ERDC TR-21-4

US Army Corps
of Engineers®
Engineer Research and
Development Center

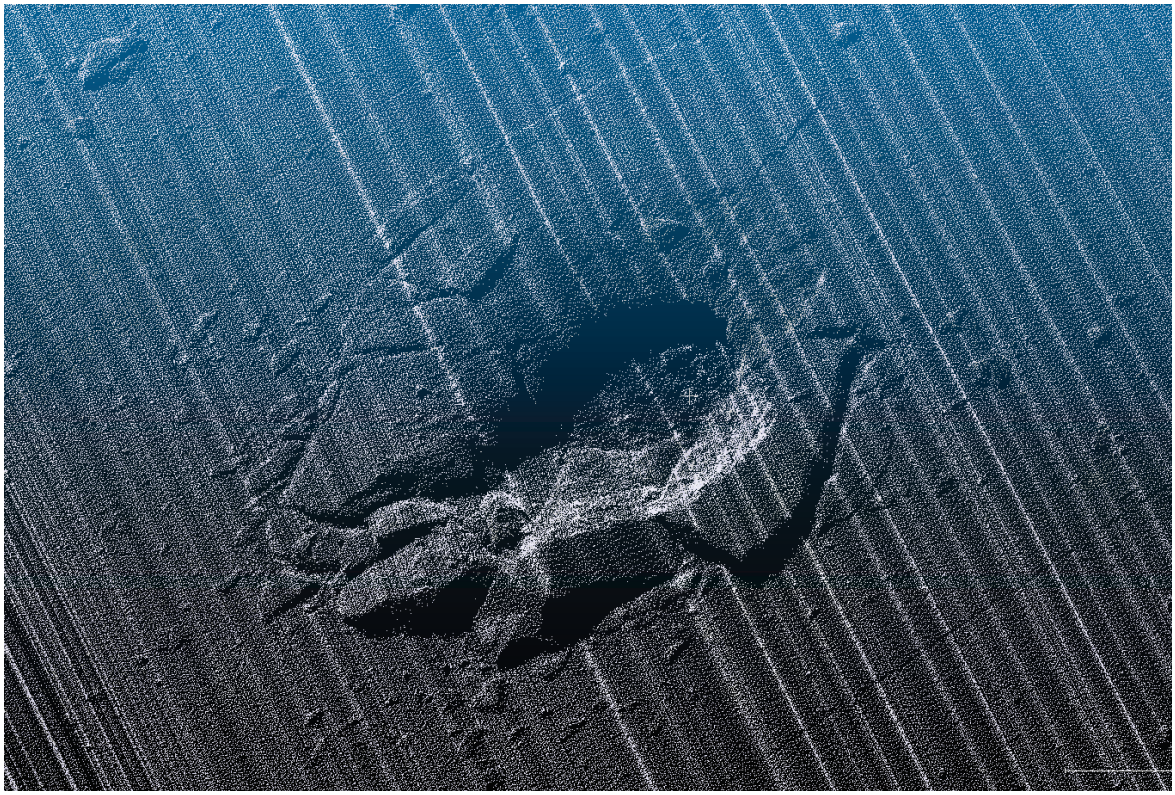
ERDC
ENGINEER RESEARCH & DEVELOPMENT CENTER

Rapid Airfield Damage Assessment System (RADAS)

First Generation Automated Assessment of Airfield Damage using LiDAR Point Clouds

Ernest S. Berney IV, Andrew B. Ward, and Naveen B. Ganesh

March 2021



Engineer Research and
Development Center

Approved for public release; distribution is unlimited.

The U.S. Army Engineer Research and Development Center (ERDC) solves the nation's toughest engineering and environmental challenges. ERDC develops innovative solutions in civil and military engineering, geospatial sciences, water resources, and environmental sciences for the Army, the Department of Defense, civilian agencies, and our nation's public good. Find out more at www.erdc.usace.army.mil.

To search for other technical reports published by ERDC, visit the ERDC online library at https://erdclibrary.on.worldcat.org/discovery.

First Generation Automated Assessment of Airfield Damage from LiDAR Point Clouds

Ernest S. Berney IV and Andrew B. Ward

Geotechnical and Structures Laboratory
U.S. Army Engineer Research and Development Center
3909 Halls Ferry Road
Vicksburg, MS 39180

Naveen B. Ganesh

Coastal and Hydraulics Laboratory
U.S. Army Engineer Research and Development Center
3909 Halls Ferry Road
Vicksburg, MS 39180

Final report

Approved for public release; distribution is unlimited.

Prepared for U.S. Army Corps of Engineers
Washington, DC 20314-1000

Under Project 480403, "Automatic Pavement Damage Assessment using LiDAR," and Project 471376, "Airfield Damage Detection Using LiDAR Point Clouds"

Abstract

This research developed an automated software technique for identifying type, size, and location of man-made airfield damage including craters, spalls, and camouflets from a digitized three-dimensional point cloud of the airfield surface. Point clouds were initially generated from Light Detection and Ranging (LiDAR) sensors mounted on elevated lifts to simulate aerial data collection and, later, an actual unmanned aerial system. LiDAR data provided a high-resolution, globally positioned, and dimensionally scaled point cloud exported in a LAS file format that was automatically retrieved and processed using volumetric detection algorithms developed in the MATLAB software environment. Developed MATLAB algorithms used a three-stage filling technique to identify the boundaries of craters first, then spalls, then camouflets, and scaled their sizes based on the greatest pointwise extents. All pavement damages and their locations were saved as shapefiles and uploaded into the GeoExPT processing environment for visualization and quality control. This technique requires no user input between data collection and GeoExPT visualization, allowing for a completely automated software analysis with all filters and data processing hidden from the user.

DISCLAIMER: The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products. All product names and trademarks cited are the property of their respective owners. The findings of this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

DESTROY THIS REPORT WHEN NO LONGER NEEDED. DO NOT RETURN IT TO THE ORIGINATOR.

Contents

Abstract ii

Figures and Tables vi

Preface xi

Executive Summary xii

1 Introduction 1

1.1 Purpose 1
1.2 Scope 3
1.3 Deliverables 4

2 Methodology 6

2.1 Sensors 8
2.1.1 Terrestrial LiDAR 9
2.1.2 Airborne LiDAR 12
2.2 Analysis software 15
2.2.1 RiAcquire and RiWorld 15
2.2.2 MATLAB 15
2.2.3 Geospatial Expeditionary Planning Tool 19
2.3 Integration 20

3 Data Collection 22

3.1 Test location 22
3.2 Single crater event (August 2016) 24
3.2.1 Airborne imagery collection 24
3.2.2 Ground-based LiDAR collection 26
3.2.3 Scene creation with historical LiDAR scans 28
3.3 Six crater event (December 2016) 29
3.3.1 Airborne imagery collection 30
3.3.2 LiDAR image collection 31
3.4 12 crater event (August 2017) 34
3.4.1 Airborne EO and thermal image collection 36
3.4.2 LiDAR setup with telehandler 38
3.5 Obscurant event, Fort Polk (December 2017) 42
3.5.1 LiDAR data collection 43
3.5.2 Reduction of smoke interference in point clouds 45
3.6 Avon Park event (May 2018) 49
3.6.1 LiDAR data Collection 51
3.6.2 Automated data processing 54
3.7 Silver Flag event (October 2018) 55
3.7.1 Telehandler LiDAR with NUC and Ubiquiti antennae 57
3.7.2 Riegl RiCopter UAS deployment 63

3.8 Silver Flag event (March 2019) 67
3.8.1 Reigl RiCopter LiDAR collection 69

4 Algorithm Development 73
4.1 Mesh creation methods 73
4.2 MATLAB functions for damage detection 77
4.3 Z-deviation method 78
4.3.1 Algorithm introduction 78
4.3.2 Initialization 79
4.3.3 Generate pre-surface 79
4.3.4 Generate difference-surface 80
4.3.5 Z-deviation scan 81
4.3.6 Damage classification 82
4.3.7 MATLAB implementation 82
4.3.8 Future improvements 89
4.4 Depression-filling method 90
4.4.1 Algorithm introduction 90
4.4.2 Initialization 91
4.4.3 Surface flooding 92
4.4.4 Surface draining 93
4.4.5 Isolating depressions 95
4.4.6 Damage classification 96
4.4.7 MATLAB implementation 97
4.4.8 Future improvements 103

5 Validation of Damage Detection 105
5.1 Depression-filling analysis for damage detection 105
5.1.1 Development of LiDAR tiles for analysis 105
5.1.2 August 2016 – artificial airfield 106
5.1.3 December 2016 – 6 crater event 107
5.1.4 August 2017 – 12 crater event 108
5.1.5 December 2017 – 2 crater obscurant event 109
5.1.6 March 2019 – 7 crater event 110
5.1.7 Summary of DFA damage detection 114
5.2 Z-deviation analysis for December 2016 test event 115

6 Data Delivery 119
6.1 AutoCAD shapefiles for GeoExPT 119
6.2 Identification of damage features in GeoExPT 121
6.2.1 Crater identification 121
6.2.2 Spall identification 122
6.2.3 Camouflet identification 126

7 Conclusions and Recommendations 128
7.1 Conclusions 128
7.2 Recommendations 129

7.2.1 LiDAR as a sensor platform 129
7.2.2 Integration of hardware 130
7.2.3 Algorithm advances 130
7.2.4 Continued data collection 131

References 132

Appendix A: MATLAB Damage Detection 133

Unit Conversion Factors 161

Report Documentation Page

Figures and Tables

Figures

Figure 1. Example of crater found in concrete pavement. 1
Figure 2. Example of spall field in concrete pavement. 2
Figure 3. Camouflet hole from subsurface detonation. 2
Figure 4. Illustration of LiDAR pulse and return. 9
Figure 5. Reigl VZ-2000 LiDAR scanner mounted on vehicle. 11
Figure 6. Reigl VZ-2000 LiDAR scanner mounted on cage on telehandler lift. 11
Figure 7. RiCopter with LiDAR scanner, GNSS/INS navigation system, and RGB cameras attached to the belly of the aircraft. 13
Figure 8. Two operators evaluating RiCopter prior to flight. 14
Figure 9. Pilot and observer for Ricopter. Base station antennae in foreground; Ubiquiti data transfer antennae in background. 14
Figure 10. RiCopter mission planning software outlining flight path along Silver Flag runway from launch point at the south taxiway intersection. 15
Figure 11. MATLAB software application to detect crater locations. 17
Figure 12. Modern version of MATLAB software showing crater detection and elevation scale; blue is below ground up to red, which is at the surface of the runway. 17
Figure 13. Scene of Silver Flag airfield with various UXO located throughout. 18
Figure 14. Scene of Silver Flag airfield on an intensity-scaled airfield with UXO identified as green intensity and orange and yellow as concrete. 19
Figure 15. Screenshot of GeoExPT GUI with crater and spall damage overlain on a mosaic of the damaged airfield. 20
Figure 16. Diagram of the ERDC automated LiDAR data processing for RADAS. 21
Figure 17. Silver Flag airfield located southeast of Tyndall Air Force base. 22
Figure 18. Highlighted area is south end of Silver Flag runway used for crater testing. 23
Figure 19. Silver Flag runway looking south where cratering operations occur. 23
Figure 20. Single crater event at Silver Flag. 24
Figure 21. DJI Phantom 3 (top left) and Turbo Ace Matrix E-UAS with Sony Alpha camera (right). 25
Figure 22. Point cloud reconstruction from Silver Flag crater (Matrix-E imagery); flags highlight specific UXO locations. 26
Figure 23. VZ-2000 LiDAR mounted to rear of truck to provide surface scan of crater. 27
Figure 24. LiDAR scan (intensity view) of crater at Silver Flag. 27
Figure 25. Artificial 6-crater airfield from independent LiDAR scans (top) and detail of central section (bottom). 29

Figure 26. Mosaic of 6-crater event at Silver Flag from Matrix photographs (top) and reconstructed point cloud from Matrix-E imagery (bottom). 31
Figure 27. Complete LiDAR image of 6-crater event. 32
Figure 28. LiDAR closeup of 3-crater and 2-spall set on south end of crater event. 32
Figure 29. LiDAR closeup of three craters and one camouflet on north end of crater event. 33
Figure 30. LiDAR mesh of crater at Silver Flag from angle (top) and side (bottom). 34
Figure 31. North third of test section (left), central third of test section (right) at Silver Flag – 12-crater event. 35
Figure 32. South third of test section at Silver Flag – 12-crater event. 36
Figure 33. Launch of Matrix-E with FLIR Tau-2 thermal camera and base station. 37
Figure 34. Real-time display of thermal imagery from Matrix-E drone during flight. 38
Figure 35. Assembly of LiDAR on aluminum frame for telehandler: Aluminum frame (top left), securing VZ-2000 on custom aluminum mount and stringing 100-ft Ethernet cables to telehandler (top right and lower left), and final assembly with generation mounted to pallet secured to bottom of aluminum frame (bottom right). 39
Figure 36. VZ-2000 mounted to an aluminum cage hoisted on a telehandler lift providing a sensor height of approximately 50 ft above ground level. 40
Figure 37. Scan angle limitations of the VZ-2000 Mobile LiDAR scanner. 41
Figure 38. LiDAR point cloud mesh of crater from telehandler perspective. 42
Figure 39. Colorized LiDAR scan of obscurant test section at Fort Polk with three craters and closeup of one of the crater blasts. 43
Figure 40. Telehandler mounted VZ-2000 LiDAR scanning the crater prior to application of smoke. 44
Figure 41. Delivery of smoke to test pad. 44
Figure 42. Telehandler mounted VZ-2000 Mobile LiDAR scanning in smoke. 45
Figure 43. LiDAR point cloud of damaged surface from top (top) and profile (bottom). 47
Figure 44. LiDAR point cloud of craters and camouflets after smoke removal (top) and profile (bottom). 48
Figure 45. LiDAR point cloud of two craters following removal of smoke particles in point cloud. 49
Figure 46. MacDill Auxiliary Field test locations (1=asphalt, 2=concrete, 3=gravel). 50
Figure 47. Image of asphalt section with UXO and tire crater simulants. 50
Figure 48. Tire inner tubes and UXO on gravel assault strip. 51
Figure 49. Reigl VZ-2000 mobile LiDAR mounted on telehandler at Avon Park, FL. 52
Figure 50. Overhead mosaic of a section of runway with concrete test pad and tires. 53
Figure 51. LiDAR mesh of Avon Park tires and UXO on asphalt surface. 54
Figure 52. Logistics of automated data processing at Avon Park data collect. 55
Figure 53. Layout of tires on Silver Flag near North taxiway. 56

