 2       # Image record mode: 1=Single, 2=Continuous
L = 131               # Distance to background [cm]
```

```

### FUNCTIONS ### ### ### ### ### ### ###q

def main():
    firstFrame, args = setup()
    #print('Press clicker to begin')
    GPIO.output(ledPin, GPIO.LOW) # Ensure LED begins off
    #manualTrigger(firstFrame, args)
    mainLoop(firstFrame, args)

def setup():
    # INITIALIZATION FUNCTION
    print('Initializing...')
    initializeGPIO() # Initialize GPIO
    ap = argparse.ArgumentParser() # Command line argument options
    ap.add_argument("-v", "--video", help="Path to video file")
    ap.add_argument("-a", "--min-area", type=int, default=500, help="Minimum
area size")
    args = vars(ap.parse_args())
    if args.get("video", None) is None: # Default to picam if no input video
is selected
        piCamSetup(shutterSpeed, resolution, 1, 1, 'tungsten')
        #vs = VideoStream(usePiCamera=True).start()
        #vs = camera.capture_continuous(rawCapture, format='bgr',
use_video_port=True)
    else:
        vs = cv2.VideoCapture(args["video"]) # Switch to video read if
selected
        firstFrame = None # Initialize background frame with null
    return firstFrame, args

def manualTrigger(firstFrame, args):
    # MANUAL SENSOR TRIGGER
    # Starts and stops program using clicker
    ledOn = False
    continueLoop = True
    while True:
        if GPIO.input(buttonPin) == GPIO.HIGH and continueLoop == True:
            sleep(0.1)
            if ledOn == False:
                ledOn = True
                print('Running. Press clicker to stop')
                mainLoop(firstFrame, args)
            else:
                ledOn = False
                print('Stopped. Press clicker to continue')
                continueLoop = False
        elif GPIO.input(buttonPin) == GPIO.LOW:
            sleep(0.1)
            continueLoop = True

def mainLoop(firstFrame, args):
    GPIO.output(ledPin, GPIO.HIGH) # Pulse LED/laser during capture
    sleep(ledWarmup)
    # PRIMARY LOOP FUNCTION
    i = 0
    startTime = time.time()

```

```

    for f in camera.capture_continuous(rawCapture, format='rgb',
use_video_port=True):
        frame = f.array
        elapsedTime = round(time.time() - startTime,3) # Timestamp
        frame = frame if args.get("video",None) is None else frame[1]
        GPIO.output(ledPin,GPIO.LOW)
        if frame is None:
            break
        imgGray = cv2.cvtColor(frame, cv2.COLOR_RGB2GRAY) # Convert to
grayscale
        imgGray0 = cv2.GaussianBlur(imgGray, (w, w), 0) # Apply small
blur for NR

        dotBin = dotThresholder(imgGray0,dotBlockSize,dotOffset)
        dotLoc,dotOverlay = locateDots(frame,dotBin,dotArea_min) # Track dots
        #cv2.imshow('Binary Dots',dotBin)

        imgGray = cv2.GaussianBlur(imgGray, (blurSize, blurSize), 0) #
Apply Gaussian blur

        if firstFrame is None or i == 2: # Initialize background image
            firstFrame = imgGray # NB: Includes workaround to discard first
frame from cold start
            #i = 0
            #cv2.imshow("Reference Frame",firstFrame)

            deltaFrame = cv2.absdiff(firstFrame,imgGray) # Differential image
            #cv2.imshow('Differential Frame',deltaFrame)

            imgBin = blobThresholder(deltaFrame,blobBlockSize,blobOffset) #
Differential binary image
            kernel = np.ones((w,w),np.uint8) # Structural element for dilation
            imgBin = cv2.dilate(imgBin, kernel, iterations=2) # Apply dilation
            #cv2.imshow('moTrack Binary',imgBin)

