NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

DETECTION OF SMALL UNMANNED AERIAL SYSTEMS USING A 3D LIDAR SENSOR

by

Konstantinos Paschalidis

September 2021

Thesis Advisor: Roberto Cristi
Co-Advisor: Oleg A. Yakimenko

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE				*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.				
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2021	**3. REPORT TYPE AND DATES COVERED** Master's thesis		
4. TITLE AND SUBTITLE DETECTION OF SMALL UNMANNED AERIAL SYSTEMS USING A 3D LIDAR SENSOR				**5. FUNDING NUMBERS**
6. AUTHOR(S) Konstantinos Paschalidis				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000				**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A				**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.				**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

Small unmanned aerial systems (sUAS) are a rapidly developing technology with countless applications in many areas of human activity, ranging from commercial to military use. In the latter case, counter-UAS operations have become an urgent issue. The problem is that the small size of a sUAS makes its detection quite a challenging task. Many of traditional approaches and technologies may not be applicable at all. This thesis describes a feasibility study for using a stationary 3D 360° Light Detection and Ranging (LiDAR) sensor to detect a fast-moving sUAS. Specifically, a low-end Velodyne Puck Hi-Res LiDAR was used to collect data during a series of flight tests involving different size sUASs at two rural locations. The thesis presents an analysis of the LiDAR output and the developed algorithms to detect a moving sUAS despite several challenges associated with a rich, nonstationary background return. These challenges were overcome by using Principal Components Analysis (PCA) as well as masking. The developed algorithm demonstrated that using a low-end LiDAR with a detection range of about 100 m, it is possible to detect a sUAS of about a 0.3 m cross-section, isolate it from other moving objects, and track it while as it maneuvers within a 25 m range. Obviously, using the same algorithm with a higher resolution LiDAR would allow detection at the higher ranges, thus making LiDAR-based counter-UAS technology a viable candidate for protecting against a UAS threat.

14. SUBJECT TERMS counter-sUAS, 3D 360 degrees LiDAR, motion detection, Principal Components Analysis, masking			**15. NUMBER OF PAGES** 77
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

DETECTION OF SMALL UNMANNED AERIAL SYSTEMS USING A 3D LIDAR SENSOR

Konstantinos Paschalidis
Lieutenant Colonel, Hellenic Army
MEng, National Technical University of Athens, 2010

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2021

Approved by: Roberto Cristi
Advisor

Oleg A. Yakimenko
Co-Advisor

Douglas J. Fouts
Chair, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Small unmanned aerial systems (sUAS) are a rapidly developing technology with countless applications in many areas of human activity, ranging from commercial to military use. In the latter case, counter-UAS operations have become an urgent issue. The problem is that the small size of a sUAS makes its detection quite a challenging task. Many of traditional approaches and technologies may not be applicable at all. This thesis describes a feasibility study for using a stationary 3D 360° Light Detection and Ranging (LiDAR) sensor to detect a fast-moving sUAS. Specifically, a low-end Velodyne Puck Hi-Res LiDAR was used to collect data during a series of flight tests involving different size sUASs at two rural locations. The thesis presents an analysis of the LiDAR output and the developed algorithms to detect a moving sUAS despite several challenges associated with a rich, nonstationary background return. These challenges were overcome by using Principal Components Analysis (PCA) as well as masking. The developed algorithm demonstrated that using a low-end LiDAR with a detection range of about 100 m, it is possible to detect a sUAS of about a 0.3 m cross-section, isolate it from other moving objects, and track it while as it maneuvers within a 25 m range. Obviously, using the same algorithm with a higher resolution LiDAR would allow detection at the higher ranges, thus making LiDAR-based counter-UAS technology a viable candidate for protecting against a UAS threat.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION 1
  A. COUNTER UNMANNED AERIAL SYSTEM OPERATIONS AND CHALLENGES IN DETECTING SMALL UNMANNED AERIAL SYSTEMS 1
  B. PREVIOUS RESEARCH ON THE TOPIC AND KEY RESULTS 5
  C. PROBLEM FORMULATION AND THESIS OUTLINE 9

II. THE BASICS OF 3D LIDAR TECHNOLOGY 11
  A. BASIC CONCEPT AND APPLICATIONS 11
  B. HARDWARE COMPONENTS OF THE 3D 360° LIDAR SENSOR 15
  C. SOFTWARE FOR PROCESSING LIDAR DATA 19

III. DATA COLLECTION 27
  A. TEST SETUP 27
  B. METHODOLOGY FOR EVALUATION OF SUAS DETECTION BY LIDAR SENSOR 31
  C. DATA COLLECTION PROCEDURE 33
  D. ANALYSIS OF THE RAW DATA 35

IV. DEVELOPMENT OF THE SUAS DETECTION ALGORITHM 39
  A. KEY FEATURES OF THE DEVELOPED ALGORITHM 39
  B. COMPUTER SIMULATIONS 41
  C. FLIGHT TEST DATA PROCESSING 45
  D. EVALUATION OF THE RESULTS COMPARISONS 46

V. CONCLUSIONS 51
  A. CONCLUSIONS 51
  B. RECOMMENDATIONS – FUTURE RESEARCH 51

LIST OF REFERENCES 53

INITIAL DISTRIBUTION LIST 57

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure 1. Flying sUAVs. Source: [1]. 2

Figure 2. Different mini/micro UAVs. Source: [2]. 2

Figure 3. UAV images from TV-camera, at various distances. Source: [4]. 3

Figure 4. UAV images from Bolometer, at various distances. Source: [4]. 3

Figure 5. Acoustic sensor. Source: [1]. 4

Figure 6. Multi-sensor network. Source: [1]. 4

Figure 7. Sensor platform equipped with several sensors. Source: [2]. 6

Figure 8. Experimental results for optical sensing. Source: [1]. 7

Figure 9. Comparisons on LRCS of UAVs. Source: [2]. 8

Figure 10. Theoretical scan patterns at different distances. Source: [2]. 8

Figure 11. Image produced from a LiDAR sensor that shows the route of an UAV. Adapted from [1] (colors inverted). 9

Figure 12. Images that show the frequency of the presence of aerosol samples classified as polluted dust. Source: [7]. 12

Figure 13. View of the Naval Postgraduate School campus obtained from an airborne LiDAR system. Source: [6]. 13

Figure 14. Example of Airborne Laser Terrain Mapping (ALTM) data showing vegetation removal. Source: [8]. 13

Figure 15. Simple LIDAR example, pulse return. Source: [10]. 14

Figure 16. LiDAR image of Niagara Falls. Source: [10]. 15

Figure 17. General overview of the proposed LiDAR system. Source: [5]. 16

Figure 18. Point density in one frame (a) and in series of successive frames (b). Adapted from [11] (colors inverted). 17

Figure 19. Sensors coordinate system. Source: [11]. 18

Figure 20. Dual Return example (last and strongest reflections). Source: [11]. 19

Figure 21. Forestry application with multiple returns. Source: [11]. 19

Figure 22. Overview of Ouster Studio's graphical interface. Adapted from [12] (colors inverted). 20

Figure 23. Spreadsheet view. Adapted from [12] (colors inverted). 21

Figure 24. Images produced by the same point cloud, differentiated by the attributes used for coloring. Adapted from [12] (colors inverted). 21

Figure 25. Behavior of the cropping in "Spherical" Mode. Adapted from [12] (colors inverted). 22

Figure 26. Point cloud clusters (distinguished by different colors). Adapted from [21] (colors inverted). 24

Figure 27. Semantic segmentation of point clouds. Adapted from [16] (colors inverted). 25

Figure 28. The Velodyne Puck Hi-Res LiDAR sensor. Source: [25]. 27

Figure 29. sUAVs with a maximum dimension of less than 60 cm used in the experiments. 29

Figure 30. sUAVs with a maximum dimension of more than 60 cm used in the experiments. 29

Figure 31. LiDAR sensor and some of the auxiliary equipment set up in the test field. 30

Figure 32. The environment at the NPS Test Site at Marina. 31

Figure 33. The environment at the NPS Test Site at Marina. 31

Figure 34. Flow chart of the methodology applied in this research. 33

Figure 35. In-flight images of sUAVs with a maximum dimension of more than 60 cm. 34

Figure 36. In-flight images of sUAVs with a maximum dimension of less than 60 cm. 35

Figure 37. Structure of the single return mode data packet. Source: [11]. 36

Figure 38. Example of the start of a single return mode data packet. Source: [11]. 36

Figure 39. Example of the ending of a single return mode data packet. Source: [11]. 37

Figure 40. Visualization of the LiDAR data **before** the application of the algorithm for frame with ID 203. 42

Figure 41. Visualization of the LiDAR data **after** the application of the algorithm for frame with ID 203. 42

Figure 42. Relationship between the number of points that sUAVs covered in a frame and the sUAVs' distance from the LiDAR sensor. 46

Figure 43. Relationship between the percentage of sUAVs detected and their distance from the LiDAR sensor. 47

Figure 44. Relationship between the number of points that sUAVs covered in a frame and their velocity. 48

Figure 45. Relationship between the number of points that sUAVs covered in a frame and their altitude. 49

Figure 46. False detection rate per 100 frames with height constraint and depth mask applied 50

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1. Specifications of the Velodyne Puck Hi-Res LiDAR sensor. Source: [24]. 28

Table 2. Printed results that show the route of the cluster with clusterID 2. 43

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

ALTM	Airborne Laser Terrain Mapping
API	application programming interface
CCD	charge-coupled device
FOV	field of view
FPGA	field-programmable gate array
IR	infrared
Laser	light amplification by stimulated emission of radiation
LiDAR	light detection and ranging
LRCS	laser radar cross section
MBA	Monterey Bay Academy Airfield
PCA	principal component analysis
pcap	packet captures
PCL	Point Cloud Library
RADAR	radio detection and ranging
sUAS	small unmanned aerial system
sUAV	small unmanned aerial vehicle
SWIR	short-wave infrared
UAS	unmanned aerial system
UAV	unmanned aerial vehicle
UPS	uninterruptible power source

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to express my gratitude to my thesis advisors, Professor Roberto Cristi and Professor Oleg Yakimenko, for it was under their encouragement that I was motivated to undertake this thesis research. I really appreciate their support, the trust that they showed me, and the valuable advice and guidance they provided me during the entire process of our flawless cooperation.

Also, I would like to thank my colleagues and friends, Madjer and Mariano, who were my fellow travelers in this great and productive journey to knowledge during our studies at the Naval Postgraduate School.

Furthermore, I would like to thank my family who raised me and established the foundations for this achievement. My beloved wife, Dimitra, and my precious children, Charitina and Paschalis: I am thankful for the patience and support they provided me during the challenging and constant efforts to achieve my goals, but most of all because they are my continuous inspiration of my life.

This research is dedicated to my father’s memory.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

This chapter presents the context of this research. First, it describes the status and challenges of the topic. Then it presents a brief description of previous related research and the main results of those studies. Finally, this chapter includes the formulation of the problem and the thesis outline.

While the term “Unmanned Aerial System” (UAS) refers to the system composed of the unmanned aerial vehicle (UAV) itself, its payload (sensors, communication devices, etc.) and ground base station, this thesis is about detecting the vehicle itself. However, it is a common practice to use the term UAS to denote just a UAV, hence this thesis will also use the UAS notation.

A. COUNTER UNMANNED AERIAL SYSTEM OPERATIONS AND CHALLENGES IN DETECTING SMALL UNMANNED AERIAL SYSTEMS

Small unmanned aerial vehicles (sUAVs) have become a serious threat, in both civilian and military areas [1]. During recent years, many reported incidents have involved sUAVs in situations that are threatening the security, safety, and privacy of areas of either public or private interest [1]-[3]. The threat of sUAVs has increased due to the worldwide availability of cheap sUAVs in combination with the ease of operating them [2], [3]. Consequently, the issue of detecting sUAVs is a major concern worldwide [1].

Some representative images of sUAVs are shown in Figures 1 and 2.

Figure 1. Flying sUAVs. Source: [1].

Figure 2. Different mini/micro UAVs. Source: [2].

The aforementioned concern has led to extended research on the possible ways of detecting sUAVs. One technique employs processing camera-based images [2], [3]. These images can be taken by standard cameras in the visible range, as shown in Figure 3, or by short-wave infrared (SWIR) cameras, as shown in Figure 4, in which case the quality of the images is a crucial issue affecting the detection results [2], [3].

5 m
235 pixel

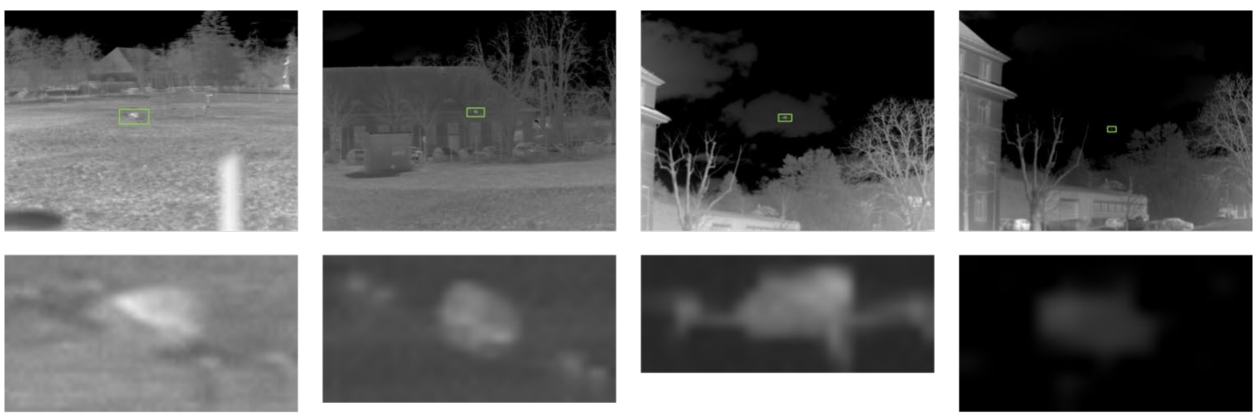
10 m
131 pixel

19 m
92 pixel

27 m
62 pixel

First row: overall view where the UAV is bounded by a green box

Figure 3. UAV images from TV-camera, at various distances. Source: [4].



5 m
68 pixel

10 m
38 pixel

19 m
27 pixel

27 m
17 pixel

First row: overall view where the UAV is bounded by a green box.

Figure 4. UAV images from Bolometer, at various distances. Source: [4].

One other approach involves radar sensors which are greatly impacted by the low laser-radar cross section (LRCS) of the majority of sUAVs [2], [3]. Also interesting is the technique that uses acoustical sensors for detecting the desired targets [2], [3], as shown in Figure 5.

Figure 5. Acoustic sensor. Source: [1].

Active imaging cameras also can be applied; however, while this method presents some advantages compared to charge-coupled device (CCD) cameras, it requires knowledge of the distance between the sensor and the UAV [2], [3].

Furthermore, Light Detection and Ranging (LiDAR) technology represents another promising method in this field [2], [3]. Finally, more sophisticated methods could be developed by combining multi-sensor networks in order to detect and track sUAVs [1]–[3], as shown in Figure 6.

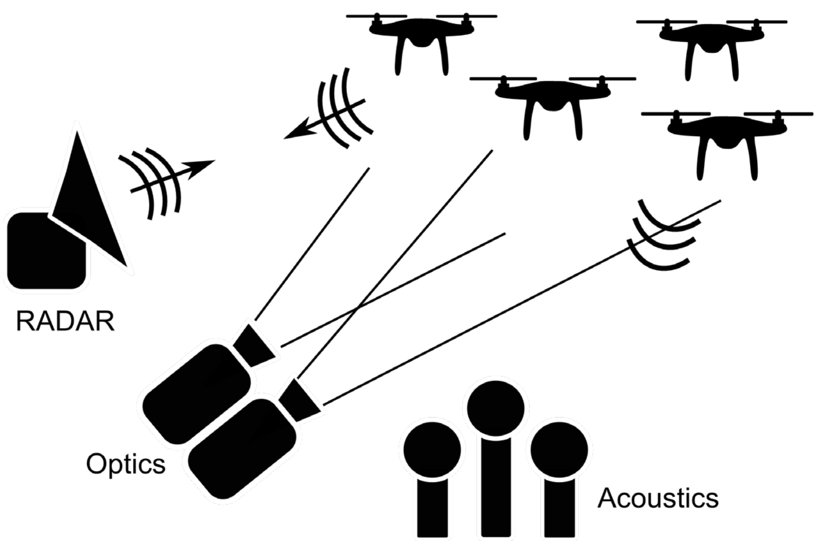


Figure 6. Multi-sensor network. Source: [1].

Most of the methods just mentioned are studied and applied to many other fields that involve detection, but there are crucial peculiarities in the case of sUAVs that make this process challenging. First, we have to deal with objects that are small in size [2], [3]. Also, these flying objects present a large range of acceleration, speed, and maneuverability in all dimensions (3D), which makes it even more difficult to detect or predict their route [2], [3]. Additionally, one other crucial feature is the small LRCS that sUAVs usually present, which make it difficult for active sensors to detect them (e.g., radar, LiDAR) [2], [3]. Moreover, the large variety of the forms of sUAVs does not permit us to classify the desired target according to a specific shape.

Considering that LiDAR sensors are studied in this research, we will highlight the features that make detection of sUAVs difficult. In particular, LiDAR sensors present low resolution that consequently hinders the detection of the small flying objects [2], [3]. Also, the limited field of view (FOV) of these sensors has a negative impact on their ability to detect sUAVs [2], [3]. Moreover, we should notice that because LiDAR sensors are active sensors; their success is negatively affected by the low LRCS of sUAVs [2], [3].

B. PREVIOUS RESEARCH ON THE TOPIC AND KEY RESULTS

Although the capabilities of LiDAR are very promising in the field of detecting sUAS, there is a paucity of literature detailing the results of such research as compared to other sensors like cameras and radars. This may be because LiDAR is a newer technology or because LiDAR is an expensive sensor with a limited range for appropriate resolution. Nevertheless, there are some interesting results from the available research that provide the basis for this thesis.