Figure 54. Example of UXO layout on airfield surface (Rockets and Blue 97s visible). 56
Figure 55. Windows-based Intel NUC 7 computer. 58
Figure 56. Use of Reigl VZ-2000 mobile LiDAR on telehandler with NUC computer mount. 58
Figure 57. Close-up of Ubiquiti airMAX Sector antenna. 59
Figure 58. Close-up of Ubiquiti airMAX Omni antenna. 60
Figure 59. Ubiquiti ground station setup. 61
Figure 60. Pelican case transport of Ubiquiti equipment. 62
Figure 61. Ubiquiti receiver antenna connected via Ethernet to land-based laptop monitored from a ground station. 63
Figure 62. Reigl RiCopter LIDAR UAS with VUX-1LR system mounted underneath. 64
Figure 63. RiCopter pilot with control station monitoring UAS in the air (left), RiCopter ground station with mission planning software laptop and communication antenna (right). 64
Figure 64. Image of flight plan for RiCopter along south end of Silver Flag test site. 65
Figure 65. Pre-flight inspection of RiCopter with two-man team. 65
Figure 66. Colored LiDAR image of two tires place on surface of Silver Flag runway. 67
Figure 67. Location of six craters plus cratered camouflet (item 7) at Silver Flag. 68
Figure 68. Location of two spalls and two camouflets at south end of crater field. 68
Figure 69. Location of additional camouflets and tire field south of the blast area. 69
Figure 70. Diagram of intended damage features on Silver Flag runway. 69
Figure 71. Reigl RiCopter undergoing preflight check list (left) and ground control station with mission planner laptop, radio transmitter, and pilot. 71
Figure 72. RiCopter prepared for twilight operations (30 min past sunset). 71
Figure 73. RiCopter lighting for visual tracking during twilight operations. 72
Figure 74. Delaunay triangulation and Voronoi Diagram for a random 3-D point cloud input. 75
Figure 75. Comparison of various mesh creation techniques. 76
Figure 76. Example of 500 x 500 grid applied to point cloud for grid estimation. 77
Figure 77. Changing z-deviation with changing surface geometry. 78
Figure 78. Comparison of cross section and all-points good area selections for z-deviation method. 80
Figure 79. Difference-surface elevation change for z-deviation method. 81
Figure 80. Combination of marked grids into one larger grid for the z-deviation method. 82
Figure 81. MATLAB-developed GUI for z-deviation method. 83
Figure 82. Load file portion from the z-deviation method GUI. 84
Figure 83. Mesh plot of imported post-surface from the z-deviation method GUI. 84
Figure 84. Cross-section menu from the z-deviation method GUI. 85

Figure 85. Width and spacing parameters for the z-deviation method GUI. 86
Figure 86. Create surface from cross-sections menu from the z-deviation method GUI. 87
Figure 87. Find craters menu from the z-deviation method GUI. 87
Figure 88. Mesh plot with damage indicator boxes from the z-deviation method GUI. 88
Figure 89. Excel table with damage locations from the z-deviation method GUI. 88
Figure 90. Save data functions from the z-deviation method GUI. 89
Figure 91. No-sink algorithm flooding example. 92
Figure 92. Depression-filling example on damaged pavement surface. 94
Figure 93. Cross-section view of a fictional crater for the no-sink method. 95
Figure 94. Before and after crater field using a depression-filling algorithm. 96
Figure 95. MATLAB-developed GUI for method three. 98
Figure 96. GUI options for initialization steps one, two, and three. 99
Figure 97. Crater section of the damage detection GUI. 100
Figure 98. Camouflet section of the damage detection GUI. 100
Figure 99. Spall section of the damage detection GUI. 101
Figure 100. Large to small depression-filling damage classification. 102
Figure 101. Start/stop section of the damage detection GUI. 103
Figure 102. Excel file output from the damage detection GUI. 103
Figure 103. Profile view of a point cloud and corresponding surface reconstructions. 104
Figure 104. August 2016 reference LiDAR point cloud tiles for analysis. 107
Figure 105. December 2016 reference LiDAR point cloud tiles for analysis. 108
Figure 106. August 2017 reference LiDAR point cloud tiles for analysis. 109
Figure 107. December 2017 reference LiDAR point cloud tile for analysis (with smoke). 110
Figure 108. March 2019 reference LiDAR point cloud tiles for analysis. 111
Figure 109. Lessening of crater depth during mesh development as point cloud resolution decreases. 112
Figure 110. March 2019 nighttime reference LiDAR point cloud tiles for analysis. 114
Figure 111. Z-deviation MATLAB settings for first trial. 116
Figure 112. Z-deviation MATLAB settings for second trial. 117
Figure 113. Results of crater detection from first trial (left) and second trial (right). 117
Figure 114. GeoExPT from August 2017 showing damage features. 120
Figure 115. Close up of GeoExPT from August 2017 showing damage features. 121
Figure 116. Image mosaic of three craters locations from August 2017 event (Tile 2). 122
Figure 117. Image mosaic of two spall locations from August 2017 event (Tile 1). 124
Figure 118. Image mosaic of two spall locations from August 2017 event (Tile 6). 125

Figure 119 Image mosaic of one camouflet location from August 2017 event (Tile 5). 127

Tables

Table 1. Laser Scanner: Reigl VUX-1LR specifications. 12
Table 2. Inertial Measurement Unit: Applanix AP-20. 12
Table 3. Point densities from RiCopter VUX-1LR and VZ-2000 LiDAR systems. 66
Table 4. RiCopter flight altitudes and speeds during LiDAR collection. 70
Table 5. No-sink surface draining scan directions. 93
Table 6. LiDAR data collection events used in validation study. 105
Table 7. Damage detection summary for August 2016 – artificial airfield. 107
Table 8. Damage detection summary for December 2016. 108
Table 9. Damage detection summary for August 2017. 109
Table 10. Damage detection summary for December 2017 obscurant event. 110
Table 11. Damage detection summary for March 2019 (50ft-5m/s). 111
Table 12. Summary of crater detection for all RiCopter flight lines. 112
Table 13. Summary of camouflet detection for all RiCopter flight lines. 113
Table 14. Summary of spall field detection for all RiCopter flight lines. 113
Table 15. Damage detection summary for March 2019 night flight (50ft-5m/s). 114

Preface

This study was conducted for the United States Air Force Civil Engineering Center under Project Number 471376, "Airfield Damage Detection Using LiDAR Point Clouds," and Project Number 480403, "Automatic Pavement Damage Assessment using LiDAR." The technical monitor was Dr. Robert A. Diltz.

The work was performed by the Airfields and Pavements Branch (GMA) of the Engineering Systems and Materials Division (GM), U.S. Army Engineer Research and Development Center, Geotechnical and Structures Laboratory (ERDC-GSL), and the Field Data Collection and Analysis Branch (HNF) of the Navigation Division (HN), U.S. Army Engineer Research and Development Center, Coastal and Hydraulics Laboratory (ERDC-CHL).

At the time of publication, Ms. Anna Jordan was Chief, GMA; Mr. Justin S. Strickler was Chief, GM; Mr. William Butler was Chief, HNF; Ms. Ashley E. Frey was Acting Chief, HN; and Mr. Nicholas Boone, GZT, was the Technical Director for Force Projection. The Deputy Director of GSL was Mr. Charles W. Ertle II, and the Director was Mr. Bartley P. Durst. The Acting Deputy Director of CHL was Mr. Keith W. Flowers, and the Director was Dr. Ty V. Wamsley.

COL Teresa A. Schlosser was the Commander of ERDC, and Dr. David W. Pittman was the Director.

Executive Summary

Beginning in 2016, the U.S. Air Force initiated work with ERDC within their Rapid Airfield Damage Assessment System (RADAS) program to develop an expedient technology to identify, classify, and map damage features on a runway surface following a bombing attack. The ERDC chose a Light Detection and Ranging (LiDAR) system as its primary sensor to accomplish this task. LiDAR provides an immediate data product by representing the ground surface as a three-dimensional point cloud along with an intensity return that can be used as a filter for the detection of unexploded ordnance. LiDAR collection can be conducted during both day and night operations being independent of surface lighting conditions and can penetrate airborne obscurants such as smoke. For the detection of damage features, the optimum deployment of LiDAR is from a near normal perspective to the runway surface from an elevated position. ERDC began their research using a mobile LiDAR sensor suspended 50 ft in the air on a telehandler before deploying a LiDAR system mounted to an unmanned aerial system (UAS) flying at altitudes between 50 and 100 ft above ground level.

ERDC developed a computational methodology whereby LiDAR sensor data are converted into georeferenced point clouds (LAS files) during time of flight data collection incorporating the spatial positioning of the point cloud, the inertial state of the UAS, and the geospatial positioning of the sensor. The LiDAR collection software segments a scan into a series of smaller LAS files at a user-specified time period for easier data analysis. ERDC then developed a series of algorithms using MATLAB that converts the LAS files into meshes on which a depression-filling algorithm (DFA) is used to identify locations on the runway where penetration beneath the pavement surface has occurred due to detonation. Developed ERDC MATLAB algorithms then identify in a stepwise fashion the location, radius, and centroidal position of each crater, camouflet, and spall/spall field present on the runway surface. The detected damage is saved as a series of small shapefiles or data objects that can be displayed within the Geospatial Expeditionary Planning Tool (GeoExPT) used by the U.S. Air Force to display runway damage. GeoExPT is then used to calculate the Minimum Airfield Operating Surface from the location and extent of all damage features. This entire computational process has been automated such that the user's only responsibility is to activate the software, deploy the LiDAR sensor, and collect the resultant shapefiles.

LiDAR data were collected during a series of field-collection events from August 2016 until March 2019 at Silver Flag Airfield, Tyndall Air Force Base, FL. These data collections simulated a damaged runway using explosives to generate craters, spalls, and camouflets in the pavement surface. LiDAR point clouds were initially generated from a Riegl VZ-2000 mobile LiDAR sensor mounted on an elevated lift so as to simulate aerial data collection. Later, data were collected from a Riegl VUX-1LR LiDAR sensor mounted on a Riegl RiCopter UAS operating at multiple altitudes and ground speeds. The optimum deployment of the LiDAR system was found to be at 50 to 100 ft above ground level with a ground speed of approximately 5 m per second (mps). Refinements to the detection algorithms and the technology to automate and transmit the data analysis were developed during these data collections.

Utilizing data sets representative of aerial LiDAR, the developed DFA was able to correctly identify and classify 24 of 24 craters with no false positives, 9 of 9 camouflet holes with no false positives, and 7 of 7 spall fields with one false positive. The ability to initialize the MATLAB software a single time and provide results that are uniform across various test events and various data collection methodologies illustrates the robustness of the analysis system. Regarding time efficiency, the ERDC methodology was able to produce identified damage with a time lag of only 2 min behind the moment of LAS file creation in flight. This indicates that with a series of between 2 and 4 UAS-based LiDAR systems, the entire airfield could be scanned and processed within a 30-min time window as dictated in the RADAS objectives.

1 Introduction

1.1 Purpose

Military aircraft require relatively smooth, level surfaces on which to perform landing and takeoff operations. Typically these surfaces are paved, consisting of Portland cement concrete (PCC) and asphalt (AC), and can extend upwards of 10,000 ft or greater in length. These large paved surfaces often become high priority targets to adversaries in theater as their damage reduces air superiority. In the current research scenario, it is assumed that the pavement surface is rendered unusable through bombing attacks that create multiple craters of varying diameters and depths. By definition (AFTTP 3-32.11 2016), a crater is any depression that extends past the hardened pavement layer into the underlying base or subgrade material (Figure 1). A spall is a depression that extends only into the hardened layer (Figure 2). A camouflet is a punctured hole in which the explosive either did not detonate or detonated underground, and its severity is not readily observed on the surface (Figure 3).

Figure 1. Example of crater found in concrete pavement.



Figure 2. Example of spall field in concrete pavement.



Figure 3. Camouflet hole from subsurface detonation.



In order to bring the damaged airfield back to an operational status, a select number of craters must be repaired to create a minimum operating strip (MOS). In order to achieve this mission, three basic stages must occur.

1. Identify all damage including type and severity along with any unexploded ordnance (UXO).
2. Send out Explosive Ordnance Disposal (EOD) teams to remove any UXO.

3. Initiate construction operations to repair selected craters to establish an MOS.

The U.S. Air Force's Rapid Airfield Damage Assessment System (RADAS) program has been tasked to address stage 1 of the rebuilding mission. The goal of the RADAS program is to provide a capability to identify all airfield surface damage including type and severity as well as locate UXO across an airbase within a 30-min time window. Past research conducted in this area has focused on using aerial- and ground-based assets to provide rapid electro-optical (EO) visualization of the airfield enabling an expert to manually identify the damage from imagery alone (Filler 2011). However, this manual-visual approach, while functional and currently the state of practice, is not able to be completed within the 30-min time window. The RADAS program is now actively seeking out alternatives to this approach that meet the 30-min time requirement and provide sufficient accuracy that limits risk to EOD, airmen, and the mission.

1.2 Scope

The ERDC's approach to accomplish the identification stage in under 30 min required two fundamental changes to past research approaches. The first was to collect airfield surface data in a way that could rapidly be digitized into a three-dimensional scene, and the second was to develop and apply automated detection algorithms to the scene to identify the surface damage and UXO. The combination of these two technologies could provide the capability to identify, with sufficient accuracy, the necessary pavement damage to compute an MOS.