            xt,xs = trackTargets(imgBin,targetArea_min,dotOverlay,dotLoc) # Track
targets

            #cv2.line(dotOverlay, (xt,0), (xt,480), (255,0,0),2)
            #cv2.line(dotOverlay, (xs,0), (xs,480), (0,255,0),2)

            if xs >= 100:
                #detectRange(xt-xs) # Display estimated target range
                stampRange(detectRange(xt-xs,L),dotOverlay)

            stampTime(elapsedTime,dotOverlay)
            cv2.imshow('Tracked targets',dotOverlay)

            rawCapture.truncate(0) # Clear buffer for next frame
            if GPIO.input(buttonPin) == GPIO.HIGH: # Export current frame
                if captureMode == 1:
                    exportFrame(dotOverlay)
                elif captureMode == 2:
                    cv2.imwrite('sequence_' + str(i) + '.jpg',dotOverlay)
                    print('sequence_' + str(i) + '.jpg saved')
            key = cv2.waitKey(1) & 0xFF # End stream upon pressing "q" key
            if key == ord('q'):# or GPIO.input(buttonPin) == GPIO.HIGH:

```

```

        GPIO.output(ledPin,GPIO.LOW)
        break
        GPIO.output(ledPin,GPIO.HIGH) # Pulse LED/laser during capture
        sleep(ledWarmup) # Delay for laser warmup
        i += 1

def piCamSetup(expTime, rez, hFlipTrue, vFlipTrue, wb):
    # INITIALIZE PICAMERA SETTINGS
    if rez == 240:
        camera.resolution = (320,240)
    elif rez == 480:
        camera.resolution = (640,480)
    elif rez == 720:
        camera.resolution = (1280,720)
    elif rez == 1080:
        camera.resolution = (1920,1080)
    elif rez == 2464:
        camera.resolution = (3280,2464)
    #camera.framerate = 32
    camera.exposure_mode = 'auto'
    sleep(1)
    camera.exposure_mode = 'off'
    sleep(1)
    camera.video_stabilization = True
    camera.hflip = hFlipTrue
    camera.vflip = vFlipTrue
    camera.awb_mode = wb
    camera.shutter_speed = int(1e6/shutterSpeed)

def trackTargets(imgBin,A_min,img_out,dotLocations):
    # DETECT AND TRACK TARGETS WITH BOUNDING BOX
    # Input binary differential image, Minimum blob area, image for output,
    and tracked dot locations
    im2,ct,hierarchy = cv2.findContours(imgBin.copy(),
cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
    x_targ = 0
    x_shad = 0
    ind_targ = 0
    ind_shad = 0
    for c in ct:
        if cv2.contourArea(c) < A_min: # Track objects above minimum size
            continue
        (x,y,w,h) = cv2.boundingRect(c)
        cv2.rectangle(img_out, (x,y), (x+w,y+h), (0,255,0),2) # Draw blue
        bounding boxes around target
        for n in dotLocations:
            if n[0] >= x and n[0] <= x+w:
                if n[1] >=y and n[1] <= y+h:
                    cv2.rectangle(img_out, (x,y), (x+w,y+h), (255,0,0),2) # Draw
                    green boxes around shadows
                    x_shad = x_shad + x + int(round(w/2))
                    ind_shad += 1
            else:
                x_targ = x_targ + x + int(round(w/2))
                ind_targ += 1
    if x_shad != 0:
        x_shad = int(round(x_shad/(ind_shad + 1)))

```