Most of the available papers on this topic use a collection of sensors that vary in number either to cooperate with each other or to compare their effectiveness [1]–[4]. The main type of LiDAR sensors used for the detection of sUAS are sensors that consist of an array of laser transmitters alternated with laser receivers, which turn 360°around a vertical axis [5], with a maximum range of approximately 100 m [2], [3]. In some cases, a set of LiDAR sensors is used [3] to increase sensor sensitivity and effectiveness. Other types of sensors are also used for collaboration with LiDAR sensors [4], increasing the efficiency in detecting the desired targets. Also, in many experiments different types of sensors are used that provide comparable results in terms of efficiency for detecting sUAVs [1]. A representative example of such a collection is illustrated in Figure 7, and the results of such sets of sensors are shown in Figure 8.

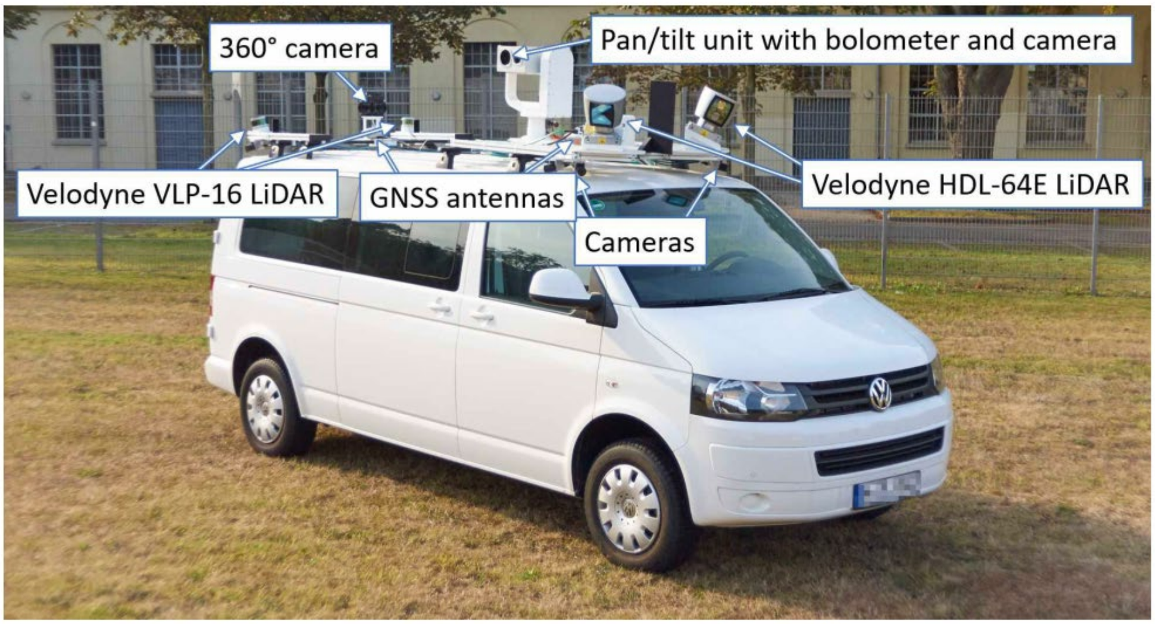
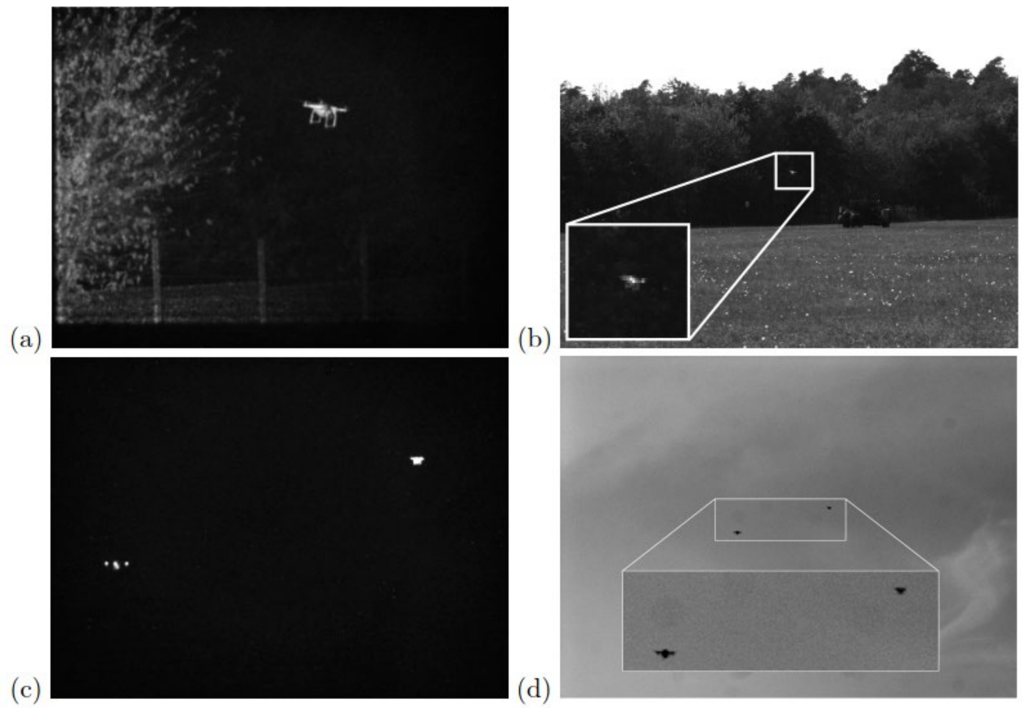


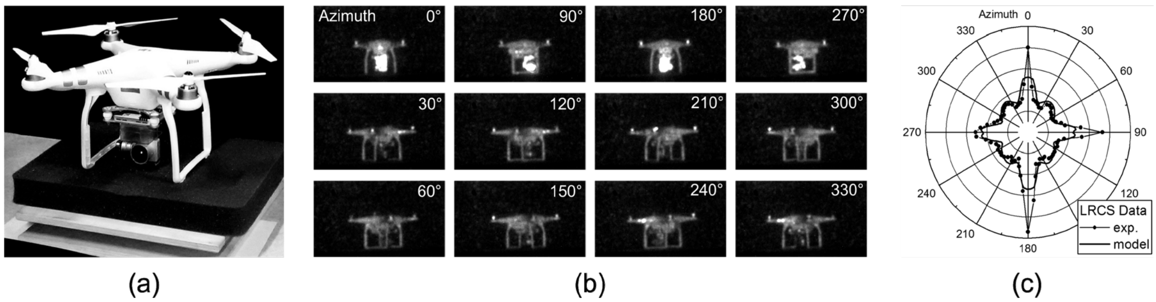
Figure 7. Sensor platform equipped with several sensors.
Source: [2].



Images from experiments of optical sensing: (a) UAV image in textured background, (b) UAV image from a laser gate, (c) UAV images from passive vision sensors, and (d) UAV images from passive imaging sensors.

Figure 8. Experimental results for optical sensing. Source: [1].

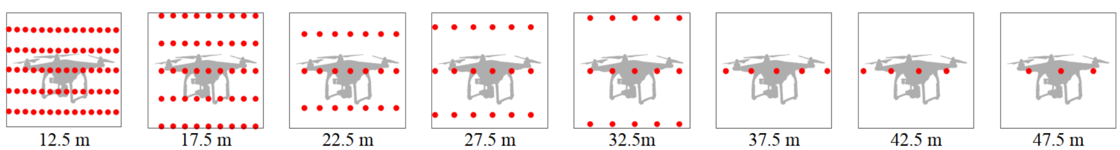
Generally, the primary results of the experiments just described lead to similar conclusions. First, it is a common conclusion that because of the bounded FOV of the LiDAR sensors only a small percentage of their scans is capable of detecting sUAVs [2], [3]. Also, the small LRCS that sUAVs have contributes to a significant decrease in the range at which they can be detected by the sensors [3]. An example of the impact of the position of the UAV to the LRCS is demonstrated in Figure 9.



Images that show the results of comparisons on LRCS of an actual UAV and theoretical models: (a) UAV, (b) UAV at various perspectives, and (c) the results of the LRCS comparisons.

Figure 9. Comparisons on LRCS of UAVs. Source: [2].

Finally, the small size of the sUAS and the limited resolution of the LiDAR sensors have a direct consequence in the reduction of the detection range of the target [3]. A characteristic visualization of the consequences due to the relation between the range and the size and resolution is shown in Figure 10.



Theoretical scan patterns related to the distances between the target and the LiDAR sensor, with a target outline

Figure 10. Theoretical scan patterns at different distances. Source: [2].

The detection rate of the sUAS when using LiDAR sensors is greatly affected by the range of operation. In particular, when the distance between the target and the sensor increases above 30 m, the detection rate decreases significantly [2], [3]. Hence, the results of these studies illustrated the efficiency of using LiDAR sensors by presenting the detection rate of the targets under different scenarios. The main parameter that seems to impact this efficiency is the range, while other parameters like the light conditions that have considerable influence on other types of sensors (e.g., cameras) [4] have a negligible impact on LiDAR sensors. Figure 11 presents the trace of an approaching UAV when a LiDAR sensor is used.

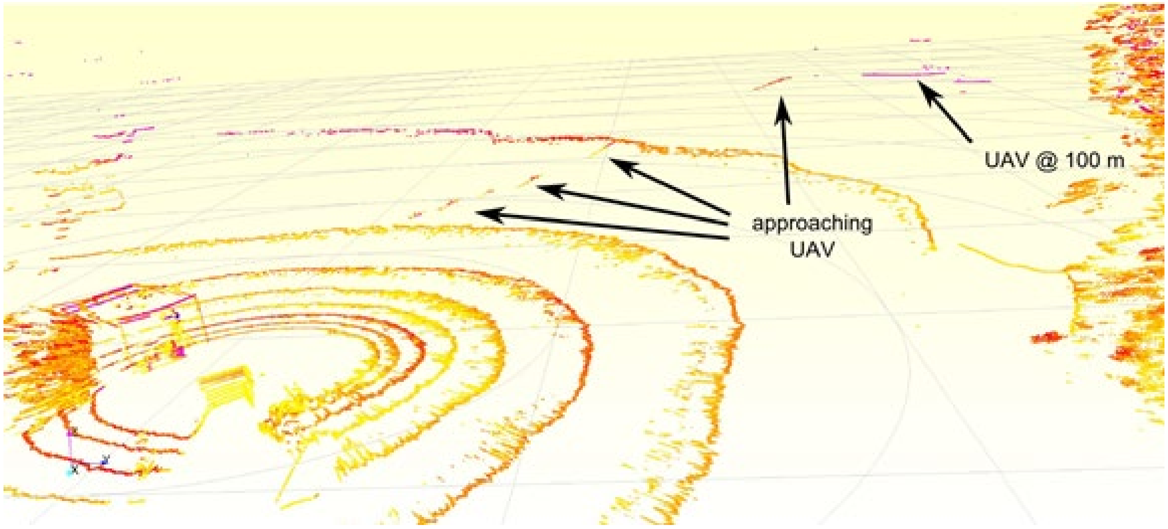


Figure 11. Image produced from a LiDAR sensor that shows the route of an UAV. Adapted from [1] (colors inverted).

C. PROBLEM FORMULATION AND THESIS OUTLINE

Considering the previous research results and by making a pragmatic estimation of the limitations and the challenges of the sUAS detection by using LiDAR sensors, we can formulate realistic goals for this research. In particular, the goal of this research is to create an efficient algorithm that would be suitable to distinguish and detect objects that are candidates for being an sUAS. The main features of such objects are their small size, their motion in three dimensions, and their distance from all the surrounding objects. Accordingly, this study excludes all the objects that may present similar characteristics but cannot be an sUAS. Also, significant to this study is the fact that LiDAR sensors provide low resolution, which consequently could skew our findings. This study focuses on detection of an sUAS in a rural environment, which increased the difficulty of detection because of the presence of not only uneven ground but also plants and trees. Plants and trees, because of their continuous movement due to wind and the fact that their surfaces are not consistent, are sources of multiple false target detections.

This thesis consists of five chapters. In Chapter II, we introduce the 3D LiDAR technology. Next, Chapter III we describe the data collection experiments that we performed and the methods that we applied to these experiments. In Chapter IV, we make an analytic presentation of the detection algorithm used and the results of the processing of the collected data. Finally, Chapter V presents the conclusions drawn from this research and recommends areas for future research on this topic.

II. THE BASICS OF 3D LIDAR TECHNOLOGY

This chapter presents some basic information on 3D LiDAR technology. In addition to an explanation of the relevant concepts, the chapter describes the hardware underpinning this technology. Next, the discussion proceeds with an explanation of the integration of those hardware components and the available software for processing the data collect by this technology.

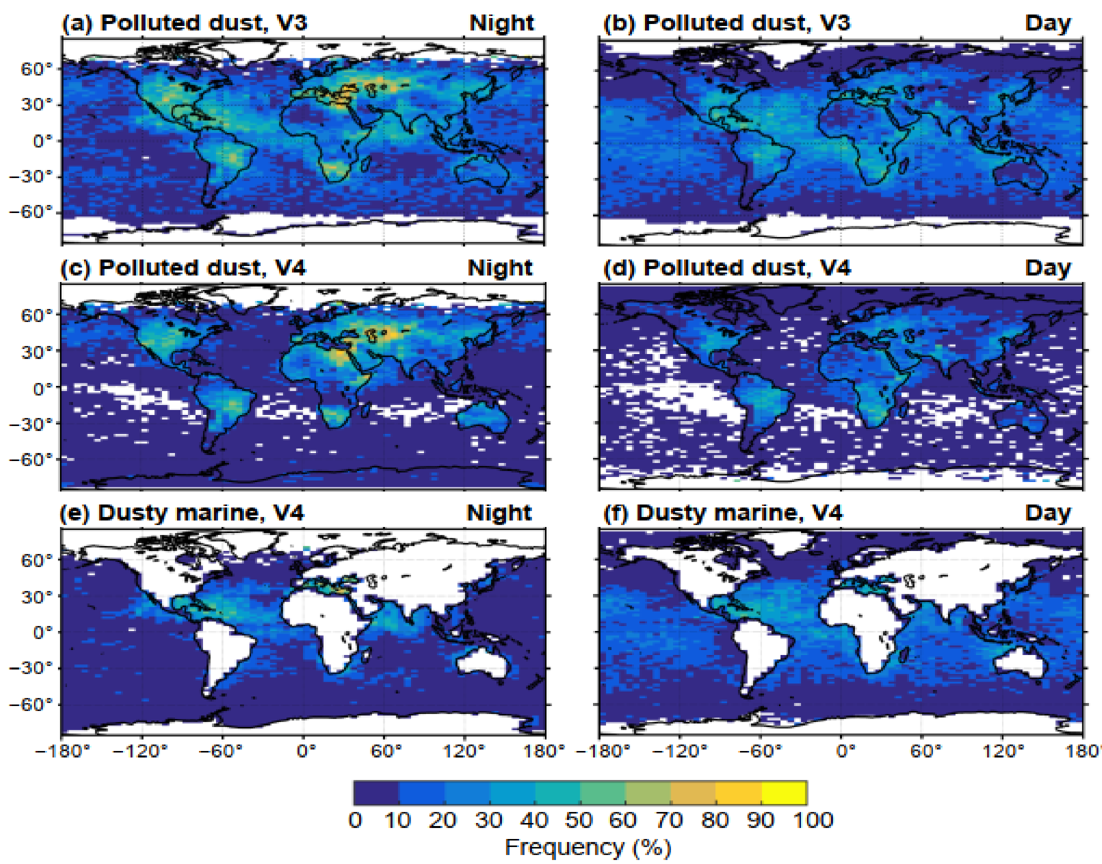
A. BASIC CONCEPT AND APPLICATIONS

First, a brief description of the nature and the capabilities of the 3D LiDAR technology can help to understand this research. In particular, the fundamental technology behind 3D LiDAR is laser technology. The implementation of LiDAR sensors is similar to that of radar sensors, although it presents many differences. Also, because of the extensive variations in LiDAR capabilities, there is a wide field of uses for this technology.

Laser technology is not a new concept. Actually, the term “laser,” which is an acronym drawn from “Light amplification by stimulated emission of radiation,” was described as early as the 1950s by Townes and Schalow [1]. This emitting radiation has some characteristic features that makes it well suited for remote sensing applications. Primarily, the radiation from each laser beam is monochromatic (i.e., it presents a unique frequency) [1], so it is easily recognizable and distinguishable. Furthermore, each laser beam does not spread significantly over distance and retains its narrow beam width [1]. Finally, we should highlight the capability of this kind of radiation to effectively perform successive switching between starting and stopping the emission of radiation [6].

LiDAR is an application of laser technology, and its functionality shares many similarities with radar (Radio Detection And Ranging) applications [1]. The main differences between LiDAR and radar are based on the aforementioned characteristic features of laser radiation that in combination with the shorter wavelengths [1] provide this technology and its products with very interesting and fruitful capabilities. Thus, there are many different areas suited to LiDAR applications, extending from those used for observing the dust and the aerosols in the atmosphere to those that are used for remote sensing of the surface and subsurface of the earth [6].

Typical examples of observing the dust and the aerosols in the atmosphere are shown in Figure 12, while in Figures 13 and 14 examples of remote sensing of the surface and subsurface of the earth, respectively, can be seen.



Frequency of occurrence of aerosol samples classified as polluted dust in V3 at night and during the day (a, b), polluted dust in V4 at night and during the day (c, d) and dusty marine in V4 at night and during the day (e, f). June–August 2007.

Figure 12. Images that show the frequency of the presence of aerosol samples classified as polluted dust. Source: [7].