Taking advantage of recent advances in imaging technology, this research used Light Detection and Ranging (LiDAR) sensors to rapidly generate a digitized airfield surface. Over the course of this research, the deployment platform changed from ground-based vehicles (truck and telehandler) to an airborne UAS platform. Where a suitable UAS platform was unavailable, researchers outfitted a telehandler with a mobile LiDAR sensor in order to simulate airborne LiDAR. The LiDAR sensor created a high-resolution, geospatially positioned, dimensionally scaled three-dimensional point cloud from which numerical analysis could be performed.

MATLAB software was utilized as an interface allowing the development of algorithms necessary to identify and measure damage along with a graphical user interface (GUI) to evaluate the success of the software.

Algorithms were developed using filling techniques to identify the extent of each damage type. UXO identification required the use of LiDAR intensity returns to discern foreign objects from background material like pavement and blast ejecta. Location of all airfield damage and UXO was determined from the real time kinetic (RTK) global positioning system (GPS) integrated with the sensor platforms.

Furthermore, this research sought to minimize the time necessary to calculate the MOS by automating LiDAR data collection and damage detection algorithms. The automation process created data files populated with information containing damage type, location, and scale. In order to inform airmen of the current status of all airfield surfaces, these data files were uploaded to the Geospatial Expeditionary Planning Tool (GeoExPT) visualization software used during past RADAS development efforts. In this new scenario, all information that would have been manually determined by the airmen is now automated, requiring only a quality control check on the returned results using real-time aerial imagery.

1.3 Deliverables

This research project provides the software necessary to enable an automated damage detection capability beginning with data acquisition and ending with a series of data files containing location, description, and extent of airfield damage due to explosive ordnance. The process requires only an operator to collect a digitized point cloud, preferably LiDAR, and an operator to verify the accuracy of detection outputs when matched against visual imagery taken of the airfield from either static photographs or video. A MATLAB GUI is also provided that houses the developed detection algorithms. This GUI can be used for additional research on future airfield surfaces and sensor packages delivering point cloud information. More directly, this GUI can be used to adjust any applied filters to improve the damage detection algorithms; however, this GUI is left unseen to the user for typical mission usage and its parameters fixed to default values. The U.S. Air Force mandated that GeoExPT software be the final destination of detected airfield damage and, therefore, limited the form and extent of data output from the MATLAB software.

It is important to note that this technical report will focus primarily on the technology and algorithms developed to identify volumetric damage present on an airfield surface. A companion technical report will detail the technology and algorithms developed to identify the UXO present

on an airfield surface. The approach to solving each problem, volumetric damage versus object detection, varies considerably and will best be understood by the reader in a two-volume set. Data collection for both missions was conducted simultaneously, but development of the volumetric damage detection procedure is currently in a more mature state than that of UXO detection.

2 Methodology

Defining the specific airfield damage present on a runway following detonation is critical in determining an optimum sensor platform. The key damage typical on runway surfaces are as follows.

1. Crater – This is a cavity in the airfield surface that has penetrated all the way to the level of the base course where soil is exposed. A cratering munition can also generate an upheaval of the pavement surface surrounding the edge of the crater. This upheaval reduces the durability of the pavement structure and creates surface roughness unsuitable for aircraft gear.

2. Spall – This is a shallow cavity in the airfield surface that does not penetrate to the base course but is retained only in the hardened surface. Spalls tend to come in groups owing to the munition types that cause this type of damage.

3. Camouflet – This is an exposed hole in the pavement surface of narrow diameter where a munition has penetrated the surface and detonated below the surface. This can cause cavitation under the pavement along with upheaval or caving in of the surface depending on the depth of detonation.

4. Unexploded Ordnance – This can be any type of explosive munition exposed on the pavement surface or within a crater that remains live and can pose a threat to the repair team. These objects can be of diverse material types, typically metal, and can vary in size from a few inches to several feet in length.

The physical damage described above requires an analysis of depth to determine its classification, thus requiring a prior knowledge of the pavement thickness where the munition detonated. Unexploded ordnance can take on a variety of shapes and orientations, making automated detection difficult amongst the debris typical of a cratered scene.

Current approaches to assessing the pavement condition following a munitions attack require airmen to be present on site and to either drive a vehicle around the airfield at a safe distance from the runway or use high-powered binoculars from an elevated platform to visually estimate the size

and position of each damage type (AFTTP 3-32.11 2016). These techniques suffer from the following limitations.

- They put airmen at risk of injury from UXO or follow-on attacks.
- The approach yields poor quality data for detailed estimation of repair time and repair material quantities.
- Identification of UXO is difficult due to poor positional viewing of an airfield and large standoff positions.
- Visual identification is difficult when considering air obscurants and potential night-time attack.
- The process is slow, requiring several hours to complete.

To overcome these limitations, the following attributes must be present within the deployed system.

- Assessment must be remote such that no airmen are unnecessarily put in harm's way.
- Assessment must be aerial to minimize damage to the sensor package and to provide ease of movement across the airfield.
- Assessment must be capable of providing actionable data at night and in the presence of dust/obscurant fields.
- Assessment must be able to view the airfield surface with a high resolution such that small UXO can be identified.
- System must be accurate enough to identify the size and depth of craters for classification along with their geospatial location.
- System must be accurate enough to identify UXO down to a 3-in.-sized object and note the geospatial location.
- System should be simple enough that airmen without a background in sensor technology can operate the equipment effectively.
- System should automatically identify damage type, size, and location digitally without user interaction.

Given these attributes and guidance from the U.S. Air Force, it was decided that the following technologies would be explored.

- Deployment of sensors from a multi-rotor UAS.
- A LiDAR sensor package would be deployed that can collect a three-dimensional point cloud of the airfield surface for engineering analysis.
- A LiDAR sensor package would provide data for day or night operations and in dust or obscurant environments.

- GPS systems employed on the UAS and linked to the LiDAR system would provide geospatial positioning of the airfield damage.
- LiDAR can actively determine the dimensions on the airfield surface and UXO to within 2 cm or higher resolution from a 15-m height in single pass down the centerline of the pavement capturing data from shoulder to shoulder.
- LiDAR returns an intensity value on the laser reflection that can indicate the presence of UXO for further investigation.
- Software in MATLAB would be written to automatically identify craters, spalls, and camouflets from point cloud data noting their size and position.

2.1 Sensors

Many of the advancing technologies for remote assessment focus on the use of UAS-deployed EO-based cameras. Mosaics created from aerial imagery of the airfield and its associated damage are a valuable resource in visually assessing the condition of the airfield and providing a background on which to display locations of automatically detected damage. Several companion research efforts in the RADAS program focus on automated analysis of this imagery using pattern recognition derived from trained convoluted neural networks (CNNs) to identify airfield damage. In the early stages of the research, ERDC looked briefly at using the same aerial imagery to generate photogrammetry-based 3-D point clouds. What was discovered was that photogrammetry derived point clouds could not provide a satisfactory solution to the RADAS problem. The primary deficiencies were as follows.

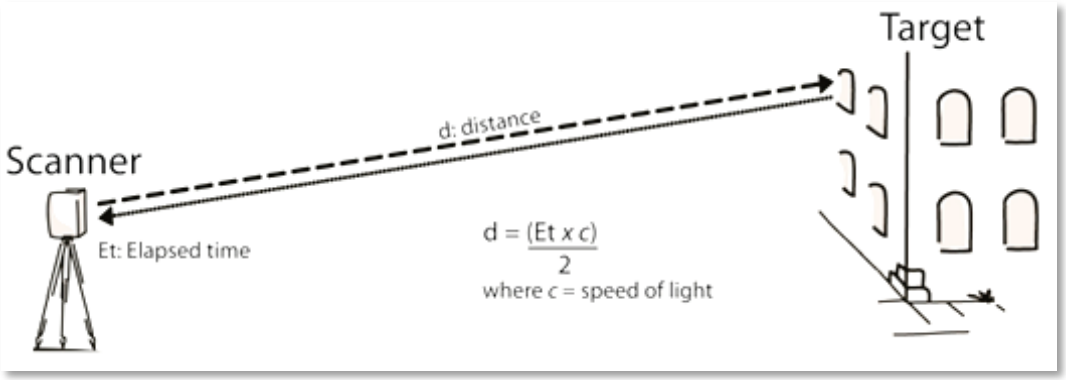
1. Processing time is too slow for the RADAS mission requirement. It takes hours to days to process the hundreds of photos necessary to derive point clouds of sufficient density to identify the small surface damage present on the airfield.
2. EO data collection is limited to times of day when lighting conditions provide enough contrast on the ground surface to resolve objects; therefore, this would not suffice in nighttime or dusty conditions.
3. Vertical accuracy of photogrammetry-based point clouds is poorer than those derived from LiDAR, giving rise to the presence of phantom objects on the ground surface that would increase the number of false positives detected.

Given these limitations, at the present level of photogrammetry-based 3-D reconstruction techniques, a LiDAR solution was chosen to meet the time and accuracy requirements demanded of the RADAS mission requirements.

2.1.1 Terrestrial LiDAR

LiDAR is a remote sensing technology that uses a pulsed laser light to measure the distance to a target surface by measuring the time of return of reflected pulses off the object's surface (Figure 4). A LiDAR system has a GPS attached to it so that each data point has a real-world location, allowing for engineering analysis to be performed on the developed point cloud. The mobile LiDAR system employed in this research introduces an inertial measurement unit (IMU) that determines the position of the laser with each pulse as a vehicle or aircraft moves with the laser system to correct the GPS position for movement of the sensor. When all these systems are integrated, a dimensionally and positionally accurate three-dimensional point cloud is immediately generated.

Figure 4. Illustration of LiDAR pulse and return.



The terrestrial LiDAR system employed in this research was a Riegl VZ-2000 mobile LiDAR unit and was used as the primary data collection sensor from 2016 to 2018 and used as a backup in 2019. It was mounted in two configurations for this research. Initially, it was mounted in a horizontal configuration on the rear of a pickup truck to enable rapid movement down the runway (Figure 5). The horizontal position of the scanner provided a non-symmetric, 130-deg measuring swath transverse to the rear of the vehicle. The truck-mounted LiDAR approach allowed for rapid movement along the airfield surface at speeds of 10-30 mph;

however, to avoid scanning the edges of the mounting frame, the LiDAR had to be tipped 20 deg outward from the rear of the truck creating an oblique angle to the ground surface. This oblique angle coupled with the low altitude made it difficult to resolve large crater damage because no data could be collected at the deepest points of the craters, and the angle caused fewer points to be collected in small, deep camouflets in the pavement surface. Further, the low height (< 10 ft) at which the sensor was deployed required multiple passes along the airfield surface to collect a complete point cloud of the entire runway. Stitching of LiDAR from multiple collection runs requires extensive manual post processing to eliminate variances in point spacing due to GPS error and, therefore, is not a useful approach for the time-sensitive RADAS mission.

Due to the lack of a UAS capable LiDAR system at the time of the research, it was decided to mount the VZ-2000 on a cage and lift it with a telehandler to a height of approximately 50 ft above the ground surface to simulate UAS deployment and cover the runway surface in a single swath (Figure 6). The telehandler-mounted LiDAR worked especially well, providing a wide area of coverage and data nearly normal to the pavement surface that allowed resolution of deep craters and camouflets. The drawback was that the system had to be driven at slow (< 5 mph) speeds to prevent the frame from tipping over. This slow movement with a high-density LiDAR system collected point clouds that were far denser in points per square meter than what could be obtained from a UAS-based system. Therefore, this system was limited in its ability to assess how well objects would ultimately be able to be detected from an airborne platform.

Figure 5. Reigl VZ-2000 LiDAR scanner mounted on vehicle.



Figure 6. Reigl VZ-2000 LiDAR scanner mounted on cage on telehandler lift.



2.1.2 Airborne LiDAR

Beginning in the fall of 2018, ERDC was able to deploy the Riegl RiCopter UAS LiDAR system (Figure 7). The Riegl RiCopter platform hosted the Riegl VUX-1LR mobile LiDAR sensor. The VUX-1LR uses the same scanning pattern and contains the same IMU, an Applanix AP20, as the VZ-2000 but generates a lower overall point density owing to power and size constraints. This system was used in two data collection events to determine the ability of a UAS system to detect surface objects with a reduced point cloud density. The details of the VUX-1LR and Applanix AP20 sensor are given in Table 1 and Table 2, respectively.

Table 1. Laser Scanner: Reigl VUX-1LR specifications.

Parameter	Specification
Laser Wavelength	1550 nm
Target Operating Altitude	Varied (50 - 100' predominantly)
Accuracy	15 mm
Precision	10 mm
Pulse Repetition Frequency	880 kHz
Effective Measurement Rate	750 kHz
Scan Angle	360 degrees
Beam Divergence	0.5 mrad

Table 2. Inertial Measurement Unit: Applanix AP-20.

Parameter	Specification
Weight	0.68 kg
Size	85 mm x 85 mm x 66 mm
Embedded GNSS	L1/L2, GPS, GLONASS, BeiDou, Galileo, QZSS, SBAS, L-Band
Power	5 v DC
Output refreshing rate	200 Hz
Time tagging	PPS Output
Heading accuracy (PPK)	0.035 degrees
Roll and pitch accuracy (PPK)	0.015 degrees
Horizontal Accuracy (PPK)	< 0.05 m
Vertical Accuracy (PPK)	< 0.1 m

Figure 7. RiCopter with LiDAR scanner, GNSS/INS navigation system, and RGB cameras attached to the belly of the aircraft.



During the data collection events, the RiCopter was flown at varying altitudes from as low as 50 ft above ground level (AGL) to as high as 400 ft AGL. The RiCopter was also flown at two primary speeds, 5 m/s and 10 m/s, with the former being about the slowest the drone can fly and maintain an accurate flight path and the latter the preferred flight speed of the drone for maneuverability.