```

    if x_targ != 0:
        x_targ = int(round(x_targ/(ind_targ + 1)))
    return x_shad, x_targ

def locateDots(img_out, imgBin, A_min):
    # LOCATE AND TAG DOTS
    # Input original image (for overlay) and processed binary image
    highlighting dots
    # Returns array with dots' X-Y coordinates and overlay image
    # Enable export to save preview image and dot coordinates as CSV
    im2, contours, hierarchy =
cv2.findContours(imgBin, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE) # Locate
edges of dot blobs
    dotLocations = np.array([0,0])
    for c in contours: # Draw dots on centroids of target
        M = cv2.moments(c)
        if M["m00"] != 0 and cv2.contourArea(c) >= A_min:
            cX = int(M["m10"] / M["m00"])
            cY = int(M["m01"] / M["m00"])
            cv2.circle(img_out, (cX, cY), 1, (0,0,255), -1)
            dotLocations = np.vstack([dotLocations, [cX, cY]])
    return dotLocations, img_out

def dotThresholder(img_in, blockSize, offset):
    # CONVERTS IMAGE TO BINARY FOR DOT DETECTION
    # Returns binary image
    # Input image, average block size, and threshold offset
    #ret, thresh = cv2.threshold(img_in, int(255*(1-
level)), 255, cv2.THRESH_BINARY)
    #ret, thresh =
cv2.threshold(img_in, 0, 255, cv2.THRESH_BINARY+cv2.THRESH_OTSU)
    thresh = cv2.adaptiveThreshold(img_in, 255, cv2.ADAPTIVE_THRESH_MEAN_C, \
cv2.THRESH_BINARY, blockSize, offset)

    return thresh

def blobThresholder(img_in, blockSize, offset):
    # CONVERTS IMAGE TO BINARY FOR MOTION TRACKING
    # Returns binary image
    # Input image, average block size, and threshold offset
    #ret, thresh = cv2.threshold(img_in, int(255*(1-
level)), 255, cv2.THRESH_BINARY)
    #ret, thresh =
cv2.threshold(img_in, 0, 255, cv2.THRESH_BINARY+cv2.THRESH_OTSU)
    thresh = cv2.adaptiveThreshold(img_in, 255, cv2.ADAPTIVE_THRESH_MEAN_C, \
cv2.THRESH_BINARY, blockSize, offset)

    return thresh

def initializeGPIO():
    # INITIALIZE GPIO PINS
    GPIO.setmode(GPIO.BCM) # Pin nomenclature
    GPIO.setwarnings(False)
    GPIO.setup(ledPin, GPIO.OUT)
    GPIO.setup(buttonPin, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
    GPIO.output(ledPin, GPIO.LOW) # Ensure LED starts OFF

def exportFrame(img):
    # EXPORTS CURRENT FRAME

```

```

GPIO.output(ledPin,GPIO.LOW)
filename = raw_input('File name to export current image: ') + '.jpg'
cv2.imwrite(filename,img)
print('Export complete')

def stampTime(time,img):
    # LABEL CURRENT FIGURE WITH TIME
    cv2.putText(img,str(time) + ' s',(10,resolution-
10),cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),1)

def stampRange(r,img):
    # LABEL CURRENT FIGURE WITH ESTIMATED TARGET RANGE
    cv2.putText(img,'Target at ' + str(round(r)) + ' cm',(100,resolution-
10),\
                cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),1)

def detectRange(deltaX,L):
    # DETECT RANGE in cm FROM PARALLAX
    w = 30 # Space between camera sensor and dot projector [cm]
    #L = 116 # Distance from sensor to background [cm]
    theta = np.arctan(L/(w/2)) # Incident angle of projector [rad] normal to
background
    wp = 640 # Width of image [px]
    phi = 1.3439 # Horizontal camera fov [rad]
    deltaPhi = np.arctan((2*deltaX*np.tan(phi/2))/wp) # Compute angular
parallax
    d = (w*np.sin(theta - deltaPhi))/(np.cos(theta - deltaPhi) + \
        (np.sin(theta-deltaPhi))/np.tan(theta))
# Estimate target range [cm]
#print('Target distance = \t' + str(round(d)) + ' cm')
if d < L: # BS filter
    return d
elif d == 0 or d > L:
    d = L
    return d

main()
GPIO.output(ledPin,GPIO.LOW)
#vs.stop() if args.get('video',None) is None else vs.release() # Stop
stream
cv2.destroyAllWindows() # Close residual windows
print('Done.')

```