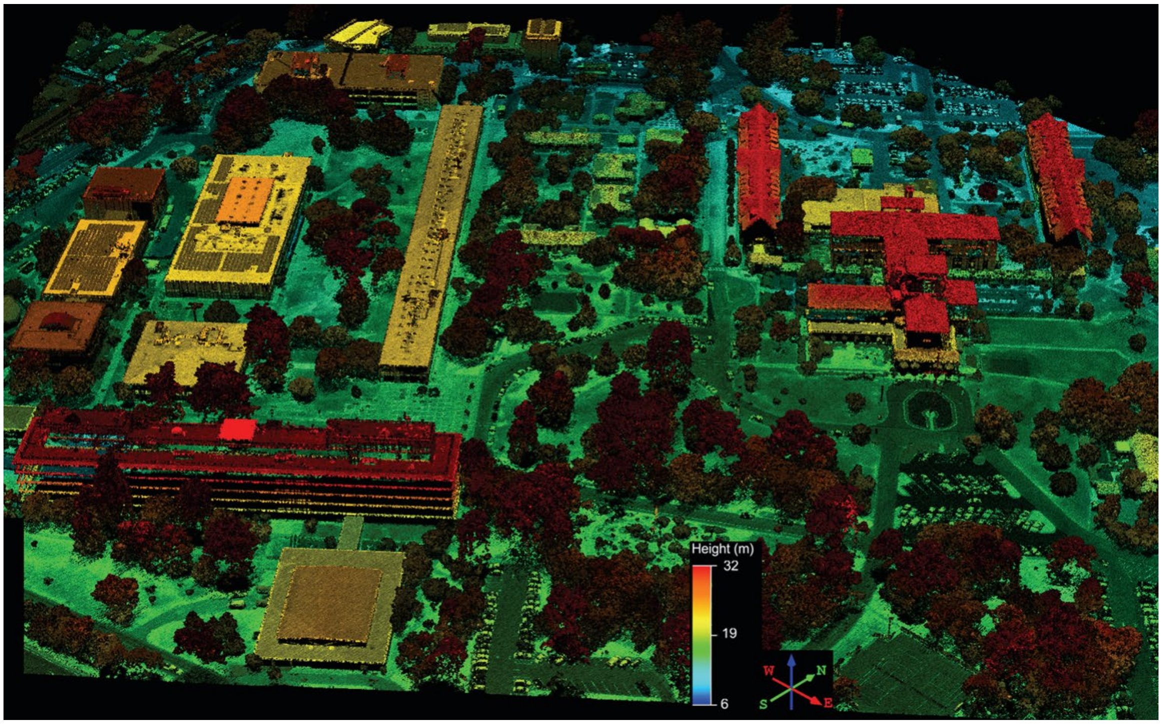


Figure 13. View of the Naval Postgraduate School campus obtained from an airborne LiDAR system. Source: [6].

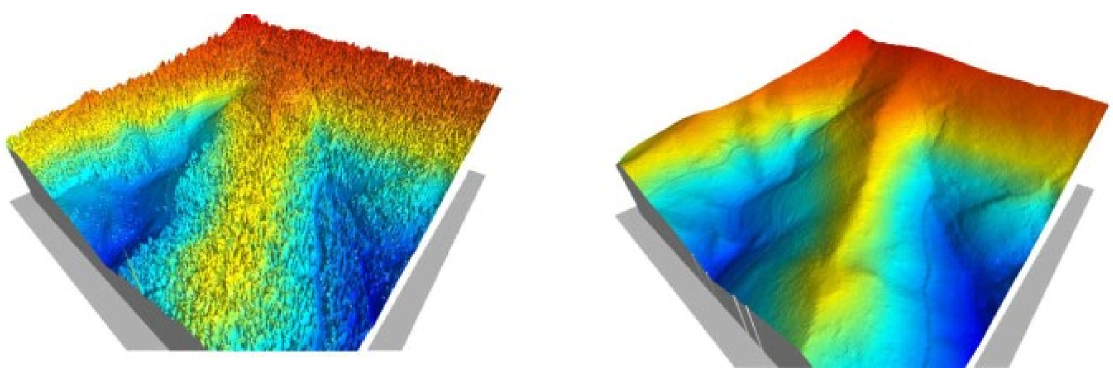


Figure 14. Example of Airborne Laser Terrain Mapping (ALTM) data showing vegetation removal. Source: [8].

The basic concept of the applications of the LiDAR systems is to radiate pulses of light, which are reflected by the nearest surface that these pulses of light encounter. The photo-detectors of the system capture the returning light, which is recognized by its unique frequency [9]. An illustration of this procedure is presented in Figure 15.

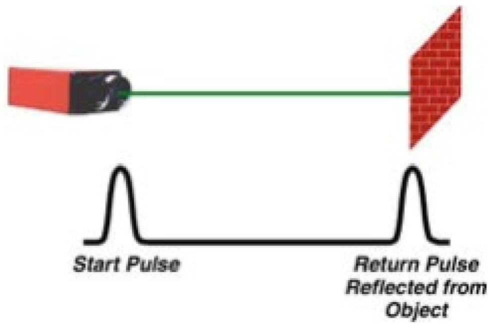


Figure 15. Simple LIDAR example, pulse return. Source: [10].

By computing the time that elapsed between the transmission and the reception, we acquire the distance between the system and the reflective surface [10], as is shown in Equation 2.1:

$$R = \frac{t_r - t_t}{2c} \tag{2.1}$$

One additional capability of LiDAR systems is their ability to identify the intensity of the reflected light [10]. The portion of the received laser light is the result of many parameters, as it is the distance from the target and the type of its surface (i.e., snow may reflect about 18 times more light than black asphalt) [10]. These properties are immensely useful in remote sensing from long distances where we may consider that the distance is almost the same for all the targets. Hence, by processing these LiDAR data, we obtain images of the environment which look very much like regular images from a standard camera.[10]. In Figure 16, we can see an image of the Niagara Falls, captured by exploiting the aforementioned properties.

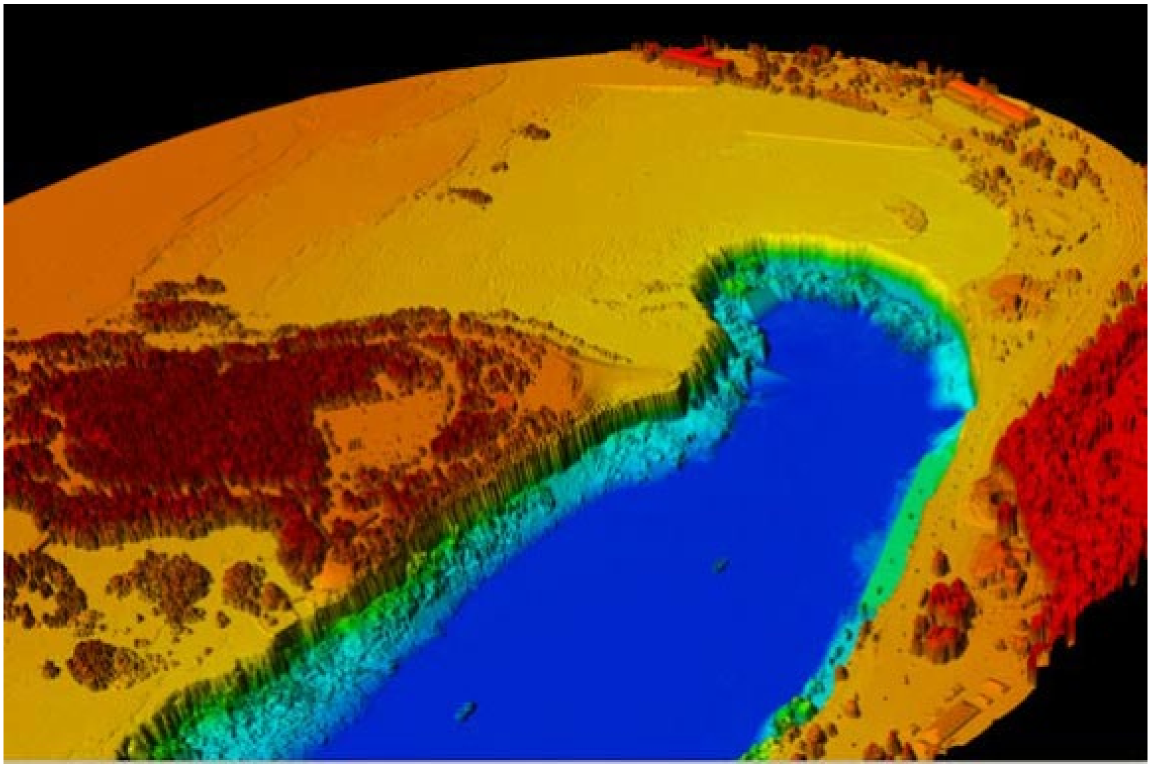


Figure 16. LiDAR image of Niagara Falls. Source: [10].

B. HARDWARE COMPONENTS OF THE 3D 360° LIDAR SENSOR

Nowadays there are many integrated systems for collecting LiDAR data. Those most suitable for our study are integrated LiDAR systems that have the ability to scan the surrounding environment and provide real-time LiDAR data with high resolution [11]. Also, it is preferable for these sensors to perform in the Infra-red (IR) spectrum in order to be compatible with the regulations for eye safety [11].

For these reasons, systems consisting of several LiDAR transmitters and detectors with a standard angle among them [11] are ideal for our case. Each pair of the transmitters and detectors forms a channel that operates at well-defined frequencies all distinct from each other. This array of channels is placed within a compact housing [11]. This array spins speedily within its fixed case and scans the surrounding environment by firing each laser tens of thousands of times per second [11]. In this way it provides, in real-time, a substantial set of 3D point data of the surrounding environment [11]. In Figure 17 is a depiction of how the array scans its surroundings, effectively creating a surveillance zone.

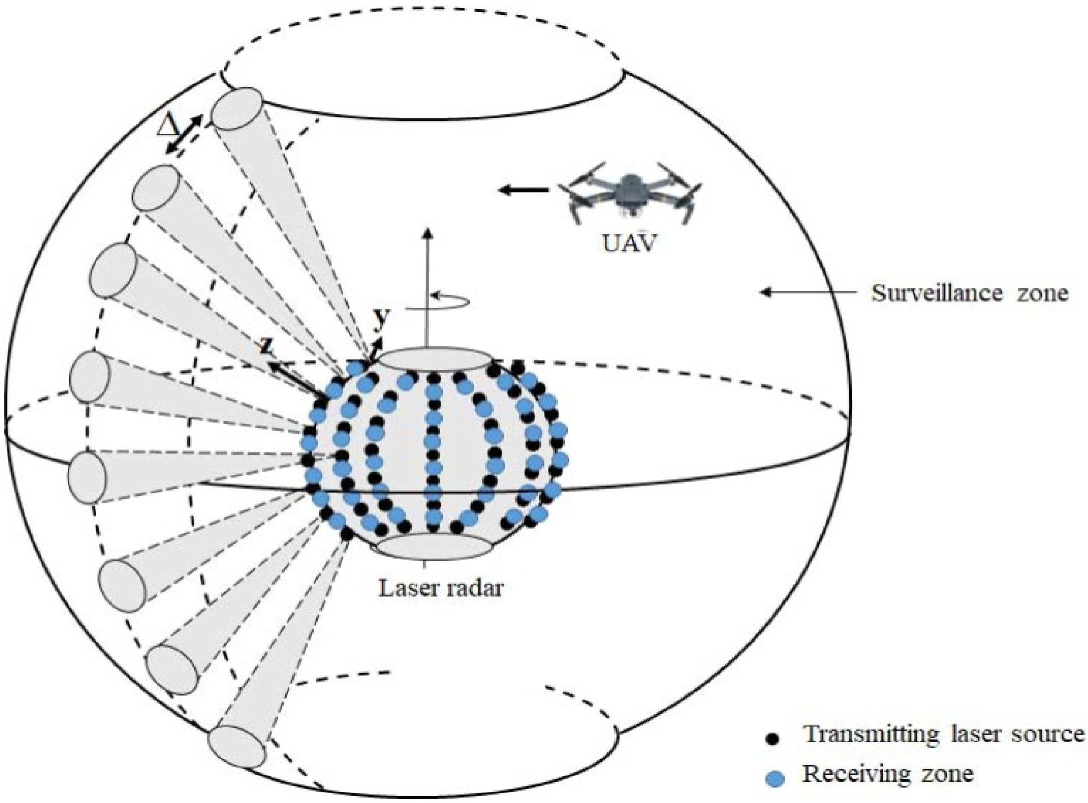


Figure 17. General overview of the proposed LiDAR system. Source: [5].

The crucial resolution feature of the sensor is analyzed in two directions, the Horizontal angular (Azimuth) resolution and the Vertical angular resolution [11]. The way the LiDAR samples the environment is by a rotating head that fires a fixed number of laser pulses per second (the "firing rate") [11]. As a consequence, the resulting Azimuth angular resolution is determined by the rotating speed of the head (in degrees/sec), and it can be computed as [11]:

$$Azimuth_{Resolution}(°) = speed_{rotation}(°/second) \times firing_{cycle}(seconds) \quad (2.2)$$

Consequently, if we increase the rotation speed of the head, the angle we can resolve also increases and vice versa [11]. Since the goal of detecting small moving objects requires both small angular resolution as well as fast tracking, the rotation rate of the LiDAR head chosen must be a compromise between these two conflicting requirements.

A different situation is in the vertical plane where there is a fixed number of channels all firing at the same time [11]. Therefore, the vertical angular resolution is determined by the total field of view (FOV) and the number of firing channels as:

$$Vertical_{Resolution}(°) = \text{FOV}(°) \,/\, number_{of\ vertical\ firing\ channels} \qquad (2.3)$$

In Figure 18, we can see a visualization of the point density in one frame and in successive frames. The difference between the vertical and the horizontal resolution is obvious, as is how much the deviation increases during the evolution of scanning.

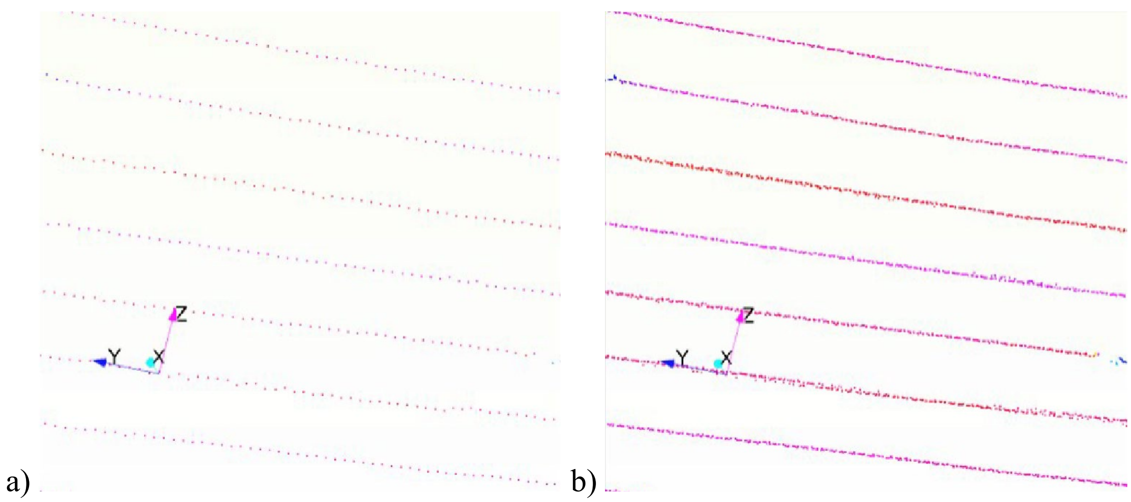


Figure 18. Point density in one frame (a) and in series of successive frames (b). Adapted from [11] (colors inverted).

In these kinds of sensors, the data being collected reports the distances from the sensor in spherical coordinates (radius r, elevation ω, azimuth α), with the origin (0,0,0) defined at the LiDAR sensor [11]. In order to convert this spherical data to Cartesian coordinates (X, Y, Z), we need to apply the following formulas [11]:

$$X = R\ \cos(\omega)\ \sin(\alpha)$$

$$Y = R\ \cos(\omega)\ cos\ (\alpha)$$

$$Z = R \sin(\omega)$$

Figure 19 is a graphic representation of the equations.

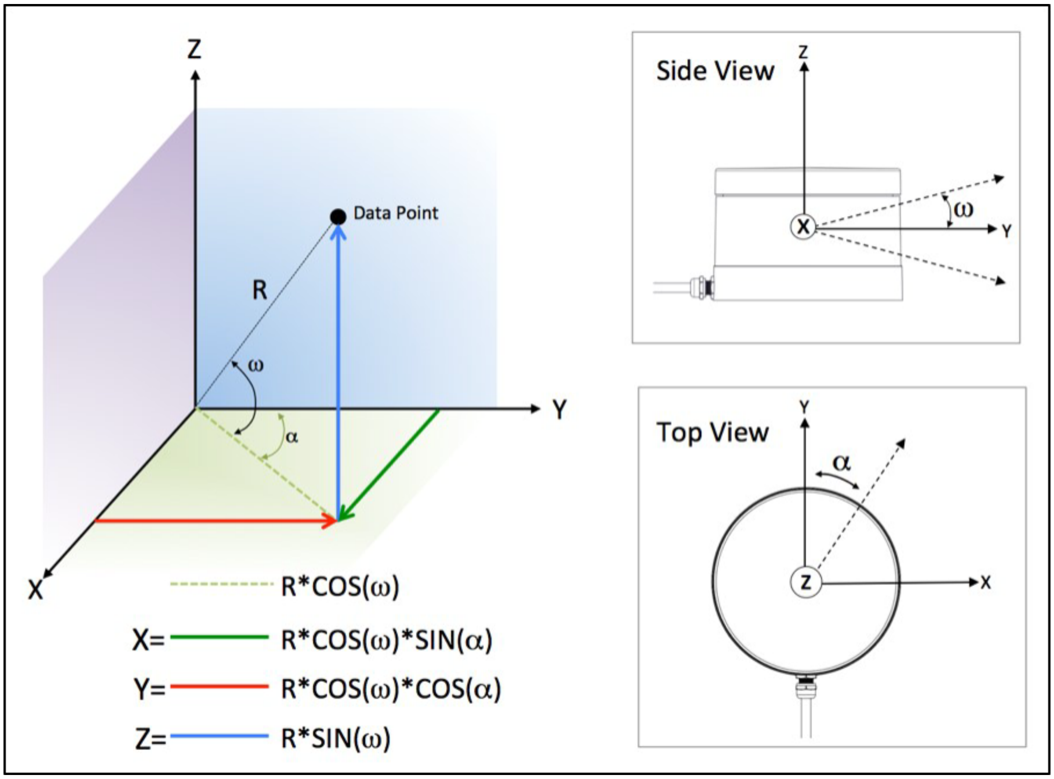


Figure 19. Sensors coordinate system. Source: [11].