The RiCopter required a team of two personnel to conduct operations (Figure 8). The team would assemble and prepare the drone for flight, ultimately with one individual as the pilot and the second as the visual observer (Figure 9). The visual observer is tasked with monitoring flight statistics on a mobile laptop with the installed mission planning software (Figure 10). Flights were conducted with pre-planned flight missions so that the data collection occurred in an automated fashion, but the pilot reserved control of the UAS from flight sticks to ensure safety if any deviations from the flight plan occurred.

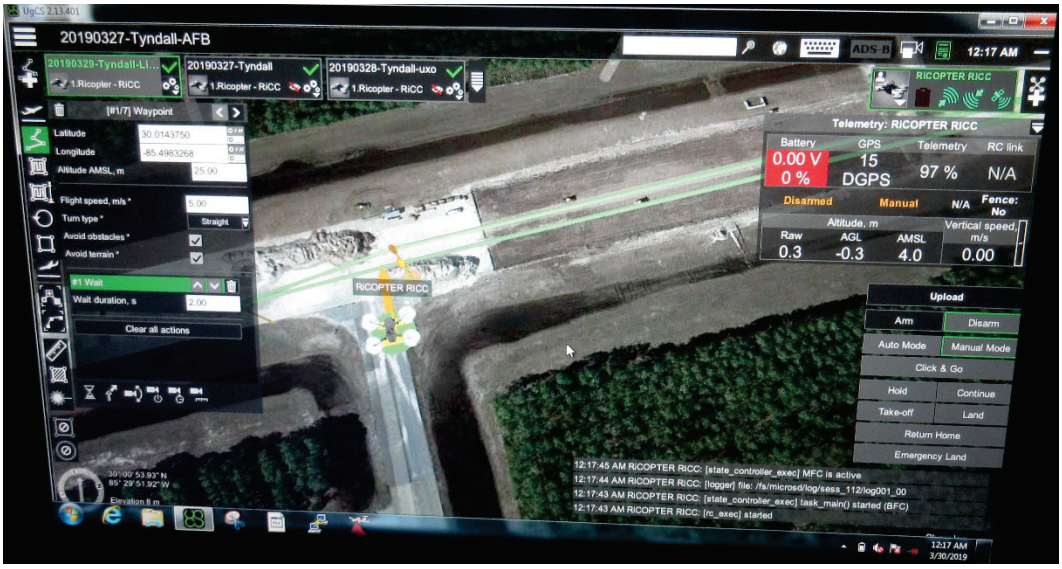
Figure 8. Two operators evaluating RiCopter prior to flight.



Figure 9. Pilot and observer for Ricopter. Base station antennae in foreground; Ubiquiti data transfer antennae in background.



Figure 10. RiCopter mission planning software outlining flight path along Silver Flag runway from launch point at the south taxiway intersection.



2.2 Analysis software

2.2.1 RiAcquire and RiWorld

The Reigl sensor packages used in this research effort required data acquisition and reduction using their manufacturer's proprietary software. LiDAR point data were acquired from the sensors using RiAcquire. The generation of a georeferenced point cloud in a LASer (LAS) file format was accomplished with the use of RiWorld, which transformed the scan data into the coordinate system of the trajectory, usually WGS84 or UTM, by integrating the GPS position of the points and the IMU position of the sensor with the sensor data. Once a LAS file was generated from these software packages, it could then be viewed and analyzed with a number of commercial software systems capable of ingesting this data type. One benefit derived from using the combined Riegl software suite is that the user can synchronize both RiAcquire and RiWorld internally on the data acquisition computer, allowing data collection and LAS file generation to occur automatically without any intermediate action.

2.2.2 MATLAB

Analysis of point clouds generated from the LiDAR sensors was performed using the MATLAB development environment. MATLAB has a native capability to read in the .LAS file format and contains multiple applications that can perform calculations on point clouds. Customization of these applications was made to meet the specifications of RADAS. One series of

volumetric calculations on the point cloud identifies any significant surface damage based on a measurement of surface depressions and classifies them according to the crater, spall, and camouflet criteria. A MATLAB GUI was designed to allow visualization of this procedure (Figure 11), and the various output files it generates are shown in Figure 12. During automated processing, the locations and classification of any volumetric damage are recorded in output files, and the visualization GUI is not required.

Figure 11. MATLAB software application to detect crater locations.

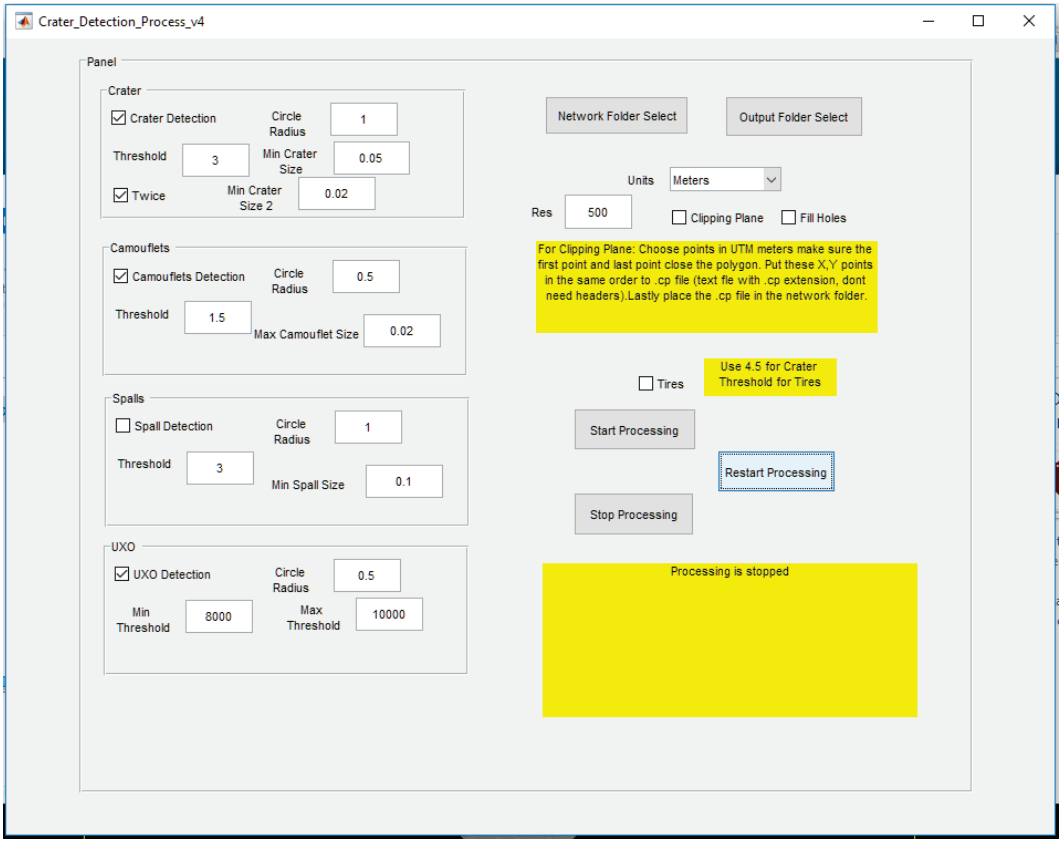
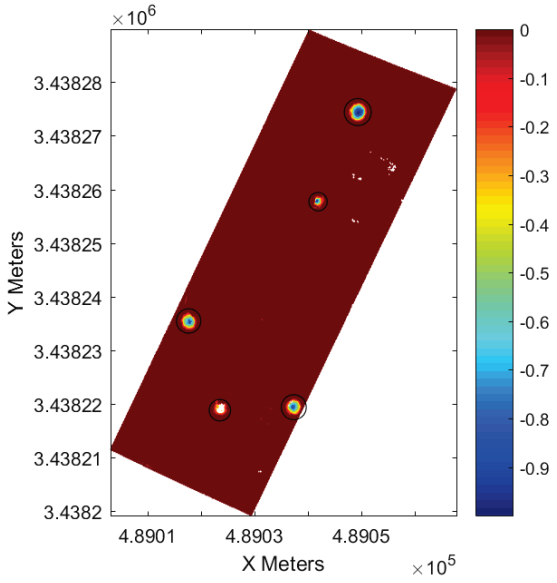


Figure 12. Modern version of MATLAB software showing crater detection and elevation scale; blue is below ground up to red, which is at the surface of the runway.

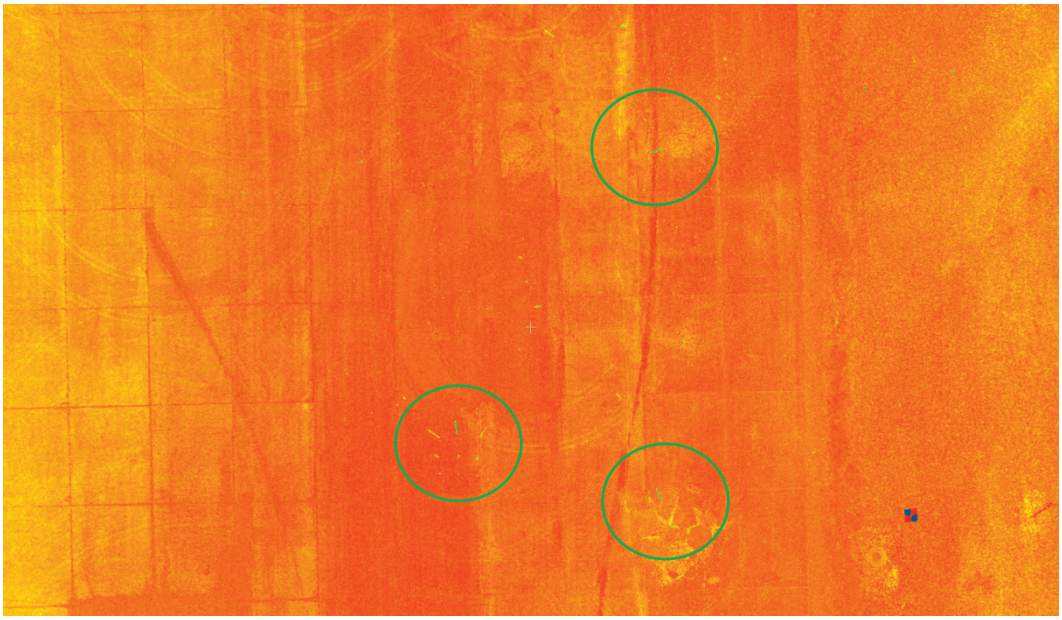


A second set of calculations is conducted on the intensity magnitude originating from the return signatures of each LiDAR point. The value of these intensities can provide an indication of metallic (higher reflectance) or non-metallic (lower reflectance) objects. This can act as a filter to identify the presence of UXO. Figure 13 shows an aerial photograph of a section of airfield with UXO scattered throughout the scene. In Figure 14, the same image is recolored according to intensity magnitude from the LiDAR returns. The UXO in green can be identified from the concrete in yellow and orange.

Figure 13. Scene of Silver Flag airfield with various UXO located throughout.



Figure 14. Scene of Silver Flag airfield on an intensity-scaled airfield with UXO identified as green intensity and orange and yellow as concrete.



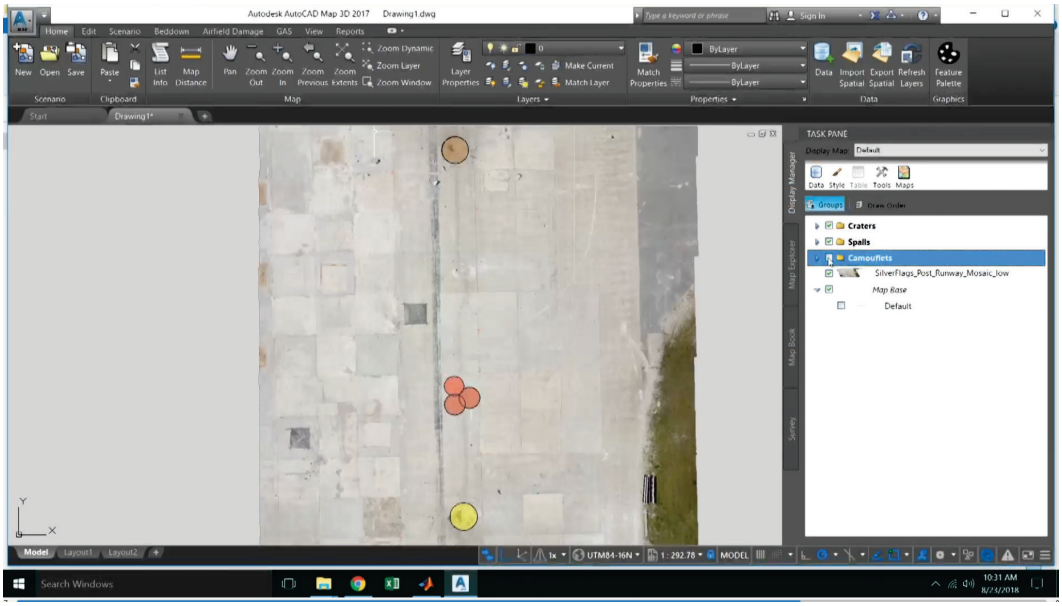
The MATLAB algorithm is designed to search for new LAS files generated by Riegl software. Once the software locates a new, finished LAS file, it begins volumetric processing. The result of the team's volumetric algorithms is automatically output to a user-accessible folder, allowing for damage detection to be done without user knowledge of LiDAR data or MATLAB software.

2.2.3 Geospatial Expeditionary Planning Tool

The Geospatial Expeditionary Planning Tool (GeoExPT) is a decision-support software tool for mission planners and engineers. This software provides the means to create geospatially accurate base layout plans, meet bed-down requirements, automate aircraft parking, and analyze and repair airfield damage for optimal selection of the Minimum Airfield Operating Surface (MAOS) in accordance with Department of Defense (DoD) standards. For the purposes of the RADAS program, GeoExPT has been used as a GUI to import images from aerial reconnaissance, allow manual selection of damage from drop-down menus, and perform the calculations required to identify the MAOS once all physical damage and UXO have been identified on the airfield map.

The results of the ERDC developed MATLAB software are a series of data files consisting of damage location, type, and size. This information is displayed via GeoExPT as a series of colored icons within the viewer window so that the user can provide a visual quality control check on the accuracy of the identified damage. GeoExPT takes all identified damage and UXO and determines the MAOS necessary to define airfield repair operations. Figure 15 shows a screenshot of the GeoExPT GUI with crater and spall damage highlighted on top of a photographic mosaic.

Figure 15. Screenshot of GeoExPT GUI with crater and spall damage overlain on a mosaic of the damaged airfield.

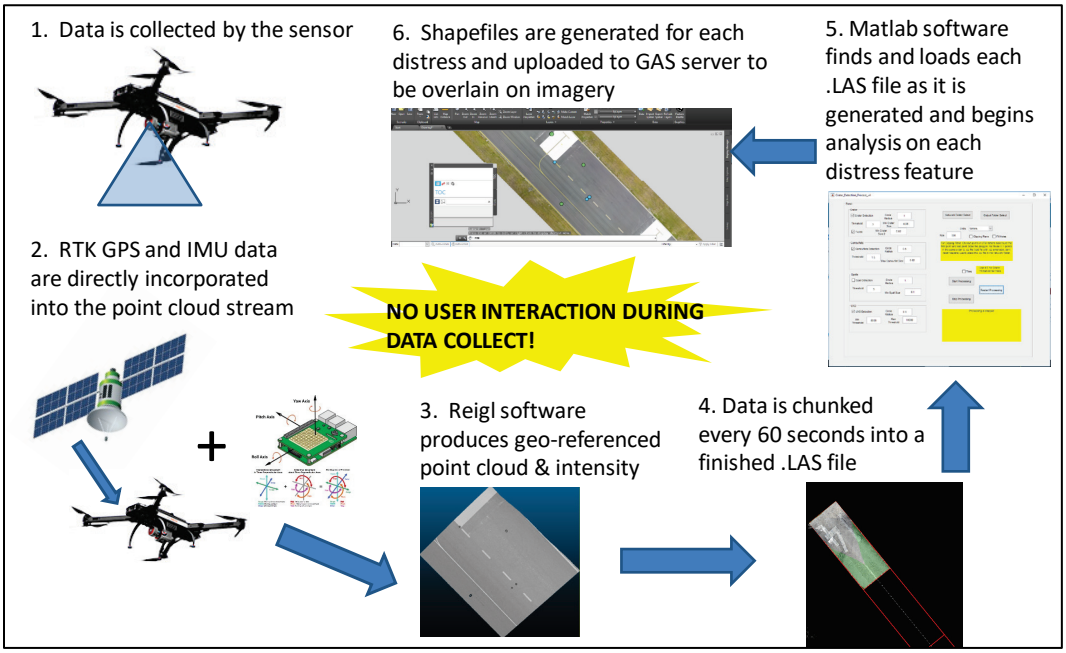


2.3 Integration

The goal of this research is to produce an automated analysis system in which an airman does not need to know the particulars of how each individual software package functions. All that is required of a field operator is to launch a UAS with an attached LiDAR sensor, initiate the software to begin data collection and processing, and receive processed data of interest. An illustration of this process is shown in Figure 16. In this diagram, the first step is initiating the software and launching the UAS in a pre-programmed flight path requiring little hands-on direction by the pilot. All subsequent activities are hands off to the user. The next phase of the process is data collection by the onboard LiDAR, IMU, and GPS sensors which, using RiAcquire and RiWorld, will create a geo-referenced point cloud from which analysis can begin. The next step is to write out a finished LAS file from this continuous

data collection every 60 sec to keep LAS file sizes manageable. Over the course of a mission, 10 to 20 LAS files must be analyzed in MATLAB. The advantage in MATLAB is that these files can be analyzed in parallel if analysis of a particular file takes longer than the 60-sec collection increment. Once the algorithms have detected volumetric damage and UXO, this information is recorded in a small file that can be remotely transmitted to the operator at a ground-based computer to populate the GeoExPT interface for quality control and MAOS selection.

Figure 16. Diagram of the ERDC automated LiDAR data processing for RADAS.



3 Data Collection

3.1 Test location

Examples of varying airfield damage types were obtained from three primary collection events all occurring at the Silver Flag airfield at Tyndall Air Force Base (Figure 17 and Figure 18). Silver Flag is a non-operational concrete runway used for training events on airfield repair operations for a variety of military scenarios. Silver Flag has a dedicated section of runway where live cratering events are conducted and where data were obtained for this research (Figure 19).

Figure 17. Silver Flag airfield located southeast of Tyndall Air Force base.



Figure 18. Highlighted area is south end of Silver Flag runway used for crater testing.

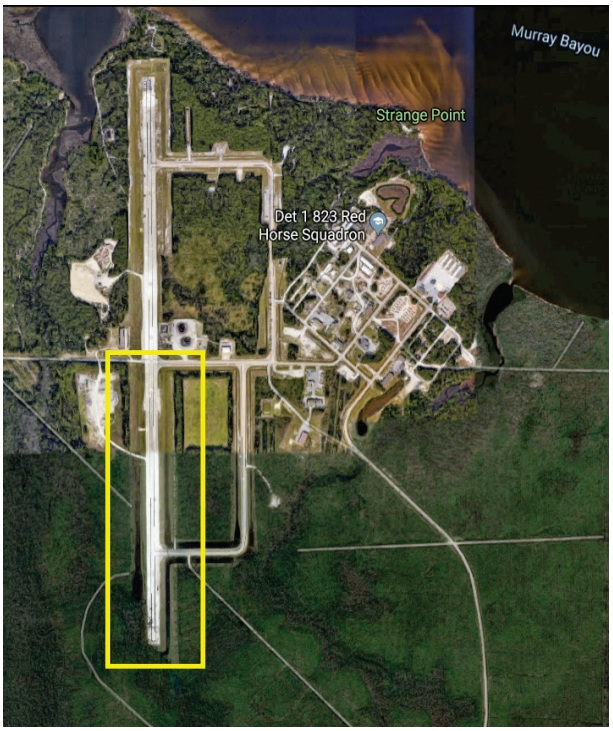


Figure 19. Silver Flag runway looking south where cratering operations occur.



3.2 Single crater event (August 2016)

In August 2016, a single large crater was created at Silver Flag to provide an initial data collection opportunity for the RADAS team (Figure 20). No spalls or camouflets were detonated at this event. The crater was created using a buried explosive charge placed within a 6-in.-diam cored hole in the concrete. This process was repeated throughout subsequent test events. This was the first data collect at Silver Flag in which ERDC participated as a member of the RADAS team. At the time, ERDC was still experimenting with the idea that either LiDAR or photogrammetry could be used as a possible solution to the RADAS mission and so used both technologies on site.

Figure 20. Single crater event at Silver Flag.



3.2.1 Airborne imagery collection

The ERDC team was equipped with two UAS's. i.e., a DJI Phantom 3 and the Turbo Ace Matrix-E. The Matrix UAS was equipped with aSony A6000 24 megapixel (MP) camera (Figure 21) to provide data collection normal to the pavement surface from an elevated position. The Phantom collected 4K high-definition video and 12.4MP still shots, while the Matrix-E collected only still images at 24MP resolution. The imagery was

taken at an altitude of 50 ft to maximize pixel resolution on the ground surface. Collected images were stitched together into a three-dimensional point cloud using the Agisoft Photoscan processing software (Figure 22).

Figure 21. DJI Phantom 3 (top left) and Turbo Ace Matrix E-UAS with Sony Alpha camera (right).



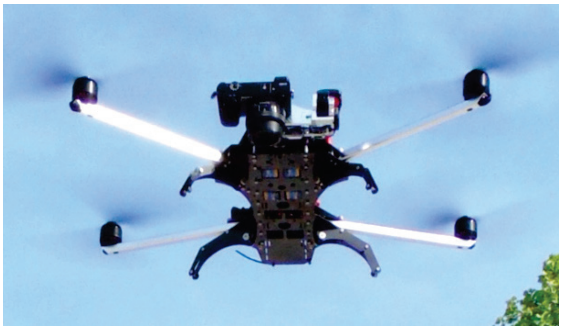




Figure 22. Point cloud reconstruction from Silver Flag crater (Matrix-E imagery); flags highlight specific UXO locations.



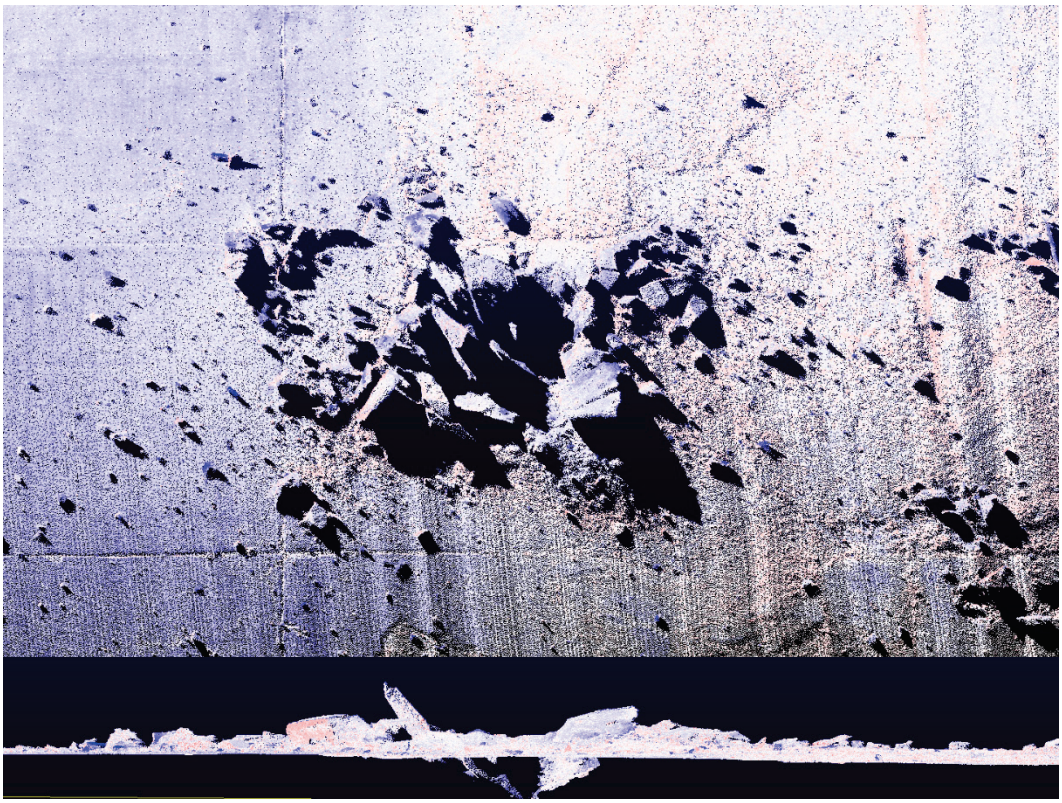
3.2.2 Ground-based LiDAR collection

For LiDAR data collection, ERDC deployed a Reigl VZ-2000 mobile LiDAR system mounted on the rear of a truck (Figure 23). The Reigl LiDAR system was mounted at a height approximately 8 ft above the ground surface and pointed 20 deg outwards from normal to prevent interference with the scan images from the device housing. The truck drove along the perimeter of the debris field outside the crater area to avoid damage to the vehicle during data collection. The 130-deg side-to-side sweep angle of the laser enabled data to be collected from a low elevation across the area of the crater as the vehicle traversed along the edges. However, the low grazing angle of the sensor did not allow for penetration of the LiDAR into the center of the crater to establish crater depth (Figure 24).

Figure 23. VZ-2000 LiDAR mounted to rear of truck to provide surface scan of crater.



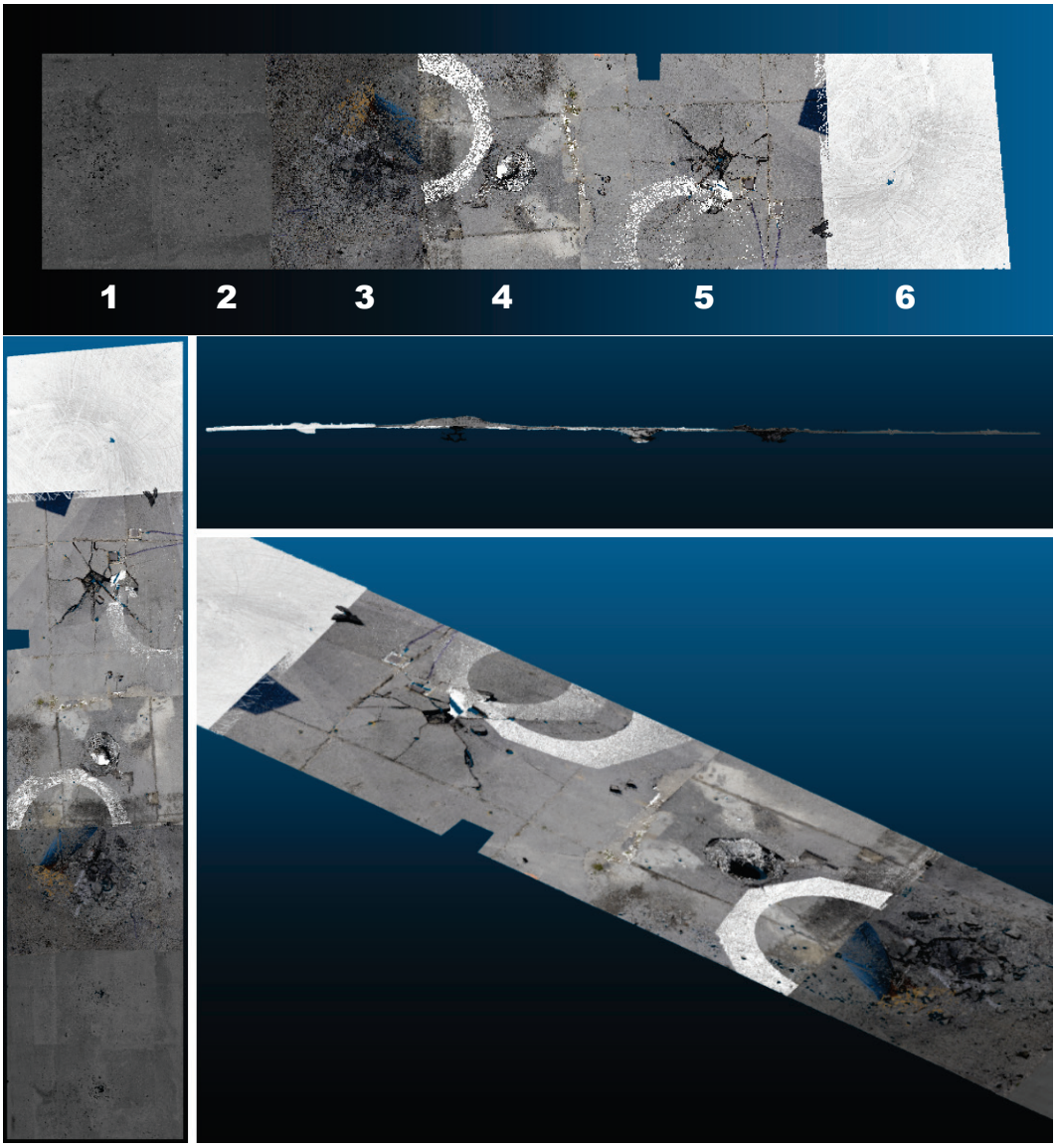
Figure 24. LiDAR scan (intensity view) of crater at Silver Flag.