Although we have already pointed out that each beam along the distance retains its narrow width, in reality there is a beam divergence, meaning that a laser beam slowly, gradually grows larger after leaving the sensor [11]. Hence the width of the beam could be large enough to be reflected by multiple objects. This effect can be exploited by the sensors in order to acquire the desired data; it can be accomplished by adjusting which reflection we want to capture (i.e., the strongest or the last, or both of them) [11]. Figure 20 shows a possible scenario where there are two different reflected portions of the same beam, while in Figure 21 we can see the case that there are multiple reflected portions of the same beam.

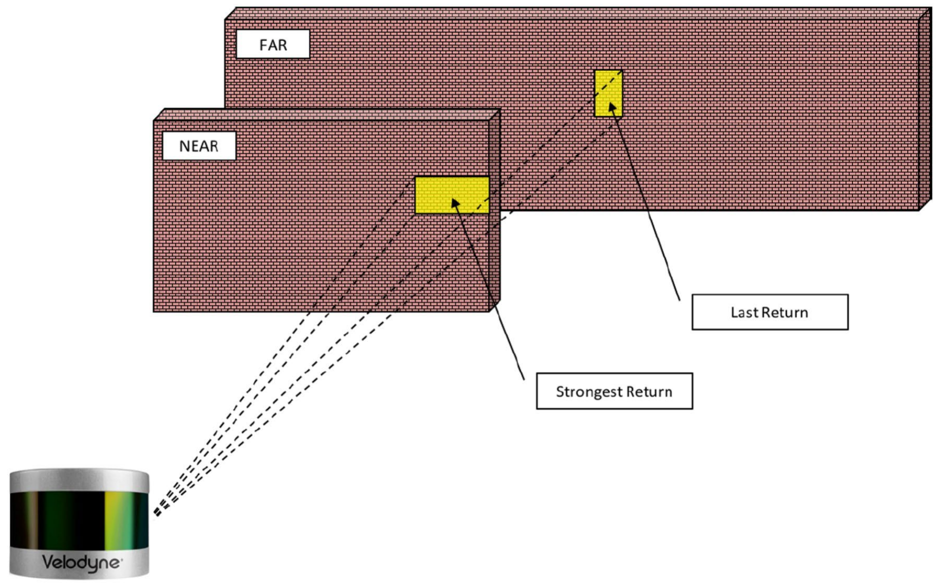


Figure 20. Dual Return example (last and strongest reflections). Source: [11].

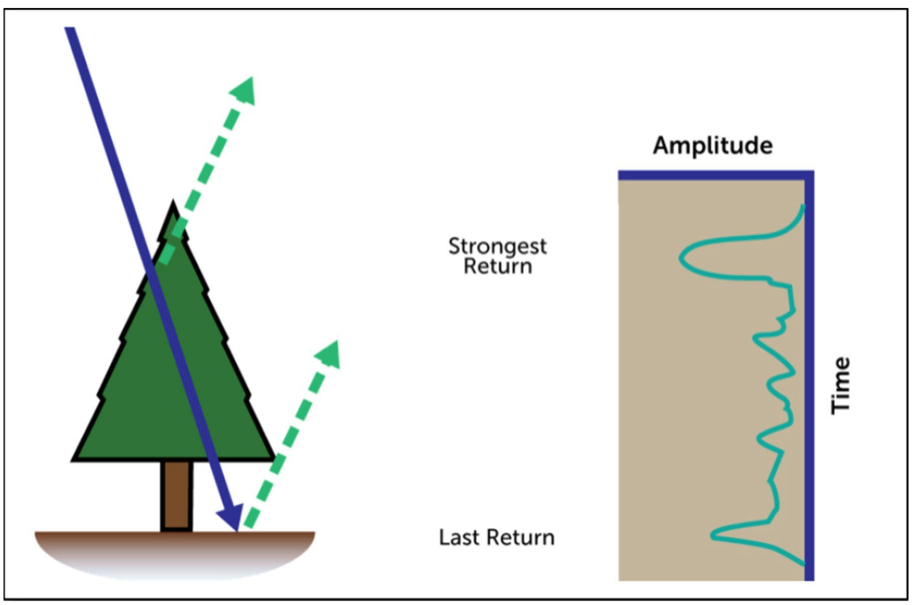
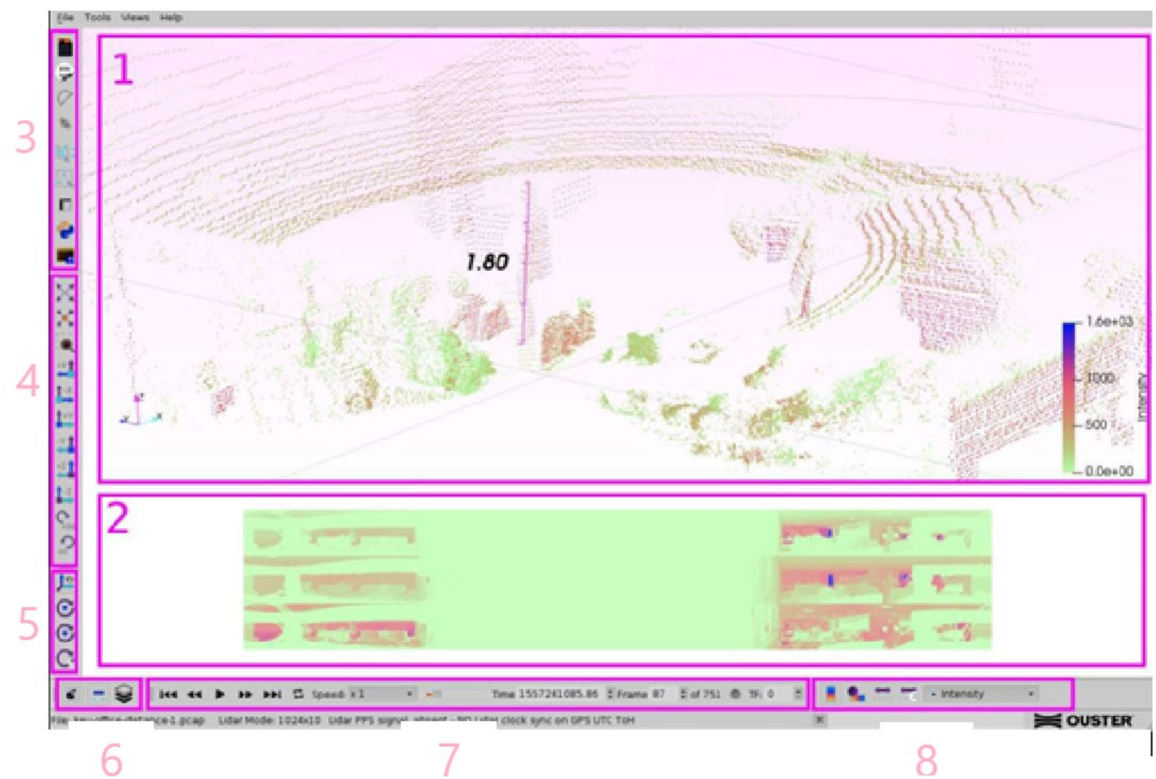


Figure 21. Forestry application with multiple returns. Source: [11].

C. SOFTWARE FOR PROCESSING LIDAR DATA

Currently, many software applications are available to capture, visualize, and process LiDAR data. These applications can be separated into two general groups. One group consists of the applications designed to capture and visualize the LiDAR data. Such types of software are provided mainly by the manufacturers of LiDAR sensors [12]. A second group could be considered the software that presents capabilities of advanced processing of the LiDAR data [13].

The software provided by the manufacturer usually is designed specifically for their sensors and cannot be applied to other sensors. Generally, this software is capable of performing real-time visualization, processing, and recording of the data that are being captured from the LiDAR sensors.[12]. In particular, it can render either live streaming data or stored data as long as they are recorded in an appropriate format. Some common view formats are the “3D view,” “2D view,” and “Spreadsheet view” [12]. Some typical examples of these choices of views are illustrated in Figures 22 and 23, respectively.



1) 3D View of point cloud data, 2) 2D 360° image view, 3) Basic control toolbar, 4 & 5) View toolbars, 6) Measurement and projection toolbar, 7) Player control toolbar, 8) Colormap toolbar.

Figure 22. Overview of Ouster Studio’s graphical interface. Adapted from [12] (colors inverted).

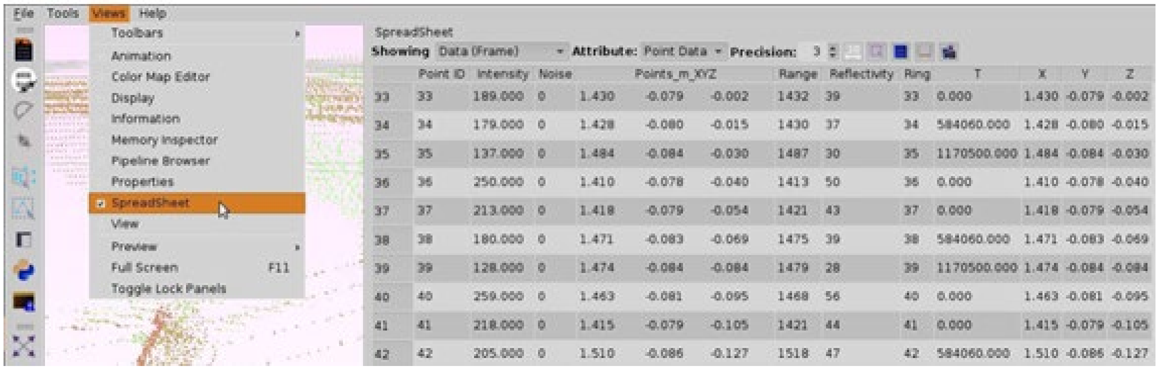
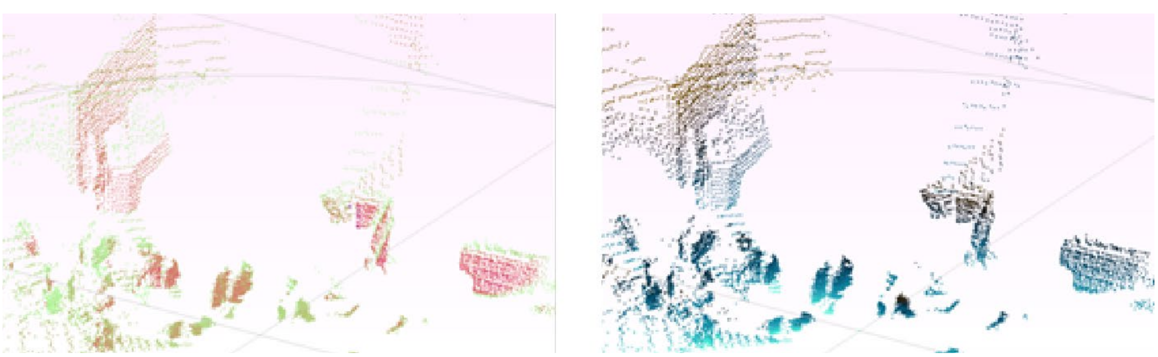


Figure 23. Spreadsheet view. Adapted from [12] (colors inverted).

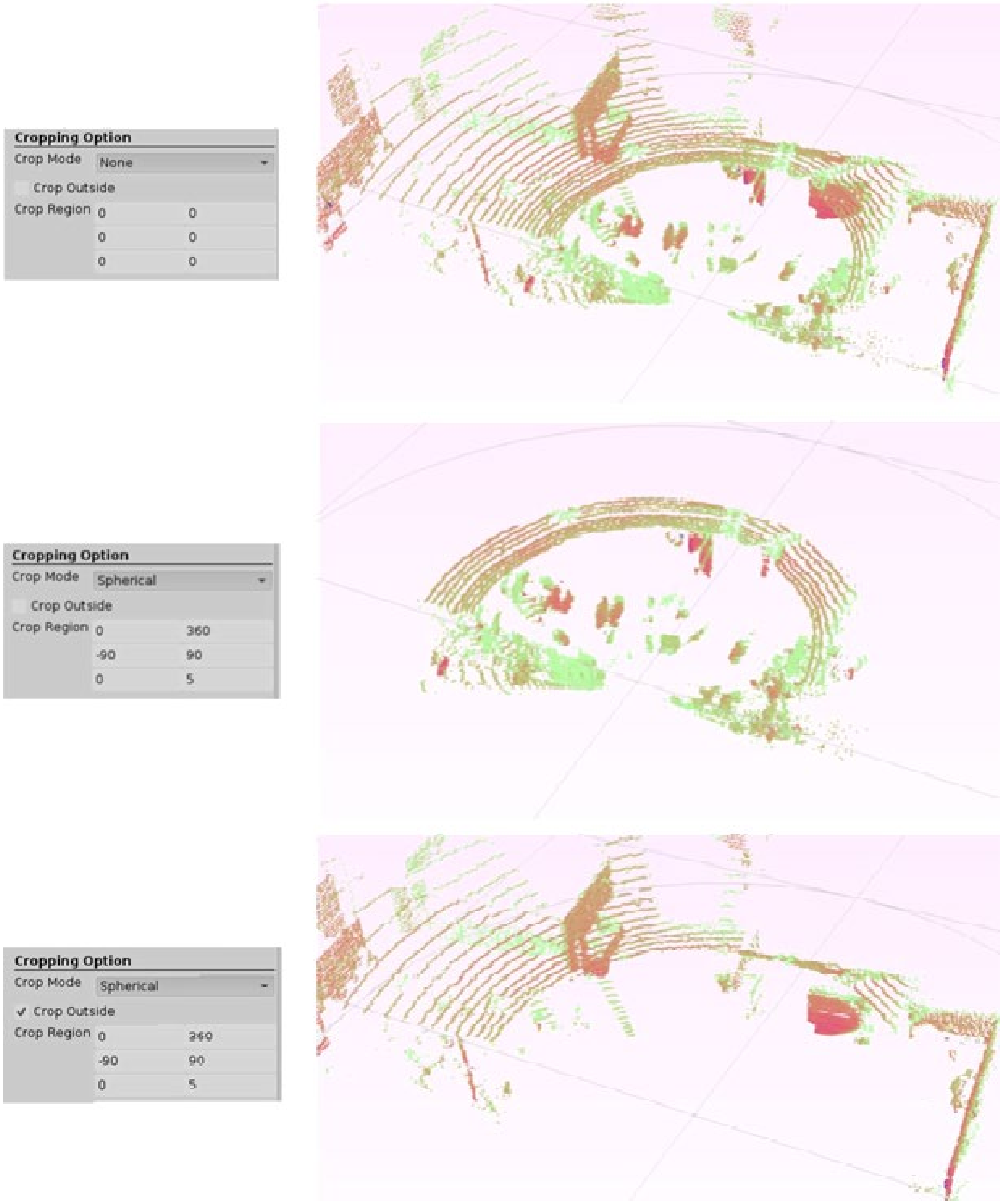
To customize these view formats, capabilities are provided for adjusting the image features, such as changing the colors based on the points characteristics (intensity, distance, etc.) or filtering the desired data [12]. The recording of the captured data produces files containing LiDAR data. A typical format for this kind of data files is the .pcap format [12]. The pcap (packet captures) is an application programming interface (API) that has the ability to provide information comprehensively from a large amount of data, like the traffic of networks [14]. Furthermore, with this software it is possible to perform basic processing of the sensor data, like cropping part of the point cloud and keeping only the rest of the data [12].

Figure 24 presents some implementations of adjusting the colors based on the points characteristics. Additionally, Figure 25 shows the results of a cropping operation.



a) Coloring by Intensity b) Coloring by Ring

Figure 24. Images produced by the same point cloud, differentiated by the attributes used for coloring. Adapted from [12] (colors inverted).



Top image: no cropping. Middle image: cropped to a 10-meter diameter radius sphere. Bottom image: cropped "outside" a 10-meter diameter radius sphere.

Figure 25. Behavior of the cropping in "Spherical" Mode. Adapted from [12] (colors inverted).

At this point, we should clarify the form of the LiDAR data that we acquire from the related sensors. These data are divided into frames where each frame corresponds to a complete rotation of the array of sensors [15]. Generally, the LiDAR sensors provide "point cloud" data, which are sets of data points in 3D space. Each of these points describes a location on a real-world object's surface in Cartesian coordinates (X, Y, and Z), and the total set of these points map the entire surface of the surrounding objects [16]. The types of the point clouds may be grouped into two categories. One type is the "Organized" point clouds that have the format M x N x C (where M = number of rows, N = number of columns, and C = number of channels), and the "Unorganized" point clouds that have the format M x C (M = number of points and C = number of channels) [16].

On the other hand, the software that provides capabilities for extended and advanced processing of the LiDAR data are programming and computing platforms like MATLAB [17], [13] and standalone, and large scale libraries for 2D/3D image and point cloud processing, like Point Cloud Library (PCL) [18] [19]. In this study, we make use mainly of MATLAB. MATLAB provides a powerful toolbox that consists of a plethora of algorithms, functions, and applications for designing, analyzing, and testing LiDAR data [20]. In addition, there are many examples that use these tools for processing the LiDAR data, which makes this toolbox especially handy to use. Some crucial capabilities among the others that could be used for the aim of this study are the capability of "segmentation" [21] and the capability of processing the LiDAR data as a stream of frames, as in the case of a video file [22].

Segmentation associates each point in a frame of a 3D point cloud to a cluster of points that is described by a class label [21]. There are different methods for applying data clustering, but the most suitable for our case is by evaluating the distance between two neighboring points and classifying them into the same cluster only if their distance is below a specified threshold [21]. The result of clustering is the classification of each point of a frame into a cluster, and each cluster is a probable object. Hence, in that way, we can check whether each cluster (probable object) qualifies as a potential desired detection target. Figure 26 presents the clusters of a point cloud that are distinguished by their different colors, while Figure 27 displays colored clusters using advanced processing for the classification.