3.2.3 Scene creation with historical LiDAR scans

For initial software development, a single crater did not provide a sufficient test bed to evaluate a multiple crater scenario. Prior to ERDC joining the RADAS research effort, the Air Force had conducted a series of single crater events and scanned the blast areas with a static, tri-pod mounted LiDAR system. ERDC decided to artificially create a multi-crater scene using the single Silver Flag blast and combining it with the prior single crater events. Five additional crater LiDAR scans were pieced together with the Silver Flag capture such that the edges of each scan were aligned vertically to appear as a uniform flat surface with varying crater sizes along its length and the varying areal geometries cropped to create uniform runway edges to represent a standard rectangular airfield surface (Figure 25). The colorization and intensity magnitudes vary from scene to scene, owing to the different times of day and seasons in which the data were collected; however, the volumetric consistency between scans was preserved in the scaled LiDAR LAS files.

Figure 25. Artificial 6-crater airfield from independent LiDAR scans (top) and detail of central section (bottom).



The initial software development in MATLAB to determine the location of craters based on volumetric changes in the surface was based on this artificial data set and then later validated on the subsequent six crater event that took place in December 2016.

3.3 Six crater event (December 2016)

In December 2016, a multi-crater exercise was conducted to produce a more realistic scene of airfield cratering and to provide a demonstration of the most current RADAS capabilities to assess airfield damage. This event created six individual craters along with two sets of spalls (three spalls in a

set) and two camouflets along with various UXO distributed throughout the blast area following the detonations. The ERDC team was equipped with an identical set of equipment to the August 2016 exercise, i.e., truck-mounted Reigl VZ-2000 mobile LiDAR, Matrix-E quadcopter, and the DJI Phantom 3 UAS. ERDC had three primary goals at this test event.

1. Collect aerial EO imagery and mobile ground-based LiDAR and rapidly develop three-dimensional point clouds using either photogrammetry or the processed LiDAR.
2. Apply the newly developed software detection algorithms in a real-time scenario to the developed point clouds to identify the locations of any craters, spalls, or camouflets.
3. Report real-time results to AFCEC sponsors in order to demonstrate the team's success in rapidly identifying volumetric damage through point cloud analysis.

This event was ERDC's first opportunity to demonstrate the combined technology of LiDAR point cloud collection with the MATLAB analysis tools that could survey the point cloud and identify volumetric damage. To this point in the RADAS research program, only image-based detection approaches had been attempted. These approaches either utilized manual selection of damage by the airmen or convolutional neural network algorithms to automatically highlight regions of interest where damage might reside.

3.3.1 Airborne imagery collection

The ERDC team was equipped with the same set of UAS platforms used in the single crater event data collection, i.e., the DJI Phantom 3 and the Turbo Ace Matrix-E (Figure 21). Imagery was taken at a similar altitude of 50 ft to maximize pixel resolution on the ground surface. The images were stitched together using the Agisoft Photoscan processing software. Figure 26 shows the resultant image mosaic and the three-dimensional point cloud generated.

Figure 26. Mosaic of 6-crater event at Silver Flag from Matrix photographs (top) and reconstructed point cloud from Matrix-E imagery (bottom).



3.3.2 LiDAR image collection

For LiDAR data collection, ERDC deployed a Reigl VZ-2000 mobile LiDAR system mounted on the rear of a truck as performed in the August 2016 data collection (Figure 23). The Reigl LiDAR system was mounted at a height approximately 8 ft above the ground surface and pointed 20 deg outwards from normal to prevent interference with the scan images from the device housing. The truck drove along the perimeter of the debris field outside the crater area to avoid damage to the vehicle during data collection. The 130-deg side-to-side sweep angle of the laser enabled data to be collected from the low elevation across the area of the crater as the vehicle traversed along the edges. Figures 27, 28, and 29 show processed

LiDAR scans obtained for analysis using the VZ-2000 sensor. Data was collected at a high, 1000-MHz frequency rate to ensure good point density from the LiDAR sensor.

Figure 27. Complete LiDAR image of 6-crater event.

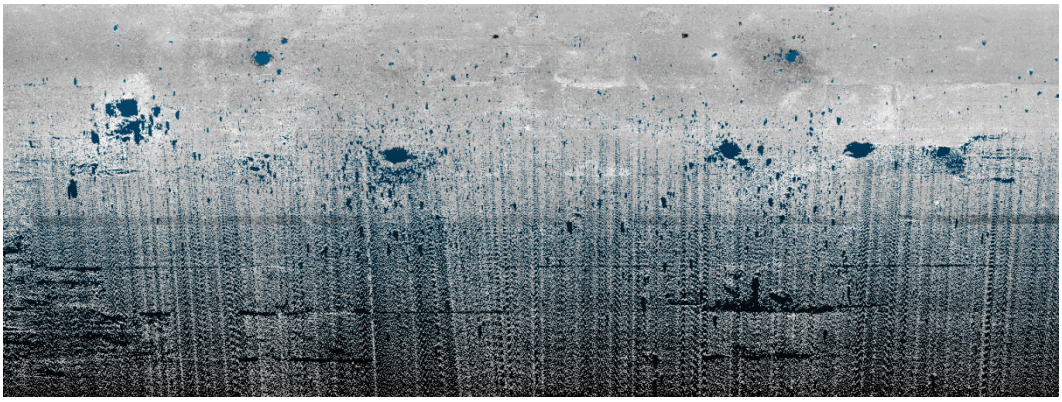


Figure 28. LiDAR closeup of 3-crater and 2-spall set on south end of crater event.

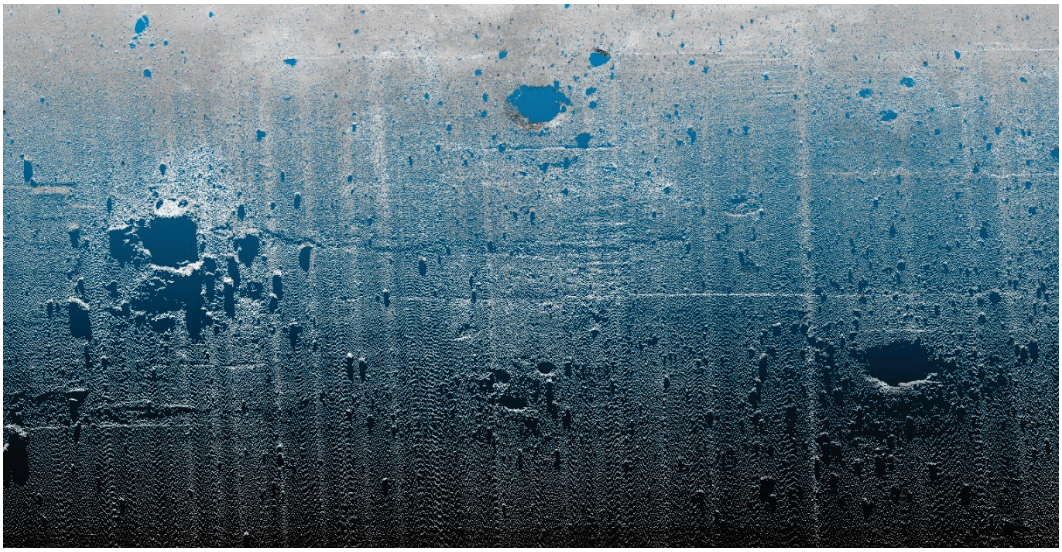
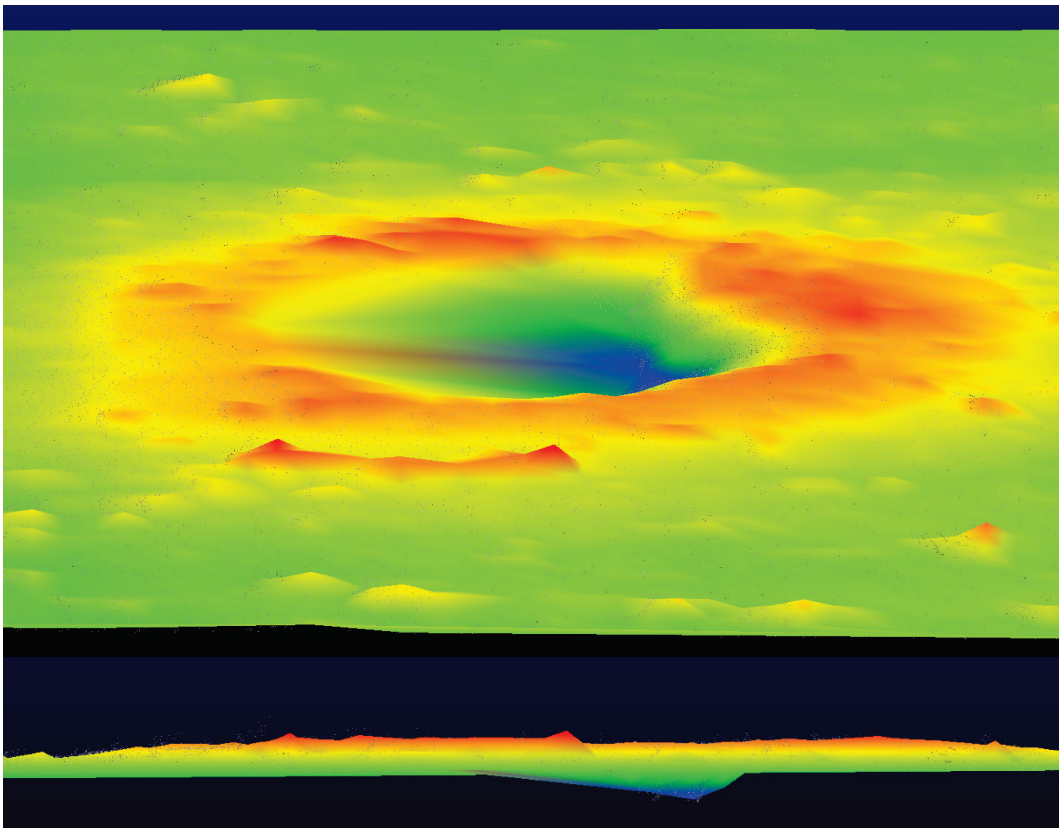


Figure 29. LiDAR closeup of three craters and one camouflet on north end of crater event.



Upon closer inspection of the single crater in Figure 30, it can be seen that minimal penetration of the LiDAR beams into the crater occurs such that only depth on one side wall of the crater is observed. This is due to the low grazing angle of the LiDAR scanner. At such an oblique angle, the LiDAR beams are unable to penetrate the depth of the crater due to upheaval and debris. This problem was resolved by utilizing an aerial LiDAR system in subsequent data collections. Aerial LiDAR data collection provides a normal view into surface damage and is therefore immune to interference by debris and upheaval.

Figure 30. LiDAR mesh of crater at Silver Flag from angle (top) and side (bottom).



3.4 12 crater event (August 2017)

In August of 2017, a large cratering event was conducted at Silver Flag, which involved the creation of 12 craters, four sets of spalls (three detonations each), and three camouflets. The craters were distributed over a much wider area than any of the previous exercises providing a more challenging environment for the data collection teams. The distribution of damage is shown in Figure 31 and Figure 32 along the length of the Silver Flag runway. The intent of this data collection exercise was to provide the RADAS team with a large-scale blasting event whereby a number of instances of each damage type could be evaluated.

Figure 31. North third of test section (left), central third of test section (right) at Silver Flag – 12-crater event.



Figure 32. South third of test section at Silver Flag – 12-crater event.



3.4.1 Airborne EO and thermal image collection

The ERDC team was equipped for this data collection with two sUASs, both Turbo Ace Matrix-E systems. One Matrix-E carried the Sony A600 camera for acquiring 24 MP still images. The ERDC team had decided from prior data collection events that photogrammetry could not yield point clouds in the compressed timeframe demanded by the RADAS mission. Therefore, ERDC implemented future EO imagery solely for the creation of aerial mosaics for use in overlaying damage features within GeoExPT and for internal reference.

On the second Matrix-E sUAS, ERDC deployed a FLIR Tau-2 Infrared camera with still images captured with a 640 × 480pixel resolution (Figure 33). The intent of the thermal camera was to determine if changes in thermal radiation from UXO could be picked up with this sensor package. Flights were flown at an elevation of 50 AGL both during the middle of the day and at dawn when the thermal crossover was occurring (cool ground temperatures, but warming UXO from sunlight). The Tau-2 returned real-time video to an observation screen (Figure 34) to allow a quick visual determination of the success of the data collect. However, as was discovered during a prior thermal collect at ERDC, even during the

thermal crossover periods in the early morning, UXO thermal signatures could not be distinguished from the background. This is primarily due to the low pixel density of the sUAS thermal sensors. UXO tended to be smaller than an individual pixel when captured from a 50-ft altitude. The use of the 50-ft height was to ensure shoulder-to-shoulder coverage across the runway in a single pass so that only a single pass of the drone down the runway was required to achieve the compressed RADAS timeframe.

Figure 33. Launch of Matrix-E with FLIR Tau-2 thermal camera and base station.



Figure 34. Real-time display of thermal imagery from Matrix-E drone during flight.



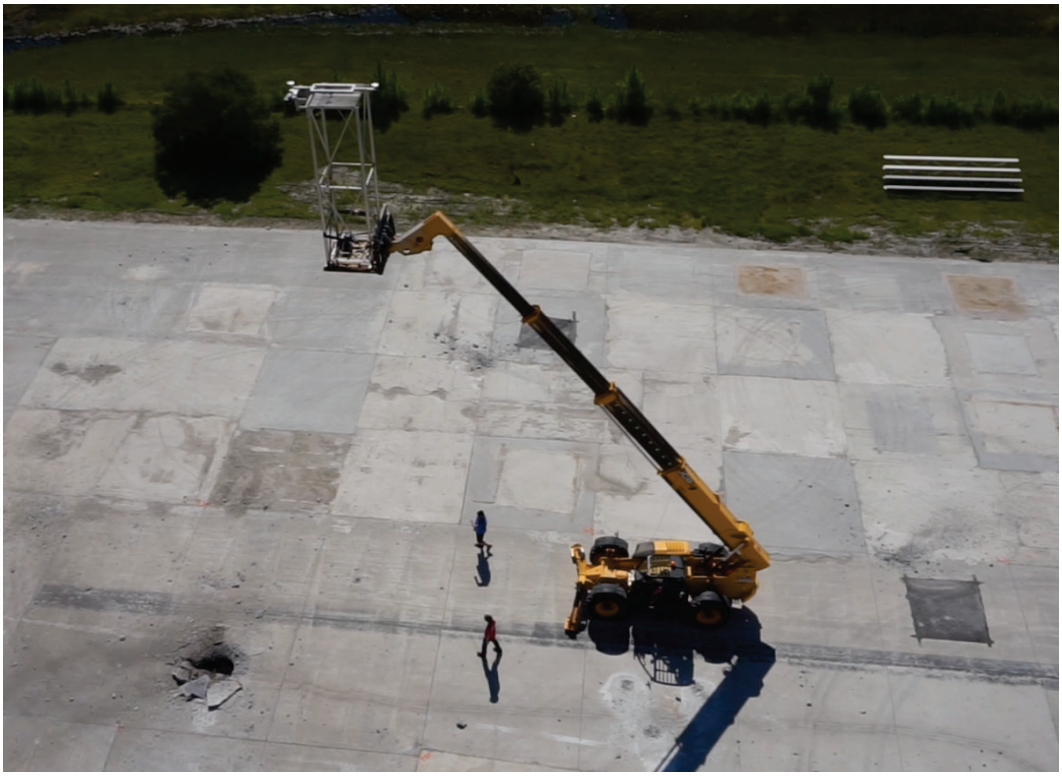
3.4.2 LiDAR setup with telehandler

A new approach was introduced at this data collect whereby the mobile VZ-2000 LiDAR unit that was previously truck mounted was now mounted to a 10-ft-high aluminum cage and lifted on the forks of a telehandler that elevated the LiDAR unit to between 50 and 60 ft above the ground. This simulated a LiDAR data perspective that could be achieved by a UAS-based LiDAR system. Such a system was not ready for deployment at the time of this data collection. The telehandler was then driven across the test site to capture the damage from varying angles at speeds between 2 and 3 km/hr. The elevated position of the sensor enabled significant penetration of the LiDAR beams into the surface damage greatly improving the accuracy of detection and classification as was hypothesized at the start of the ERDC research. The frame system used for placement of the VZ-2000 is described in Figure 35 and shown in Figure 36.

Figure 35. Assembly of LiDAR on aluminum frame for telehandler: Aluminum frame (top left), securing VZ-2000 on custom aluminum mount and stringing 100-ft Ethernet cables to telehandler (top right and lower left), and final assembly with generation mounted to pallet secured to bottom of aluminum frame (bottom right).

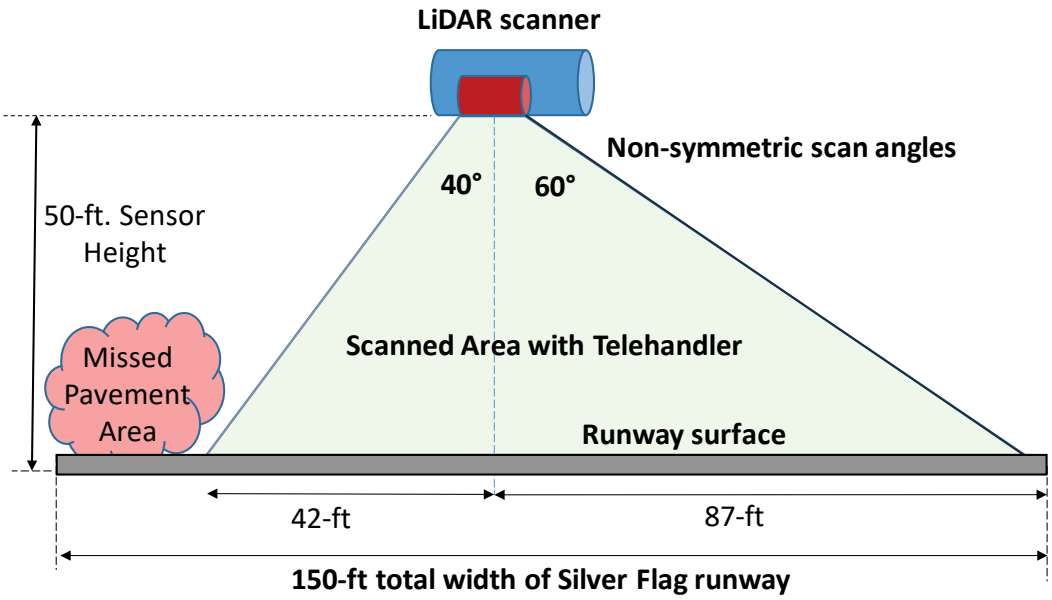


Figure 36. VZ-2000 mounted to an aluminum cage hoisted on a telehandler lift providing a sensor height of approximately 50 ft above ground level.



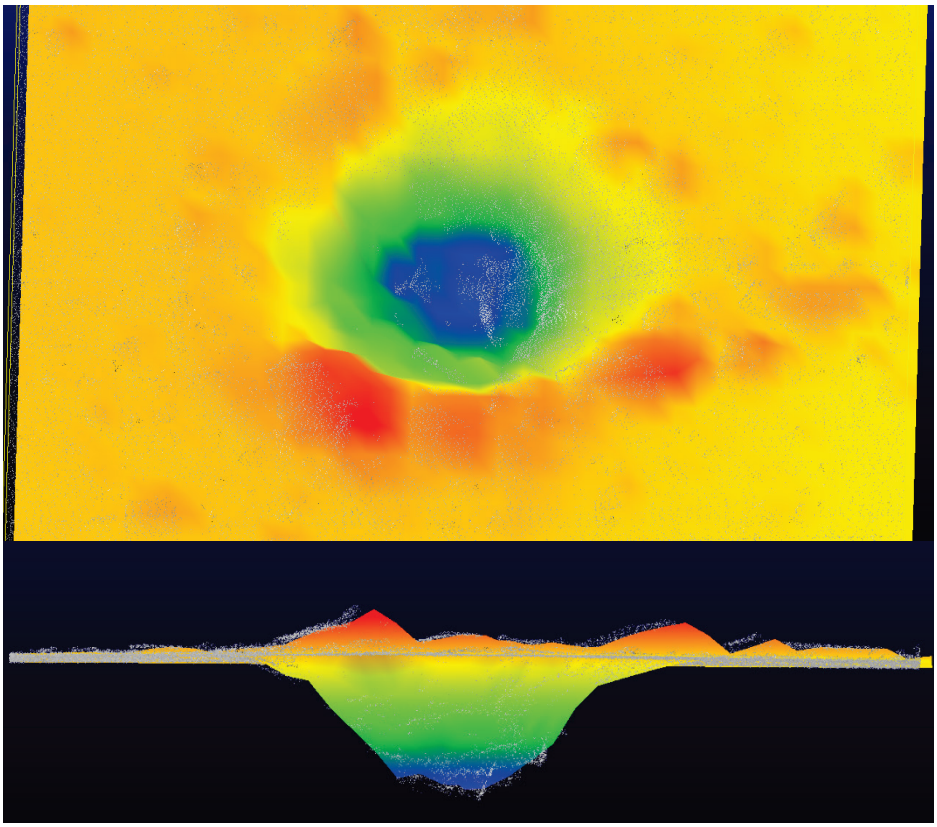
Data were collected using several passes at this test event. An attempt was made to perform a single-pass LiDAR data collect by driving the telehandler down the center of the runway. Such a pass would simulate the path of a UAS. However, several overhead water lines mounted to telephone poles were suspended over the southern test location (Figure 32) at 30- to 40-ft elevation, which required lowering the telehander boom to drive underneath them changing the point density along the route. Furthermore, the VZ-2000 is equipped with a total of 100° of scan angle that, when turned horizontally, only permits a partial scan of the runway surface from the 50-ft elevation, the maximum allowed by the telehander (Figure 37). Because of the uneven distribution of scan angle with 40° to one side and 60° to the other side, a single telehandler pass was unable to cover the runway from shoulder to shoulder, requiring several intermediate runs. Therefore, four passes with the LiDAR were conducted, one for each 3-crater set located along the runway. Each pass was driven as close to the craters as possible, being just outside the debris field so as not to destabilize the telehandler. The intent of the close runs was to obtain as normal a LiDAR point cloud as possible to simulate ideal UAS conditions.

Figure 37. Scan angle limitations of the VZ-2000 Mobile LiDAR scanner.



The nearly normal perspective of the LiDAR provided significantly improved point clouds with very little missing data within the damage features. Figure 38 demonstrates the mesh now generated from a dense point cloud indicating a significant depth of LiDAR penetration into the crater enabling creation of a smooth surface contour of the crater walls for classification and volume measurement.

Figure 38. LiDAR point cloud mesh of crater from telehandler perspective.



3.5 Obscurant event, Fort Polk (December 2017)

In December of 2017, a test event was conducted at Fort Polk, LA, to determine the ability of a LiDAR system to identify surface damage through a smoke obscurant. The test was designed to simulate the condition of an airfield immediately following a bombing attack whereby considerable dust and obscurants are in the air making EO visibility difficult. Three separate craters were detonated across a square 9-slab configuration as shown in Figure 39.

Figure 39. Colorized LiDAR scan of obscurant test section at Fort Polk with three craters and closeup of one of the crater blasts.



3.5.1 LiDAR data collection

Data were collected at the test site using the new technique initiated in the August 2017 demonstration of elevating the VZ-2000 mobile LiDAR onto a telehandler and driving it in a single pass across the test site (Figure 40). This effort was performed both prior to and following the application of smoke. Smoke was applied to the test site from canisters mounted on vehicles to the side of the test pad (Figure 41). The telehandler was then driven across the test pad with the LiDAR sensor being immersed within the smoke cloud as data were collected (Figure 42).

Figure 40. Telehandler mounted VZ-2000 LiDAR scanning the crater prior to application of smoke.



Figure 41. Delivery of smoke to test pad.



Figure 42. Telehandler mounted VZ-2000 Mobile LiDAR scanning in smoke.



3.5.2 Reduction of smoke interference in point clouds

The presence of smoke caused the LiDAR sensor to record airborne particles as objects within the point cloud creating considerable clutter in the scene. Figure 43 illustrates the second day of testing where the surface contained two detonated craters and a single camouflet. Visually, only one of the two craters can be spotted, and the camouflet is invisible on an otherwise clean concrete surface. The presence of these craters is masked by the large smoke cloud that can be seen on the profile view of the point cloud.

By rotating the smoke-filled point cloud, one can see a clear gap between the ground surface and the smoke in the scene. As such, the removal of smoke from the scene can be done by fitting a plane to the ground surface and

removing all points a fixed distance above the plane height. Such a process leaves only the damaged pavement surface (see Figure 44). This surface shows very clearly the location and contours of each crater and camouflet in sufficient detail to enable classification and measurement of its size (Figure 45). This data collection demonstrated the LiDAR's capability to penetrate obscurant fields and still render sufficient point cloud density to classify airfield damage features. It should be noted here that point density is reduced on the pavement surface where smoke has interfered with the source laser. This interference can occur on incidence or return of the laser.

Figure 43. LiDAR point cloud of damaged surface from top (top) and profile (bottom).