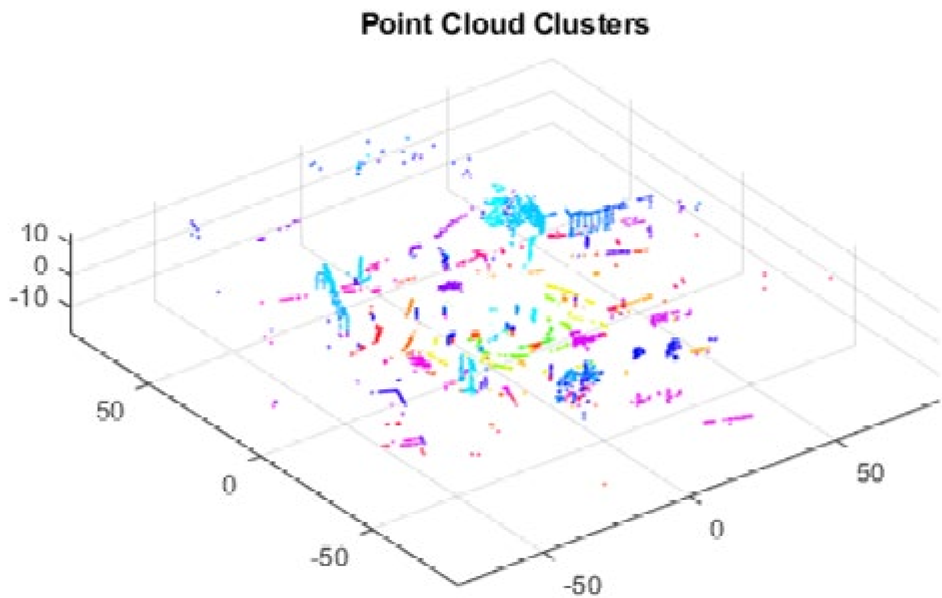
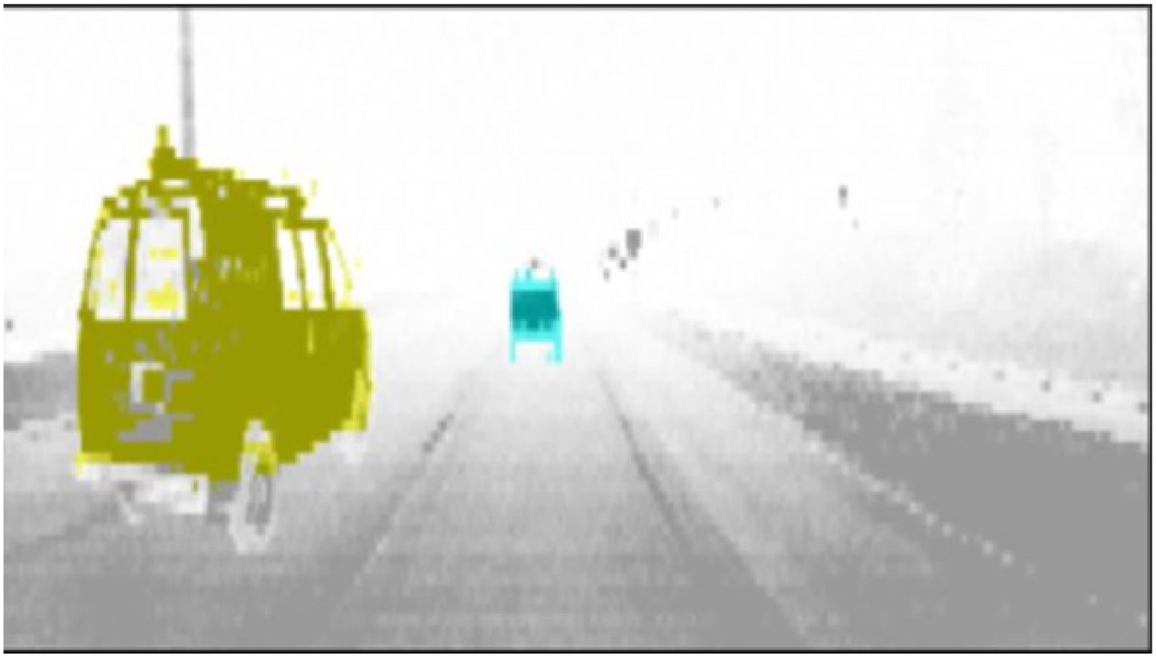


Figure 26. Point cloud clusters (distinguished by different colors). Adapted from [21] (colors inverted).



The car is shown in blue green, the truck is shown in yellow, while the background appears grayscale.

Figure 27. Semantic segmentation of point clouds. Adapted from [16] (colors inverted).

Immediately related to the previous procedure is the processing of the LiDAR data as a stream of frames. Specifically, by comparing the attributes of clusters detected in successive frames, we can characterize the nature of the corresponding targets. They may be large or small objects, moving or steady objects, new entries in the scene, and many more attributes could be extracted by this procedure. These attributes are used in this study in order to detect the sUAVs.

THIS PAGE INTENTIONALLY LEFT BLANK

III. DATA COLLECTION

This chapter describes the experiments that were conducted to collect the LiDAR data. First, the test setup is described. Next, the methodology for the test is presented. The procedure for the data collection is then described and, finally, the analysis of the raw data is presented.

A. TEST SETUP

The test setup was performed by using the appropriate equipment in challenging environments. In particular, the equipment was composed of LiDAR sensors, sUAVs, and auxiliary equipment that performed in rural environments.

The LiDAR sensor used in the experiments for this research is the "Velodyne Puck Hi-Res" LiDAR sensor [23]. This sensor corresponds to the description of the LiDAR sensors of Chapter II. It consists of 16 channels (pairs of transmitters and detectors) with a measurement range of about 100 meters [23], [24]. The horizontal angular resolution is between 0.1° and 0.4°, whereas the vertical angular resolution is 1.33° [23], [24]. Moreover, the horizontal FOV is 360°, while the vertical FOV ranges between $+10^{\circ}$ and -10° (20°) [23], [24]. Figure 28 shows the aforementioned sensor, while Table 1 presents its basic specifications.

Figure 28. The Velodyne Puck Hi-Res LiDAR sensor. Source: [25].

Table 1. Specifications of the Velodyne Puck Hi-Res LiDAR sensor. Source: [24].

	Specifications:
Sensor:	• 16 Channels • Measurement Range: 100 m • Range Accuracy: Up to ±3 cm (Typical)[1] • Field of View (Vertical): +10.0° to -10.0° (20°) • Angular Resolution (Vertical): 1.33° • Field of View (Horizontal): 360° • Angular Resolution (Horizontal/Azimuth): 0.1° – 0.4° • Rotation Rate: 5 Hz – 20 Hz • Integrated Web Server for Easy Monitoring and Configuration
Laser:	• Laser Product Classification: Class 1 Eye-safe per IEC 60825-1:2007 & 2014 • Wavelength: 903 nm
Mechanical/ Electrical/ Operational	• Power Consumption: 8 W (Typical)[2] • Operating Voltage: 9 V – 18 V (with Interface Box and Regulated Power Supply) • Weight: ~830 g (without Cabling and Interface Box) • Dimensions: See diagram on previous page • Environmental Protection: IP67 • Operating Temperature: -10°C to +60°C[3] • Storage Temperature: -40°C to +105°C
Output:	• 3D Lidar Data Points Generated: - Single Return Mode: ~300,000 points per second - Dual Return Mode: ~600,000 points per second • 100 Mbps Ethernet Connection • UDP Packets Contain: - Time of Flight Distance Measurement - Calibrated Reflectivity Measurement - Rotation Angles - Synchronized Time Stamps (μs resolution) • GPS: $GPRMC and $GPGGA NMEA Sentences from GPS Receiver (GPS not included)

Several types of sUAVs have been used during these experiments. During analysis, these types were separated by size according to those with a maximum dimension larger than 60 cm and ones with a maximum dimension less than 60 cm. Also, the sUAVs that operated in the test field were very diverse in shape and size. Nevertheless, this variation fits the scope of the research since it was not intended to focus on a specific model of sUAV. Figure 29 shows the sUAVs with a maximum dimension of less than 60 cm, and Figure 30 shows the sUAVs with a maximum dimension of more than 60 cm that were used in the experiments.

Figure 29. sUAVs with a maximum dimension of less than 60 cm used in the experiments.

Figure 30. sUAVs with a maximum dimension of more than 60 cm used in the experiments.

In addition to the main equipment just described, auxiliary equipment was also used. For example, the usual uninterruptible power sources (UPS) were used for the necessary power supply of the sensors. In addition, a common laptop was used, where the software "VeloView" was installed. Velodyne provides VeloView, which is capable of analysis, visualization, and recording of LiDAR sensor data [26]. Figure 31 shows the LiDAR sensor and some of the auxiliary equipment settled in the test field.

Figure 31. LiDAR sensor and some of the auxiliary equipment set up in the test field.

The environment in which the experiments were conducted was quite challenging for the detection procedure. Two airfields were used for the experiments: the Monterey Bay Academy Airfield (MBA) in WatsonvSille, California, and the NPS Test Site in Marina, California. Both fields are in rural areas, characterized by uneven ground and the presence of small plants, bushes, and trees. Figures 32 and 33 present the environment at the NPS Test Site at Marina.

Figure 32. The environment at the NPS Test Site at Marina.

Figure 33. The environment at the NPS Test Site at Marina.

B. METHODOLOGY FOR EVALUATION OF SUAS DETECTION BY LIDAR SENSOR

The methodology that was applied to this research pertains to identifying the exact number of the points that each detected sUAV covers in each frame in regard to the sUAV basic attributes (e.g., maximum dimension) and its motion (e.g., speed and distance from the sensor). The number of pixels detected that are assigned to each sUAV is also used as a measure of the reliability of the estimate itself.

Specifically, the LiDAR sensor captured the motion of various sUAVs flying random routes. The data that were produced by the sensor were stored and then processed through MATLAB software by applying an algorithm for detection of sUAVs. This algorithm provided the necessary information about the detected sUAVs. This information consisted of the attributes of the detected sUAVs, including the Cartesian coordinates (X, Y, Z) of the detected objects, where the origin (0,0,0) was set at the sensor location. Additionally, it included the ID of the frame in which the object was detected, the exact time that it happened, the distance of the object from the sensor, as well as the number of the points of the object.

Consequently, the desired results were derived from the preceding information: the number of the points for each detected sUAV regarding its distance from the sensor, its altitude, direction, and velocity of flight. Also, since the information provided the exact position of the sUAV in relation to the sensor, it was simple to determine the texture of its background (textured, smooth, etc.) and therefore to figure out whether the background correlates to the efficiency of the LiDAR sensor in detecting the sUAVs.

By assuming a smooth trajectory and averaging the features of different frames, we acquired the percentage of sUAV detection in relation to their motion attributes. Specifically, in several cases, between two successive detections of an sUAV, there were frames in which the targets were not detected because of the occlusion caused by obstacles like trees, or because the sUAV was located in the gap that existed between two neighboring points of a frame. Hence, in these cases we assumed that the motion between these two positions was smooth and straightforward with constant velocity (zero acceleration).

The algorithm that we applied to the sUAV detection by LiDAR is the result of the combination of various techniques. One of them was the principal component analysis (PCA), which is a method based on linear algebra used for many applications, such as face recognition [27], [28]. Through this process we try to find the optimal projection of the data vectors and transform the data on a different basis [27], [28]. This results in a set of sorted uncorrelated data with reduced dimensionality [27], [28]. In this research, the PCA algorithm was used to facilitate the comparison of data between different frames with the added advantage of more efficient computation.

Figure 34 shows a flow chart of the methodology just described that was applied in this research.

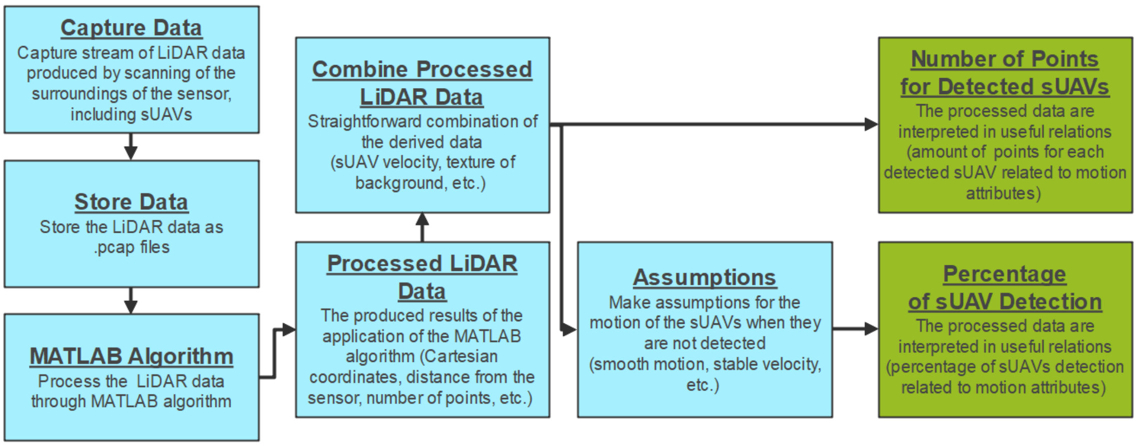


Figure 34. Flow chart of the methodology applied in this research.

C. DATA COLLECTION PROCEDURE

The Velodyne Puck Hi-Res LiDAR sensor collected the necessary data at the MBA and at the Test Site at Marina. After the required equipment was set up at the test field, the LiDAR sensor was activated and scanned the surrounding environment, while the data produced were stored in the connected laptop. In addition, the sUAVs used for testing took off and flew in random routes, in the active range of the LiDAR sensors (100 m).

Although the flight routes of the sUAVs were random, they included all the possible situations. There were flights with upwards – downwards directions; there were also flights where the sUAVs approached the sensor, moved away, and moved while keeping a stable distance from the sensor. All these directions were applied both against a textured background (plants, trees, etc.) and against a smooth background (clear sky). Additionally, in all the aforementioned cases a combination of velocities was applied, including static motion, high speeds, and low speeds.

Figure 35 presents images of the sUAVs with a maximum dimension of more than 60 cm during their flights, while Figure 36 shows images of the sUAVs with a maximum dimension of less than 60 cm during their flights.

Figure 35. In-flight images of sUAVs with a maximum dimension of more than 60 cm.

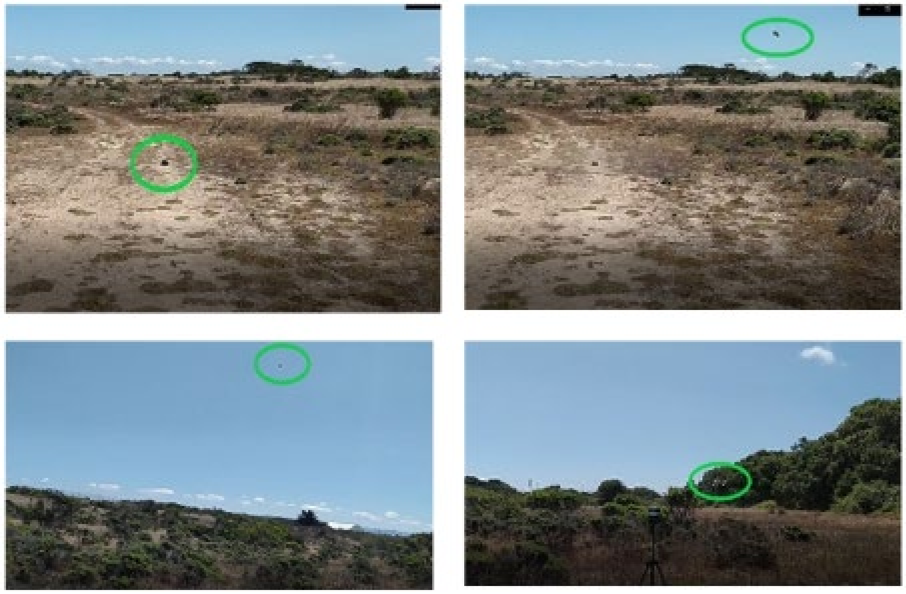


Figure 36. In-flight images of sUAVs with a maximum dimension of less than 60 cm.

D. ANALYSIS OF THE RAW DATA

The Velodyne Puck Hi-Res LiDAR sensor reports the distances from the sensor in spherical coordinates (radius r, elevation ω, azimuth α), where the origin (sensor) is declared as (0,0,0) [11]. The spherical data are converted to Cartesian coordinates (X, Y, Z) by applying simple formulas [11]. These coordinates, with other data like the timestamp, the sensor model, and the laser return mode constitute the first type of packet that this sensor generates and is called data packet [11]. The second type of packet is called position packet and provides data related to synchronization (e.g., with GPS time source) [11].

The data packets that the sensor produces consist of a large number of bytes [11]. A single data packet contains the data of 24 firing sequences and its length is 1,248 bytes [11]. Moreover, there are two possible formats of these packets, the single return mode format and the dual return mode format [11]. Figure 37 shows the typical structure of the single return format, while Figures 38 and 39 present the same format, with examples of the start and the ending of a data packet, respectively.

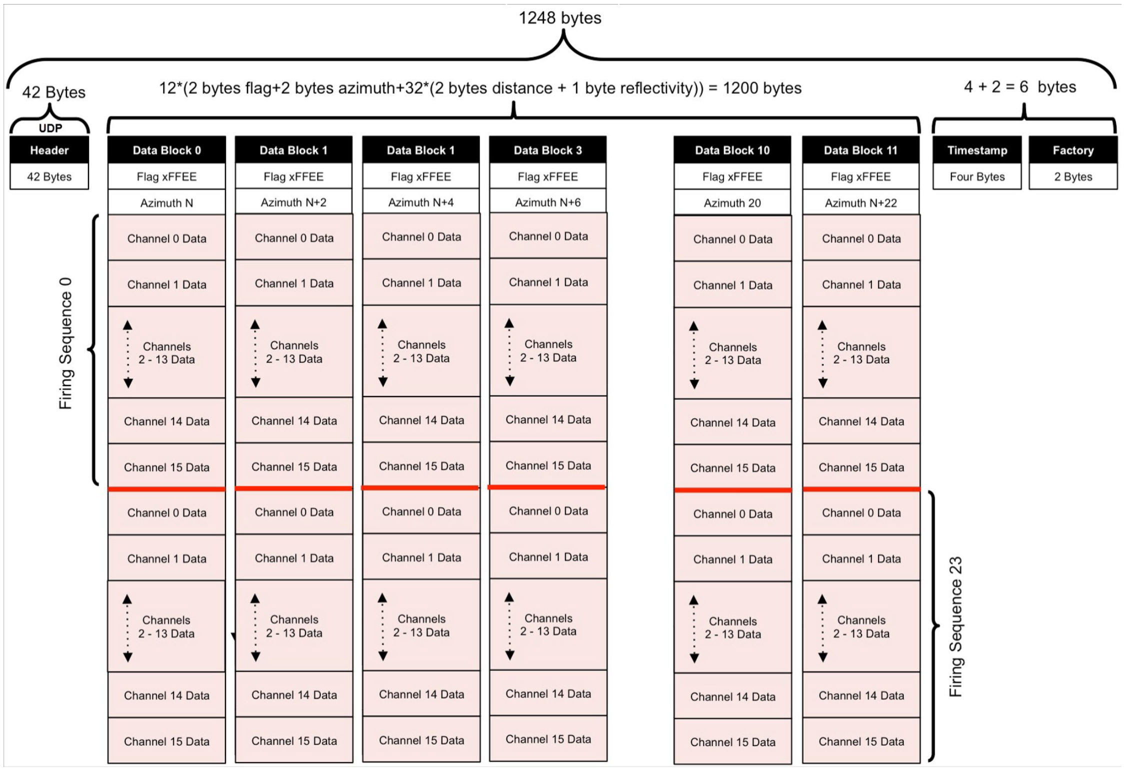


Figure 37. Structure of the single return mode data packet. Source: [11].

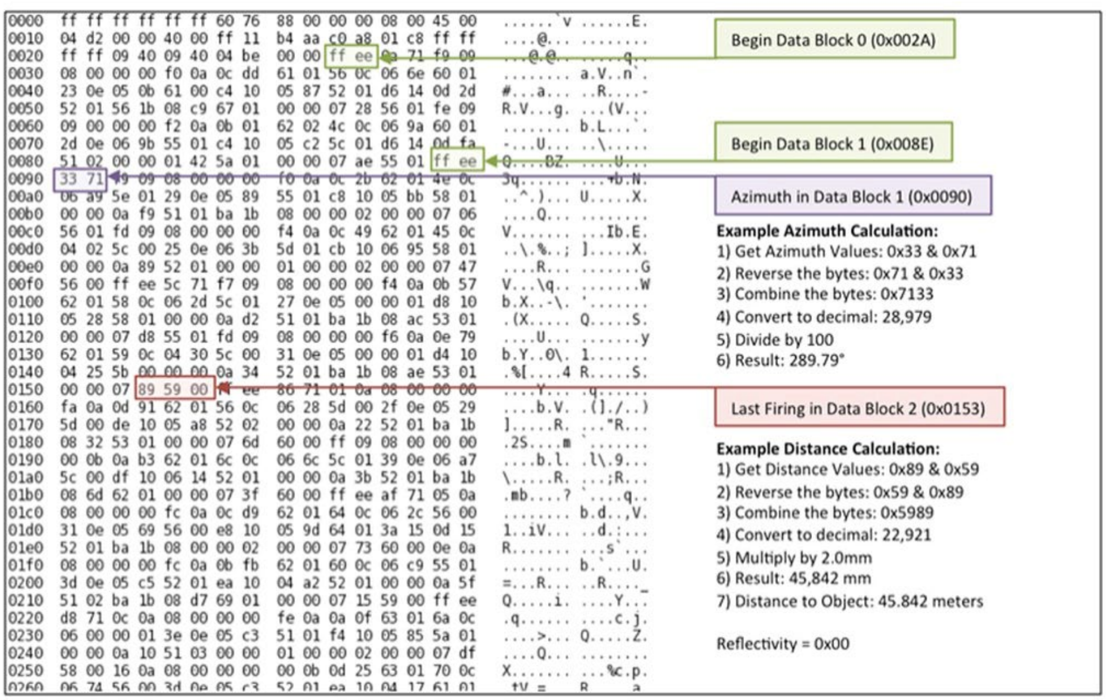


Figure 38. Example of the start of a single return mode data packet. Source: [11].

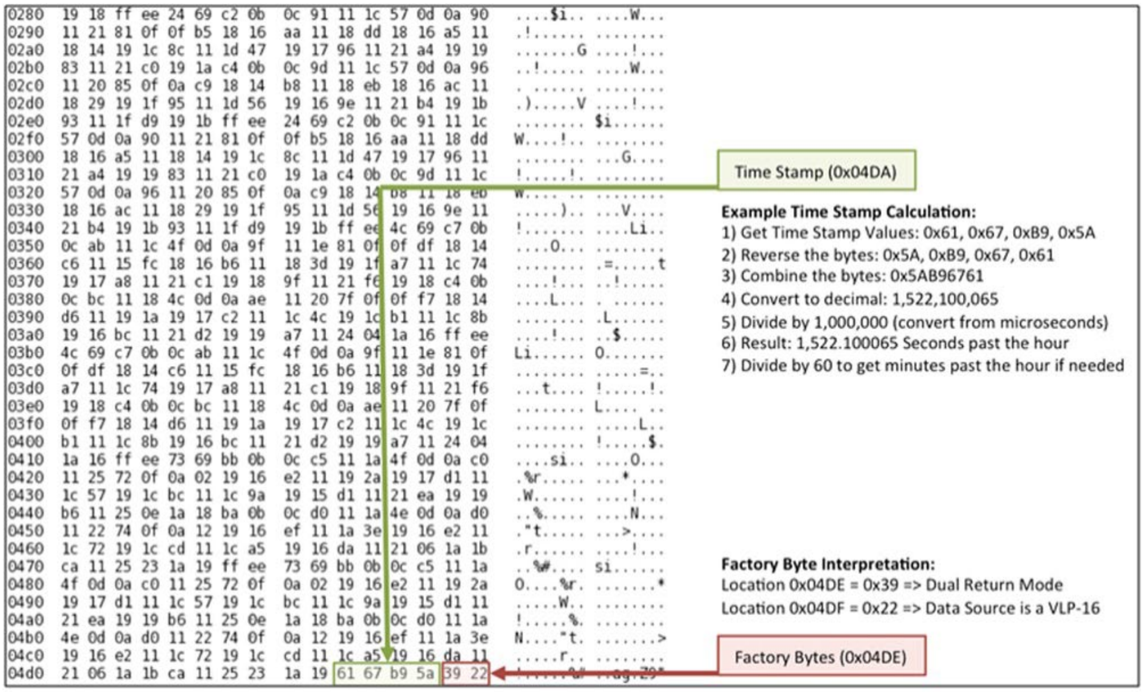


Figure 39. Example of the ending of a single return mode data packet. Source: [11].

All the data that were produced from the LiDAR sensor are stored by the VeloView software as .pcap files [11], [26]. To process the .pcap files, they should be converted to a file format called "point cloud file" [11]. This conversion is a quite challenging process [11]. Thankfully, MATLAB provides the "velodyneFileReader" object that can read point cloud data immediately, without further interventions, from .pcap files that have been captured by a Velodyne LiDAR sensor [29].

THIS PAGE INTENTIONALLY LEFT BLANK

IV. DEVELOPMENT OF THE SUAS DETECTION ALGORITHM

This chapter presents the development of a sUAS detection algorithm. The chapter starts by describing the algorithm that was used in this research. Next the computer simulations of the application of the algorithm are presented. Finally, after an explanation of how the flight test data was processed, the chapter closes with an evaluation of the comparisons of the processed data.

A. KEY FEATURES OF THE DEVELOPED ALGORITHM

MATLAB was the programming platform used for the development of the algorithm for sUAS detection. This algorithm took advantage of the powerful capabilities of this platform for processing massive amounts of data in real-time, and the capability of expressing matrix and array mathematics directly [13].

First, the algorithm reads the LiDAR data that were stored as .pcap files and interprets them as point cloud objects [29]. In this way it was quite convenient to extract and process, frame by frame, the data for each of the frames, such as the time and the Cartesian coordinates of each point of the frame.

Furthermore, a basic function of the algorithm was the segmentation of the detected points into clusters based on their 3-D range [29]. By using the distance between neighboring points as the deciding criterion, it was possible to segment the points into clusters. These clusters could be considered objects that could then be classified as solid (such as a car) or non-solid (such as the leaves of a tree). One of the main challenges of this procedure was the large number of clusters for each frame. In this algorithm, the average number of the clusters identified in each frame was about 400. It would be possible to significantly decrease this number by increasing the minimum number of points that could form a cluster. But, since the targets of interest, sUAVs, yield small clusters, we need to decrease the number of clusters without eliminating small targets.

After the segmentation was complete, the algorithm created a list of the aforementioned clusters. This list was enriched with the attributes that corresponded to each cluster. Such attributes were the median Cartesian coordinates (X, Y, Z), the maximum dimension in each axis (X max, Y max, Z max), the distance from the sensor, and the position of the central point in the frame (row and column). The list was updated after the segmentation of each frame. If an object (cluster) was shown for the first time, then it was added as a new registration, but if it was already detected in previous frames the algorithm just updated the attributes of the existing registration. An important characteristic of the list of clusters was that each cluster kept the same order in the list.

The effectiveness of the algorithm depended on its capability in identifying the same objects in successive frames. This procedure was performed by comparing the clusters of each frame with the clusters of the list. The comparison was realized by applying the method of principal component analysis (PCA). Specifically, the PCA was applied to the features of each cluster that was registered in the list of the clusters. Furthermore, we checked the identification of the clusters by comparing the summation of their features.

The large number of clusters from each frame complicated the comparison procedure and degraded its effectiveness. Hence, the algorithm decreased this number by excluding some of them. First, it excluded the big objects, meaning the clusters that had dimensions larger than a threshold. Also, it excluded the ground.

Moreover, the main way that the algorithm decreased the number of clusters was by applying a mask. In particular, this mask was inspired from techniques that perform foreground detection by extracting the background in videos [30]. In the first frames, assuming they had no sUAV, the clusters detected were classified as "background." This background was optimized by filling the gaps between the points of the same clusters and adding a margin between the background and the sensor. Hence, we considered this background as a mask for any object behind it, and that object was then excluded from the process. This mask was quite effective, since it significantly reduced the number of the clusters, and more importantly, it excluded most of the clusters that were due to plants, leaves, and branches of trees.

After the processes just described, the algorithm searched for objects that met the specifications for sUAVs. Each of the previously mentioned objects was characterized as a candidate target if it moved beyond a distance threshold (indicating that it was a moving object); it was not near other objects (indicating that it was a flying object); and it was above an altitude threshold (indicating that it was a flying object). If a cluster was characterized as a candidate target multiple times (that is, above a threshold), then it was characterized as a detected target.

Additionally, the algorithm provided the capability of visualization from the LiDAR data. Actually, various options for data visualization were provided. One option was the visualization of the unprocessed LiDAR data. Another option was the visualization of only detected objects due to the restrictions of the algorithm. Further modifications could be applied to the visualization method, like the limitation of the projected frame and the addition of labels to the projected objects.

Also, the algorithm provided the capability of extracting information from the detected targets in tables. This information pertained to the features of each cluster related to the ID of frame. The algorithm presented this information concentrated in groups and sorted in a way that the features were obvious for each detected sUAV in each frame.

B. COMPUTER SIMULATIONS

The application of the algorithm produced both visualized results as well as printed ones.

The visualization of the detected targets was realized by plotting the 3D point cloud. As referred to in the previous section, the algorithm provided various options for visualization of the results, and in this section, we present some of these options. However, because the plots of LiDAR data are quite scarce and the sUAVs cover a small number of points due to their small size, to facilitate the presentation of the algorithm functionality we have made some assumptions. In particular, we assume that the desired targets for detection were generally the moving objects, instead of just sUAVs.

Figure 40 presents the visualization of the LiDAR data before the application of the algorithm, and Figure 41 shows the visualization after the algorithm is applied in the same frame. The algorithm isolated all the moving objects, which in this case are humans. Also, it is obvious that for each active detected target (moving object) there is a label over it with the ID of the cluster to which it corresponds. Furthermore, the algorithm prints labels that show the ID of the frame, the total number of detected targets, and the number of detected objects that are active in this frame.

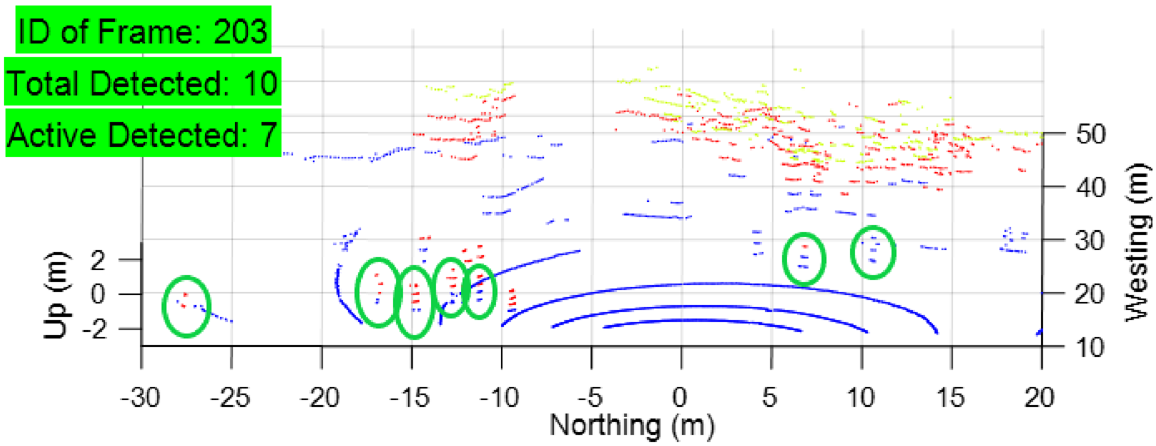


Figure 40. Visualization of the LiDAR data **before** the application of the algorithm for frame with ID 203.

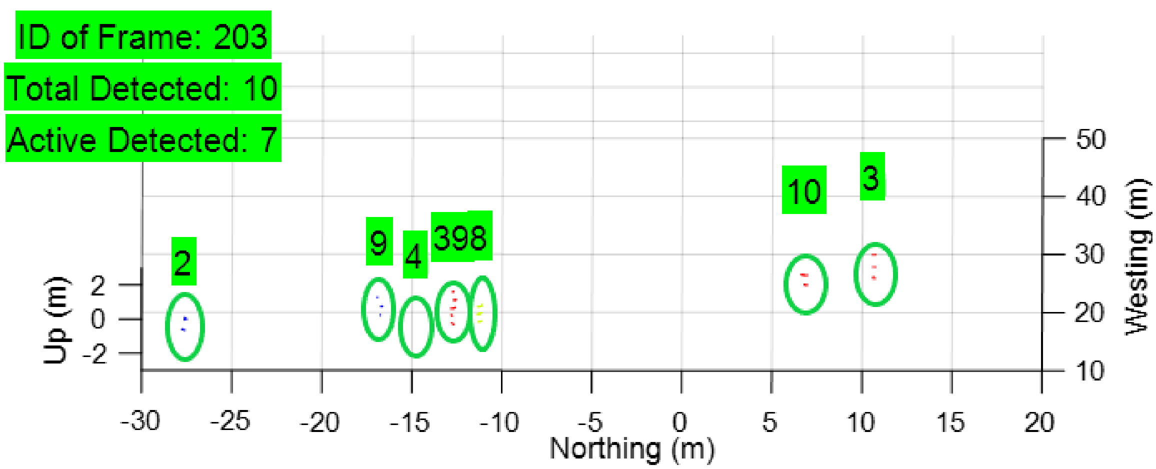


Figure 41. Visualization of the LiDAR data **after** the application of the algorithm for frame with ID 203.

Another option for extracting results by the algorithm is to print them. The printed information consists of the basic features of the detected targets in a sorted list, which is formatted to show the route of the target.

Table 2 shows an example of printed results showing the route of the cluster with clusterID 2. The first frame shown is the one with frameID 51. This is due to the fact that the first 50 frames were used for preparing the mask (background) of the scene. Hence, the first frame in which the algorithm searches for targets is the one with frameID 51. The features of coordinates, the distance, and the position in the frame (row and column) were changing at a low rate consistent with the fact that the time distance between each frame was ~0.1 sec. Finally, we can notice that the coordinates that correspond to the frame with frameID 203 are consistent with the position of the target with clusterID 2 in Figure 41.

Table 2. Printed results that show the route of the cluster with clusterID 2.

ID of cluster	ID of frame	length (points that cover)	x (coordinate)	y (coordinate)	z (coordinate)	distance from the sensor	row (of central point)	column (of central point)	time elapsed from start scanning
2	51	10	-26.35	12.10	-0.34	29.00	10	1478	95.27
2	53	9	-26.47	12.01	-0.34	29.07	10	1479	95.47
2	54	9	-26.60	11.95	-0.34	29.17	10	1477	95.57
2	55	7	-26.75	11.95	-0.34	29.30	9	1477	95.67
2	56	9	-26.95	11.92	-0.34	29.47	9	1475	95.77
2	57	9	-27.05	11.92	-0.35	29.56	9	1475	95.87
2	58	8	-27.11	11.85	-0.35	29.59	9	1476	95.97
2	59	9	-27.20	11.84	-0.35	29.67	9	1476	96.07
2	60	8	-27.29	11.84	-0.35	29.75	9	1475	96.17
2	61	9	-27.44	11.73	-0.35	29.84	9	1472	96.27
2	70	10	-28.33	11.45	-0.36	30.56	9	1466	97.17
2	71	8	-28.41	11.39	-0.36	30.61	9	1465	97.27
2	72	9	-28.51	11.34	-0.36	30.68	9	1464	97.37
2	73	8	-28.53	11.31	-0.36	30.70	9	1464	97.47
2	74	7	-28.57	11.21	-0.36	30.69	9	1463	97.57
2	75	7	-28.67	11.16	-0.36	30.77	9	1463	97.67
2	76	6	-28.68	11.07	-0.36	30.74	9	1462	97.77
2	77	6	-28.64	11.06	-0.36	30.71	9	1462	97.87
2	78	7	-28.70	10.96	-0.36	30.72	9	1461	97.97
2	79	8	-28.67	10.88	-0.36	30.67	9	1459	98.07
2	80	8	-28.66	10.78	-0.36	30.62	9	1460	98.17
2	81	7	-28.56	10.82	-0.36	30.54	9	1460	98.27
2	82	7	-28.58	10.71	-0.36	30.53	9	1460	98.37
2	83	8	-28.65	10.63	-0.36	30.56	9	1459	98.47
2	84	7	-28.69	10.62	-0.36	30.59	9	1457	98.57
2	85	6	-28.66	10.56	-0.36	30.55	9	1457	98.67
2	86	7	-28.69	10.57	-0.36	30.57	9	1457	98.77
2	87	6	-28.69	10.52	-0.36	30.56	9	1457	98.87
2	88	7	-28.73	10.46	-0.36	30.57	9	1455	98.97
2	89	8	-28.75	10.47	-0.36	30.60	9	1457	99.07
2	90	7	-28.76	10.44	-0.36	30.59	9	1456	99.17
2	91	7	-28.79	10.43	-0.36	30.62	9	1455	99.27
2	92	7	-28.78	10.42	-0.36	30.61	9	1456	99.37
2	93	6	-28.79	10.40	-0.36	30.61	9	1454	99.47
2	94	5	-28.77	10.38	-0.36	30.59	9	1455	99.57
2	95	7	-28.76	10.37	-0.36	30.57	9	1454	99.67
2	96	7	-28.73	10.38	-0.36	30.55	9	1456	99.77
2	97	7	-28.76	10.40	-0.36	30.58	9	1455	99.87
2	98	7	-28.77	10.36	-0.36	30.58	9	1456	99.97
2	99	6	-28.76	10.36	-0.36	30.57	9	1455	100.07
2	100	6	-28.77	10.37	-0.36	30.58	9	1455	100.17
2	101	6	-28.77	10.38	-0.36	30.58	9	1455	100.27
2	102	6	-28.79	10.35	-0.36	30.59	9	1453	100.37

ID of cluster	ID of frame	length (points that cover)	x (coordinate)	y (coordinate)	z (coordinate)	distance from the sensor	row (of central point)	column (of central point)	time elapsed from start scanning
2	103	6	-28.76	10.35	-0.36	30.56	9	1455	100.47
2	104	7	-28.75	10.38	-0.36	30.57	9	1457	100.57
2	105	6	-28.74	10.39	-0.36	30.56	9	1456	100.67
2	106	7	-28.75	10.40	-0.36	30.57	9	1456	100.77
2	107	7	-28.77	10.42	-0.36	30.60	9	1455	100.87
2	108	7	-28.77	10.41	-0.36	30.59	9	1455	100.97
2	109	6	-28.73	10.39	-0.36	30.55	9	1454	101.07
2	110	6	-28.73	10.40	-0.36	30.55	9	1456	101.17
2	111	6	-28.73	10.40	-0.36	30.55	9	1456	101.27
2	112	7	-28.76	10.42	-0.36	30.59	9	1458	101.37
2	113	7	-28.76	10.41	-0.36	30.58	9	1458	101.47
2	114	7	-28.74	10.37	-0.36	30.56	9	1457	101.57
2	115	7	-28.72	10.39	-0.36	30.54	9	1455	101.67
2	116	7	-28.75	10.38	-0.36	30.57	9	1455	101.77
2	117	6	-28.70	10.40	-0.36	30.53	9	1455	101.87
2	118	8	-28.75	10.40	-0.36	30.58	9	1455	101.97
2	119	8	-28.75	10.39	-0.36	30.57	9	1456	102.07
2	120	7	-28.71	10.38	-0.36	30.53	9	1456	102.17
2	121	8	-28.77	10.39	-0.36	30.59	9	1456	102.27
2	122	7	-28.74	10.40	-0.36	30.56	9	1455	102.37
2	123	7	-28.74	10.38	-0.36	30.55	9	1455	102.47
2	124	7	-28.74	10.35	-0.36	30.55	9	1454	102.57
2	125	7	-28.76	10.40	-0.36	30.59	9	1455	102.67
2	126	7	-28.77	10.38	-0.36	30.59	9	1456	102.77
2	127	7	-28.76	10.37	-0.36	30.57	9	1455	102.87
2	128	7	-28.74	10.37	-0.36	30.56	9	1457	102.97
2	129	7	-28.74	10.38	-0.36	30.56	9	1457	103.07
2	130	7	-28.76	10.35	-0.36	30.57	9	1456	103.17
2	131	6	-28.75	10.38	-0.36	30.57	9	1455	103.27
2	132	6	-28.76	10.38	-0.36	30.58	9	1454	103.37
2	133	7	-28.76	10.35	-0.36	30.57	9	1456	103.47
2	134	7	-28.74	10.40	-0.36	30.56	9	1457	103.57
2	135	7	-28.77	10.36	-0.36	30.58	9	1456	103.67
2	136	7	-28.77	10.37	-0.36	30.58	9	1458	103.77
2	137	7	-28.73	10.36	-0.36	30.54	9	1457	103.87
2	138	7	-28.73	10.37	-0.36	30.54	9	1455	103.97
2	139	7	-28.73	10.36	-0.36	30.54	9	1454	104.07
2	140	7	-28.73	10.39	-0.36	30.55	9	1456	104.17
2	141	7	-28.74	10.38	-0.36	30.56	9	1456	104.27
2	142	8	-28.75	10.39	-0.36	30.57	9	1457	104.37
2	143	8	-28.75	10.40	-0.36	30.58	9	1457	104.47
2	144	8	-28.75	10.40	-0.36	30.58	9	1456	104.57
2	145	7	-28.75	10.44	-0.36	30.59	9	1457	104.67
2	146	7	-28.76	10.43	-0.36	30.60	9	1456	104.77
2	147	7	-28.73	10.41	-0.36	30.56	9	1455	104.87
2	148	7	-28.70	10.42	-0.36	30.53	9	1455	104.97
2	149	8	-28.71	10.44	-0.36	30.55	9	1456	105.07
2	150	7	-28.71	10.42	-0.36	30.54	9	1458	105.17
2	151	7	-28.67	10.44	-0.36	30.51	9	1458	105.27
2	152	6	-28.68	10.42	-0.36	30.52	9	1457	105.37
2	153	7	-28.69	10.45	-0.36	30.54	9	1455	105.47
2	154	7	-28.71	10.42	-0.36	30.55	9	1455	105.57
2	155	7	-28.69	10.43	-0.36	30.53	9	1454	105.67
2	156	7	-28.67	10.46	-0.36	30.52	9	1456	105.77
2	157	7	-28.70	10.49	-0.36	30.56	9	1457	105.87
2	158	7	-28.64	10.43	-0.36	30.48	9	1458	105.97
2	159	7	-28.61	10.49	-0.36	30.48	9	1458	106.07
2	160	7	-28.62	10.45	-0.36	30.47	9	1456	106.17
2	161	7	-28.59	10.42	-0.36	30.44	9	1456	106.27
2	162	7	-28.62	10.44	-0.36	30.47	9	1455	106.37
2	163	7	-28.61	10.42	-0.36	30.45	9	1454	106.47
2	164	7	-28.62	10.42	-0.36	30.46	9	1455	106.57
2	165	6	-28.61	10.47	-0.35	30.47	9	1456	106.67
2	166	7	-28.61	10.44	-0.35	30.46	9	1457	106.77
2	167	7	-28.63	10.42	-0.36	30.47	9	1456	106.87
2	168	7	-28.63	10.44	-0.36	30.47	9	1457	106.97
2	169	6	-28.63	10.40	-0.35	30.46	9	1455	107.07

ID of cluster	ID of frame	length (points that cover)	x (coordinate)	y (coordinate)	z (coordinate)	distance from the sensor	row (of central point)	column (of central point)	time elapsed from start scanning
2	170	7	-28.66	10.41	-0.36	30.49	9	1456	107.17
2	171	7	-28.70	10.43	-0.36	30.54	9	1457	107.27
2	172	7	-28.71	10.42	-0.36	30.54	9	1456	107.37
2	173	7	-28.70	10.41	-0.36	30.54	9	1457	107.47
2	174	7	-28.70	10.42	-0.36	30.53	9	1455	107.57
2	175	7	-28.68	10.44	-0.36	30.53	9	1455	107.67
2	176	7	-28.68	10.46	-0.36	30.53	9	1457	107.77
2	177	7	-28.67	10.45	-0.36	30.52	9	1457	107.87
2	178	7	-28.67	10.45	-0.36	30.52	9	1455	107.97
2	179	7	-28.63	10.42	-0.36	30.47	9	1456	108.07
2	180	7	-28.63	10.47	-0.35	30.48	9	1458	108.17
2	181	7	-28.56	10.44	-0.35	30.41	9	1457	108.27
2	182	7	-28.58	10.49	-0.35	30.44	9	1457	108.37
2	183	6	-28.60	10.42	-0.35	30.44	9	1457	108.47
2	184	9	-28.58	10.48	-0.35	30.44	9	1457	108.57
2	185	7	-28.51	10.46	-0.35	30.37	9	1456	108.67
2	186	7	-28.48	10.51	-0.35	30.36	9	1457	108.77
2	187	7	-28.44	10.53	-0.35	30.33	9	1458	108.87
2	188	8	-28.36	10.59	-0.35	30.27	9	1460	108.97
2	189	9	-28.33	10.67	-0.35	30.27	9	1460	109.07
2	190	8	-28.25	10.67	-0.70	30.21	10	1459	109.17
2	191	8	-28.20	10.72	-0.35	30.17	9	1459	109.27
2	192	6	-28.11	10.74	-0.35	30.09	9	1459	109.37
2	193	6	-28.08	10.77	-0.35	30.08	9	1460	109.47
2	194	6	-28.04	10.80	-0.35	30.05	9	1460	109.57
2	195	6	-28.01	10.80	-0.35	30.02	9	1462	109.67
2	196	7	-27.97	10.83	-0.35	30.00	9	1463	109.77
2	197	6	-27.93	10.89	-0.35	29.98	9	1463	109.87
2	198	7	-27.84	10.91	-0.35	29.91	9	1464	109.97
2	199	7	-27.75	11.00	-0.35	29.86	9	1463	110.07
2	200	6	-27.72	11.00	-0.35	29.82	9	1465	110.17
2	201	6	-27.68	11.07	-0.35	29.82	9	1465	110.27
2	202	7	-27.65	11.08	-0.35	29.79	9	1466	110.37
2	203	7	-27.62	11.15	-0.35	29.79	9	1466	110.47

C. FLIGHT TEST DATA PROCESSING

The data collected by the LiDAR sensor were processed with the developed algorithm. The test was set up as described in Chapter 3, and the LiDAR sensor scanned the surrounding environment while various sUAVs were flying. The data collected through this procedure were stored and ultimately processed by the algorithm described in the previous sections. The results of the application of the algorithm were both visualized and printed.

Afterwards, the printed data were processed further through common electronic spreadsheet programs. Simple processes through these spreadsheets provided interesting information, such as the velocities of the sUAVs. Hence, all the significant information about the detected sUAVs, like their distance from the LiDAR sensor, their velocities, their altitude, the points of the frame that they cover, as well as their relation to one another, were processed to provide meaningful and fruitful results.

The results of the data processing were grouped for easier comparison. Hence, the points that a sUAV covers in a frame were distributed in four groups: 1 to 2, 3 to 5, 6 to 9, and 10+ points. Also, the distances from the sensor were distributed in four groups: 0 to 25, 25 to 35, 35 to 45, and 45+ meters (m). The velocities were also distributed in four groups: 0 to 1, 1 to 3, 3 to 5, and 5+ meters per second (m/s). The altitude from the sensor (where the sensor is in 0 altitude) were broken into four groups: 0 to 1, 1 to 3, 3 to 5, and 5+ meters (m). Furthermore, as described in Chapter III, we made some abstract assumptions and averaging to estimate the percentage of sUAVs detected according to their motion attributes.

D. EVALUATION OF THE RESULTS COMPARISONS

The procedures just described provided some significant results. The main conclusion we drew was that the major factors contributing to the successful detection of sUAVs are their distance from the sensor and the size of the sUAVs. Specifically, from the results, we note that as the distance from the sensor was increasing, the number of points that the sUAV covered in the frame was decreasing. It is common that for distances larger than 45 meters, almost 80% of the detected sUAVs covered only one or two points in the frame, whereas for distances less than 25 meters the sUAVs that covered one or two points accounted for less than 10% of the detected sUAVs. Figure 42 shows the indisputable relationship between the points that an sUAV covered in a frame and its distance from the LiDAR sensor.

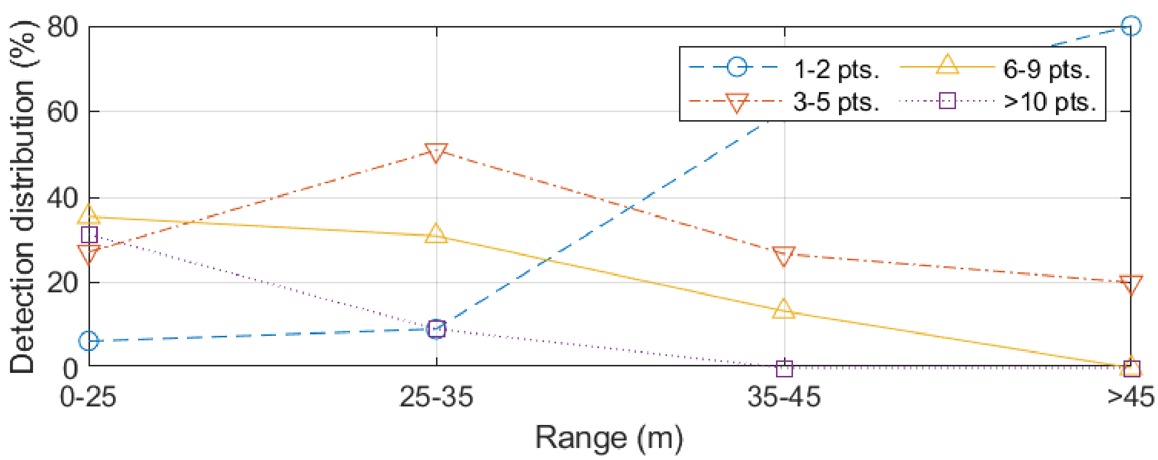


Figure 42. Relationship between the number of points that sUAVs covered in a frame and the sUAVs' distance from the LiDAR sensor.

The previous comparisons were in agreement with the results regarding the relationship between the percentage of sUAVs detected and their distance from the sensor. In particular, as their distance from the sensor was increasing, the percentage of sUAVs detected was decreasing. It is indicative that for distances larger than 45 meters, less than 10% of the sUAVs were detected, whereas for distances less than 25 meters more than 90% of the sUAVs were detected. Figure 43 shows the obvious relationship between the percentage of sUAVs detected and their distance from the LiDAR sensor.

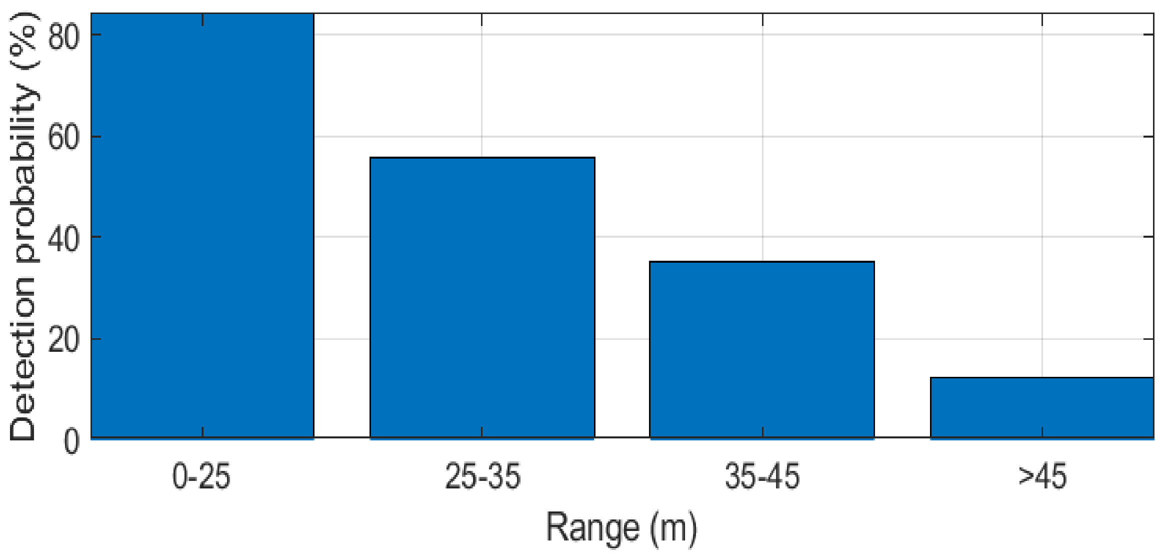


Figure 43. Relationship between the percentage of sUAVs detected and their distance from the LiDAR sensor.

The clear relationship between the successful detection of sUAVs and their distance from the sensor is consistent with the function of the LiDAR sensor. As was mentioned in Chapter I and presented in Figure 10, there are gaps between the points in each frame. These gaps are significant, especially in the vertical direction. In particular, the vertical gaps for distances from the sensor equal to 30, 50, and 100 meters are 0.70, 1.16, and 2.33 meters, respectively. Also, the horizontal gaps for distances from the sensor equal to 30, 50, and 100 meters are 0.10, 1.17, and 0.35 meters, respectively. Furthermore, sUAVs are inherently small, and their vertical dimensions are normally much smaller than their horizontal dimensions. Hence, the probability that a sUAV could be located within these gaps increases as the distance from the sensor increases.

By contrast, the results did not indicate any relation between the velocity of the sUAVs and the points that they cover in a frame. Despite significant changes in the distribution of the percentage to which each group corresponded, a specific trend in these changes was not observed that would indicate a relation among them. Figure 44 presents the relationship between the points that an sUAV covered in a frame and its velocity.

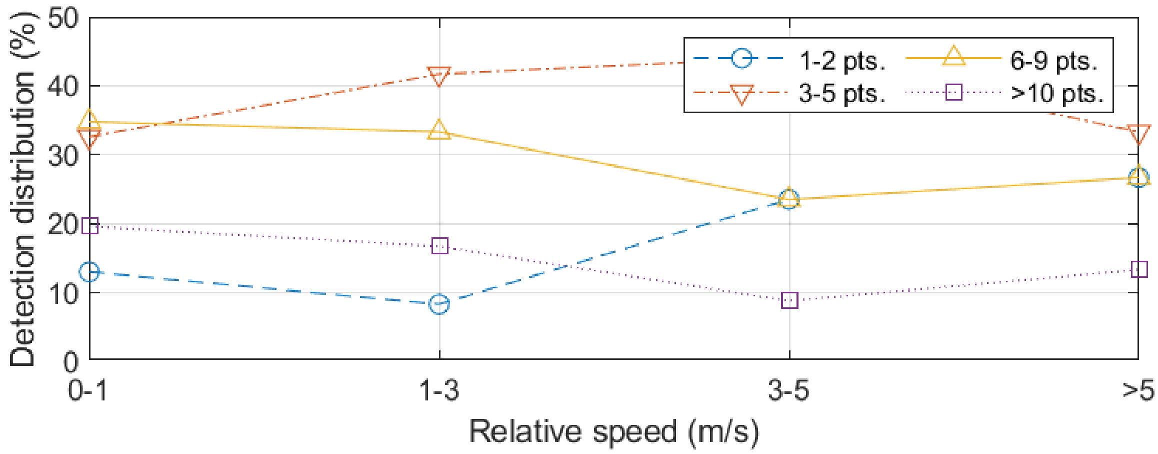


Figure 44. Relationship between the number of points that sUAVs covered in a frame and their velocity.

Similar to the velocity of the sUAVs, the results did not indicate any relation between the altitude of the sUAVs and the points that they cover in a frame. Again, there were significant changes in the distribution of the percentage to which each group corresponded, but a specific trend in these changes was not apparent. Hence, there is no indication of any relation among them. Figure 45 presents the relationship between the points that an sUAV covered in a frame and the altitude of the sUAV.

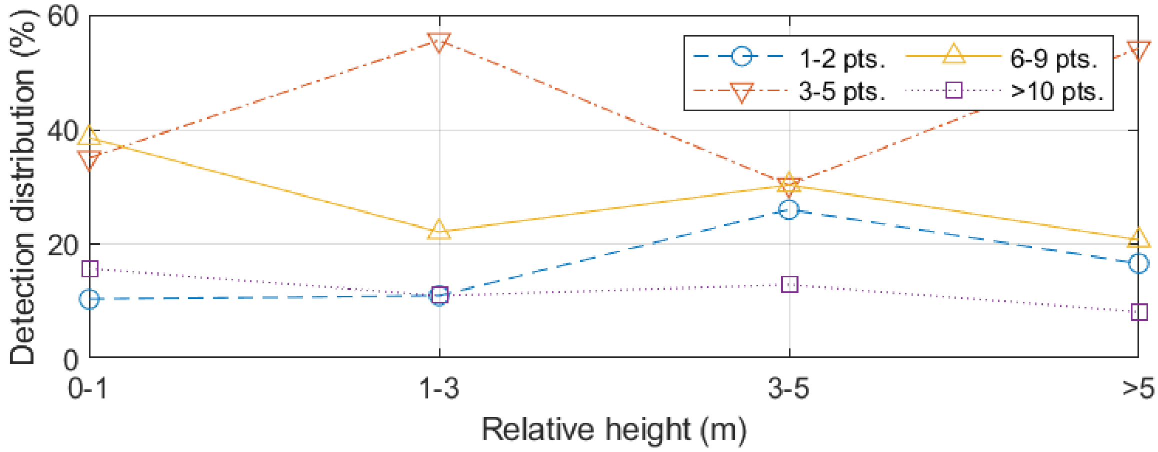


Figure 45. Relationship between the number of points that sUAVs covered in a frame and their altitude.

Furthermore, the results concerning the false detections of sUAS were significant. In particular, one major concern was the effectiveness of the algorithm in distinguishing the actual sUAS from other small moving objects. As discussed in Section IV.A, the main methods that we applied to achieve this were to exclude the ground and objects that were covered by the "mask" that we derived from the background. Specifically, small plants, as well as trees and their branches and leaves, were probable sources of false positive detections. Moreover, because the experiments took place in rural environments this issue was very pronounced.

The results of the comparisons of the collected LiDAR data that we processed were revealing about the false detection issue. Indeed, there were many false detections depending on the method we applied and the environment that we investigated. The worst-case scenario was when we searched for sUAS at low altitude; there were trees in the background and the mask that we applied was derived from only a few frames. In contrast, the best-case scenario was when we investigated for sUAS at an altitude above the sensor altitude, while we applied a mask that was derived from an adequate number of frames.

Figure 46 presents the number of false detections related to the factors just described. In the first case, there were no constraints regarding the minimum altitude in searching for sUAS. Given this, two possible masks were applied. One was derived after using the data of 10 frames, and the other was derived by using the data from 50 frames.

In addition to these results, in the second case we searched at an altitude above the level of the sensor. Here we again applied two masks, one mask that used the data from 10 frames and one mask after 50 frames. The results, presented to Figure 46, came after searching for sUAS in 100 successive frames. In conclusion, these results confirmed the challenges that small plants (near the ground) add to the procedure and the decisive contribution of the application of the mask to the effectiveness of the algorithm.

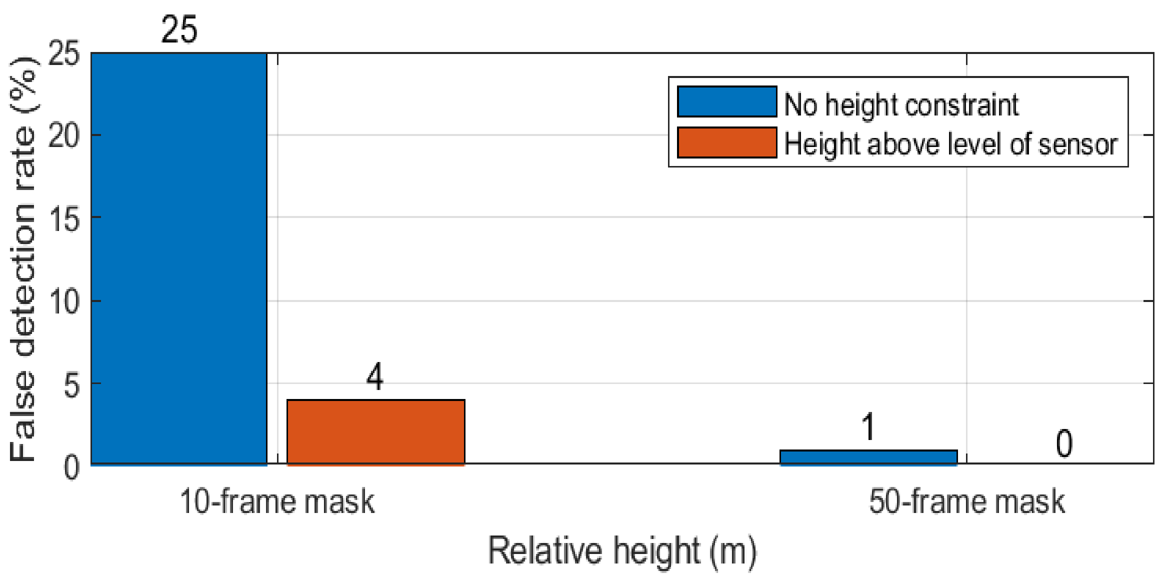


Figure 46. False detection rate per 100 frames with height constraint and depth mask applied

The preceding plots were derived from the data collected from the sUAVs with a maximum dimension greater than 60 cm. The data from the sUAVs with a maximum dimension of less than 60 cm confirmed the aforementioned results. The only difference was that the detection of these sUAVs was mainly limited to a range of 20 meters from the sensor. Beyond this distance their detection was negligible.

V. CONCLUSIONS

This final chapter presents the conclusions we can draw from the procedures presented in this research, and the chapter closes with recommendations for future research.

A. CONCLUSIONS

Considering our findings from the evaluation of the developed algorithm, we conclude that LiDAR sensors are capable of detecting moving objects and, specifically, sUAVs. Nevertheless, from the experimental results it is obvious that there are many restrictions and obstacles in this procedure. The range of detection is the most crucial restriction. Given the functionality and the limitations of commercial LiDAR sensors, in combination with the abstract features of the sUAVs in terms of size, shape, route of flights, and so forth, it is very challenging to successfully detect sUAVs with LiDAR sensors at long range.

On the other hand, this type of sensor is ideal for determining the accurate location of sUAVs, given that they are already detected. The major advantage of these sensors is that they can precisely locate the exact position of each object they detect. In addition, as we found in this research, this capability is independent of the altitudes and the velocities of the sUAVs.

In conclusion, it was proven that it is possible to detect an sUAS even with a commercial LiDAR sensor. Moreover, it was shown that the developed algorithm runs in real time, even in the interpretive environment of MATLAB. In particular, the time needed to process each frame of data was about 0.1 seconds, which is approximately the same time that the Velodyne 3D LiDAR sensor needs for creating a frame in real time.

B. RECOMMENDATIONS – FUTURE RESEARCH

The research on the capabilities of 3D LiDAR sensors for detecting sUAVs is a promising field that should be studied and investigated in depth. Hence, many more studies could and should be performed in this field.

One possible research for further exploration is the effectiveness and the necessity of the use of different sensors in combination. There are many types of sensors that are used for detecting objects. All of them present advantages and disadvantages. So, it would be interesting to study the potential of various combinations of sensors, including LiDAR sensors, and their consequent advantages and disadvantages.

Additionally, a follow-on study should focus specifically on LiDAR sensors with limited FOV but increased resolution and range. These sensors should function in combination with other types of sensors. The goal of the other sensors should be the general location of possible targets. After this general location is acquired, the LiDAR sensors should undertake the mission of detecting the accurate position of the suspected target, determine its size and shape, and track its route. The information provided by the LiDAR sensor should then be used to define the texture of this prospective target and clarify its status, and consequently, to classify or reject it as a potential target.

Furthermore, it should be investigated the improvement of the time performance by coding the algorithm in Verilog and running it on a field-programmable gate array (FPGA) [31]. The parallelism in the execution of the algorithm that FPGA can provide, could give the capability to add many more functionalities to the algorithm that either will increase the effectiveness of detection or will add additional capabilities, while sustaining the real-time execution of the code.

LIST OF REFERENCES

[1] M. Laurenzis, S. Hengy, A. Hommes, F. Kloeppel, A. Shoykhetbrod, T. Geibig, W. Johanneset al., "Multi-sensor field trials for detection and tracking of multiple small unmanned aerial vehicles flying at low altitude," in *SPIE Defense + Security*, May 2017 [Online]. Available: https://doi.org/10.1117/12.2261930

[2] M. Hammer, M. Hebel, M. Arens, and M. Laurenzis, "Lidar-based detection and tracking of small UAVs," in *Emerging Imaging and Sensing Technologies for Security and Defence III; and Unmanned Sensors, Systems, and Countermeasures*, Oct. 2018 [Online]. Available: https://doi.org/10.1117/12.2325702.

[3] M. Hammer, M. Hebel, B. Borgmann, M. Laurenzis, and M. Arens, "Potential of lidar sensors for the detection of UAVs," in *Laser Radar Technology and Applications XXIII*, May 2018 [Online]. Available: https://doi.org/10.1117/12.2303949

[4] M. Hammer, B. Borgmann, M. Hebel, and M. Arens, "UAV detection, tracking, and classification by sensor fusion of a 360° lidar system and an alignable classification sensor," in *Laser Radar Technology and Applications XXIV*, May 2019 [Online]. Available: https://doi.org/10.1117/12.2518427

[5] M. Salhi and N. Boudriga, "Multi-Array Spherical LIDAR System for Drone Detection," in *2020 22nd International Conference on Transparent Optical Networks (ICTON)*, Jul. 2020 [Online]. Available: https://doi.org/10.1109/ICTON51198.2020.9203381

[6] R. C. Olsen, *Remote Sensing from Air and Space*, 2nd ed. Bellingham, WA, USA: SPIE Press, 2016, pp. 247–250

[7] M. H. Kim *et al.*, "The CALIPSO version 4 automated aerosol classification and lidar ratio selection algorithm," *Atmospheric Measurement Techniques*, vol. 11, no. 11, pp. 6107–6135, Nov. 2018 [Online]. Available: https://doi.org/10.5194/amt-11-6107-2018

[8] A. I. Carswell, "Advances in laser ranging," in *Proceedings of Laser Radar Technology for Remote Sensing*, vol. 5240, Jan. 2004 [Online]. Available: https://doi.org/10.1117/12.510233

[9] D. W. Gast, "LIDAR design for space situational awareness," M.S. thesis, Dept. of Astronautical Engineering, NPS, Monterey, CA, USA, 2008 [Online]. Available: http://hdl.handle.net/10945/3994

[10] M. F. Helt, “Vegetation identification with Lidar,” M.S. thesis, Dept. of Space Systems Operations, NPS, Monterey, CA, USA, 2005 [Online]. Available: http://hdl.handle.net/10945/2030

[11] *VLP-16 User Manual.* Velodyne LiDAR, Inc., San Jose, CA, USA, 2019 [Online]. Available: https://velodynelidar.com/downloads/#manuals%20first

[12] *Ouster Studio User Guide.* Ouster, San Francisco, CA, USA, 2019 [Online]. Available: https://data.ouster.io/ouster-studio/Ouster%20Studio%20User%20Guide.pdf

[13] MathWorks, “MATLAB - MathWorks - MATLAB & Simulink.” Accessed Jul. 06, 2021 [Online]. Available: https://www.mathworks.com/products/matlab.html

[14] C. A. Fowler and R. J. Hammel, “Converting PCAPs into Weka mineable data,” in *15th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, Jun. 2014[Online]. Available: https://doi.org/10.1109/SNPD.2014.6888681.

[15] *Software User Manual.* Ouster, San Francisco, CA, USA, 2021 [Online]. Available: https://data.ouster.io/downloads/software-user-manual/software-user-manual-v2p0.pdf

[16] MathWorks, “Point Cloud - MATLAB & Simulink.” Accessed July 06, 2021 [Online]. Available: https://www.mathworks.com/discovery/point-cloud.html?s_tid=srchtitle

[17] O. A. Yakimenko, “*Engineering Computations and Modeling in MATLAB®/Simulink®,*” Reston ,VA, USA: American Institute of Aeronautics and Astronautics, 2011.

[18] Point Cloud Library (PCL), “pcl.” Accessed Jul. 06, 2021 [Online]. Available: https://pointclouds.org/

[19] R. B. Rusu and S. Cousins, “3D is here: Point Cloud Library (PCL),” in *2011 IEEE International Conference on Robotics and Automation*, May 2011 [Online]. Available: https://doi.org/10.1109/ICRA.2011.5980567.

[20] MathWorks, “Lidar Toolbox - MATLAB.” Accessed Jul. 06, 2021 [Online]. Available: https://www.mathworks.com/products/lidar.html

[21] MathWorks, “Segmentation - MATLAB & Simulink.” Accessed Jul. 06, 2021 [Online]. Available: https://www.mathworks.com/help/lidar/deeplearning.html

[22] MathWorks, “Read next video frame - MATLAB readFrame.” Accessed Jul. 06, 2021 [Online]. Available: https://www.mathworks.com/help/matlab/ref/videoreader.readframe.html

[23] *Product Guide.*Velodyne LiDAR, Inc., San Jose, CA, USA [Online]. Available: https://velodynelidar.com/downloads/#product_guides%20first

[24] *High Resolution Real-Time Lidar Sensor.*Velodyne LiDAR, Inc., San Jose, CA, USA, 2019 [Online]. Available: https://velodynelidar.com/downloads/#manuals%20first

[25] Velodyne Lidar, "Puck Hi-Res Lidar Sensor." Accessed Aug. 12, 2021 [Online]. Available: https://velodynelidar.com/products/puck-hi-res/

[26] *Veloview User Guide.* Velodyne LiDAR, Inc., San Jose, CA, USA, 2017 [Online]. Available: https://usermanual.wiki/Document/VeloViewUserGuide.686051238.pdf

[27] W. Puyati and A. Walairacht, "Efficiency improvement for unconstrained face recognition by weighting probability values of modular PCA and wavelet PCA," in *2008 10th International Conference on Advanced Communication Technology*, Feb. 2008 [Online]. Available: https://doi.org/10.1109/ICACT.2008.4494037.

[28] Y. Sun, "Combined neural network and PCA for complicated damage detection of bridge," in *2009 Fifth International Conference on Natural Computation*, 2009 [Online]. Available: https://doi.org/10.1109/ICNC.2009.580.

[29] MathWorks, "Read point cloud data from Velodyne PCAP file - MATLAB." Accessed Aug. 13, 2021 [Online]. Available: https://www.mathworks.com/help/vision/ref/velodynefilereader.html

[30] A. García and J. Bescós, "Real time video foreground extraction based on context-aware background subtraction," Mar. 25, 2015 [Online]. Available: https://www.researchgate.net/publication/228806073.

[31] Uwe Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Arrays*, 3rd ed. Berlin Heidelberg: Springer, 2007

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California