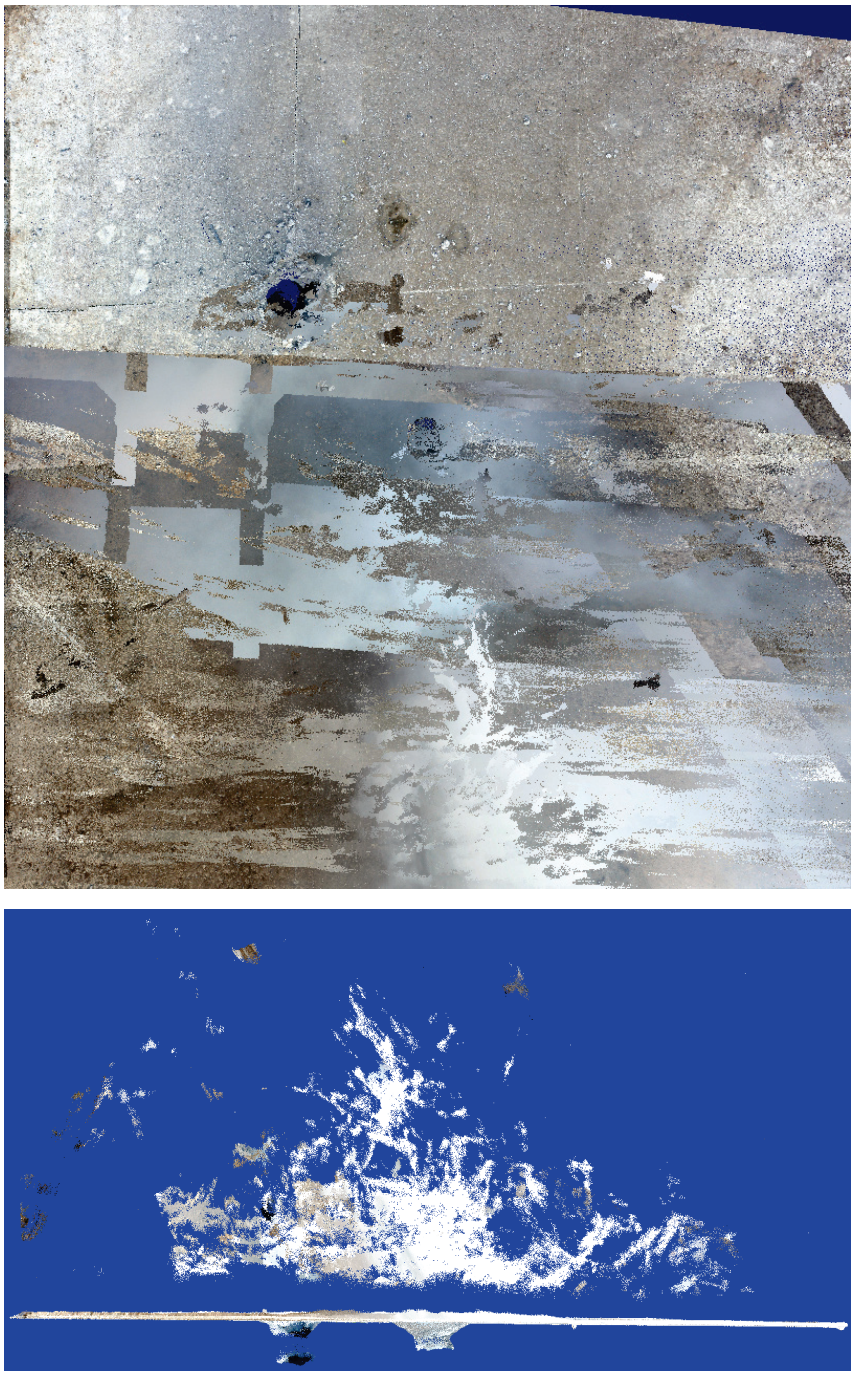


Figure 44. LiDAR point cloud of craters and camouflets after smoke removal (top) and profile (bottom).

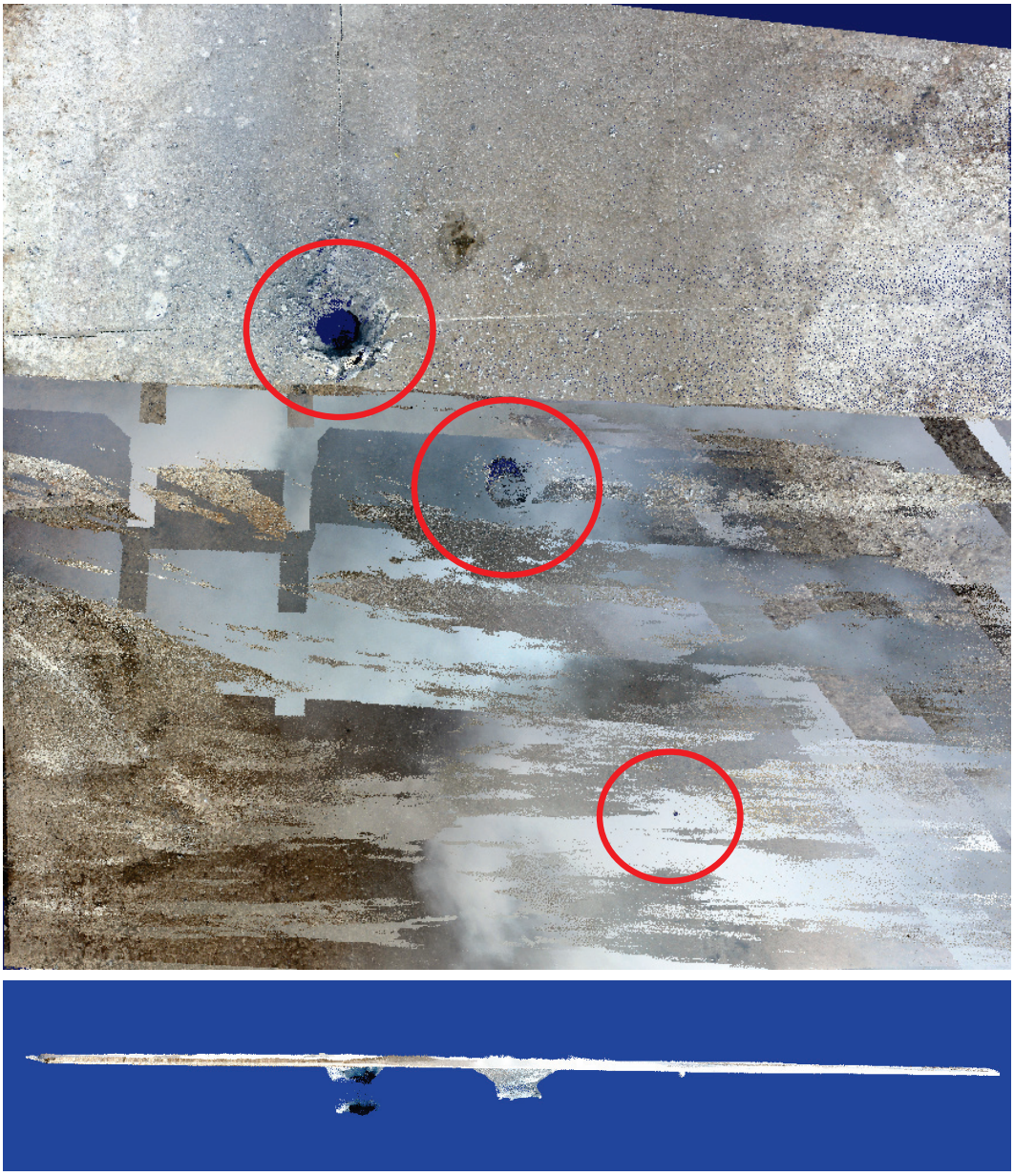
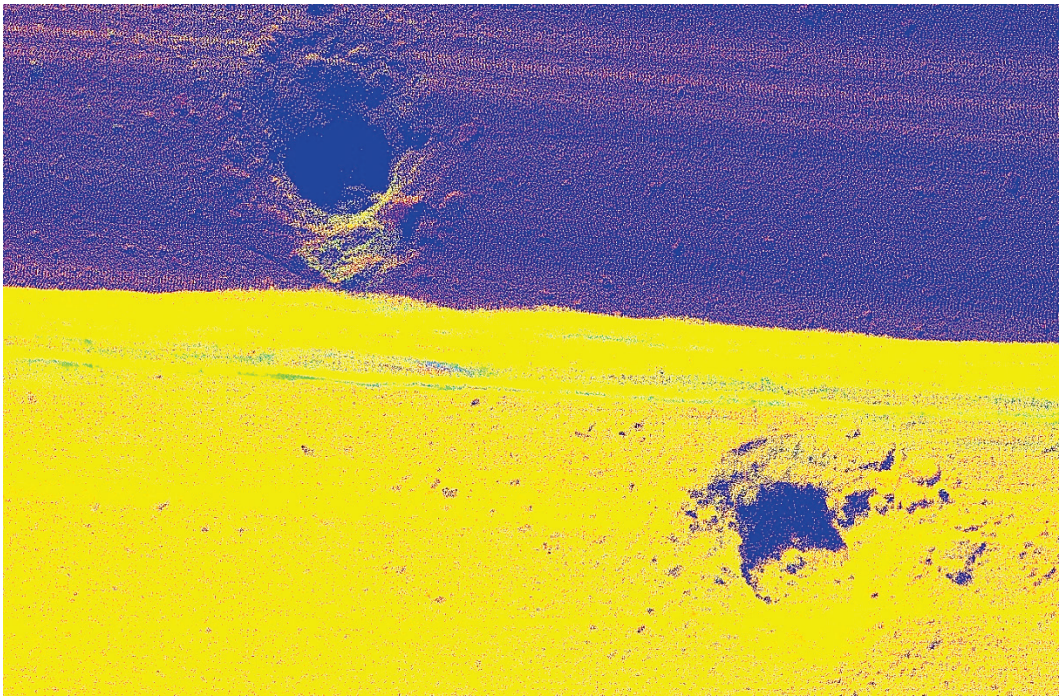


Figure 45. LiDAR point cloud of two craters following removal of smoke particles in point cloud.



3.6 Avon Park event (May 2018)

In May of 2018, a test event was conducted in Florida at the Avon Park Air Force Range, MacDill Air Force Base Auxiliary Field (Figure 46). Silver Flag was unavailable for testing during the summer months, so this location was selected as an alternate site. Blasting was not allowed at this facility so black-tire inner tubes were used as surrogate craters (Figure 47). This approach was used in prior data collects at Silver Flag in addition to the actual craters to provide training data for neural network algorithms in use by other RADAS teams, but had been ignored by the ERDC team to this point. However, since these tire inner tubes were the principal object of detection for this exercise, the MATLAB volumetric algorithms were modified to identify objects lying above the ground surface rather than beneath. This would prove to be beneficial as the inner tube surrogates were used again in the October 2018 collect when no blasting could take place.

Figure 46. MacDill Auxiliary Field test locations (1=asphalt, 2=concrete, 3=gravel).

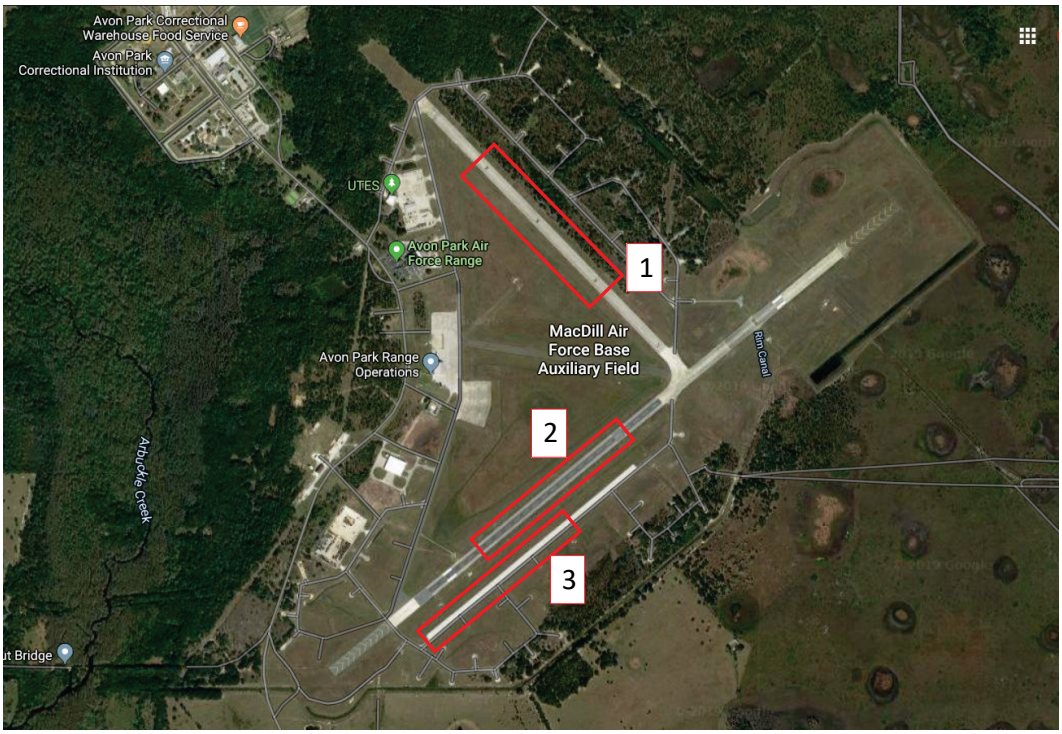


Figure 47. Image of asphalt section with UXO and tire crater simulants.



The benefit of the Avon Park test site was it provided the first data collection opportunity on two surfaces not previously studied, i.e., an asphalt runway surface and an unsurfaced gravel runway. The primary taxiways were asphalt interspersed with concrete tie-down pads. The dark background of the asphalt generated far different LiDAR intensity returns

versus concrete to help determine the effectiveness of UXO detection on asphalt. A short gravel assault strip had been constructed parallel to the main concrete runway for training events and served ERDC as a test location for crater detection on soil (Figure 48).

Figure 48. Tire inner tubes and UXO on gravel assault strip.



3.6.1 LiDAR data Collection

For this data collection exercise, only the mobile VZ-2000 LiDAR unit mounted on a telehandler was deployed by ERDC (Figure 49). At this stage of research, any future visual imagery would be derived solely from other RADAS team members flying more advanced UAS with EO sensors (Figure 50). However, as can be seen in the EO imagery, variance in pavement reflection obviated by the black asphalt versus the white concrete resulted in an inconsistent white balance, causing bright or dark spots on the pavement surface where visual detection of objects is difficult. LiDAR was collected in a single pass down the centerline of the test sections and provided a normal view of the pavement surface from an altitude of approximately 50 ft.

Figure 49. Reigl VZ-2000 mobile LiDAR mounted on telehandler at Avon Park, FL.

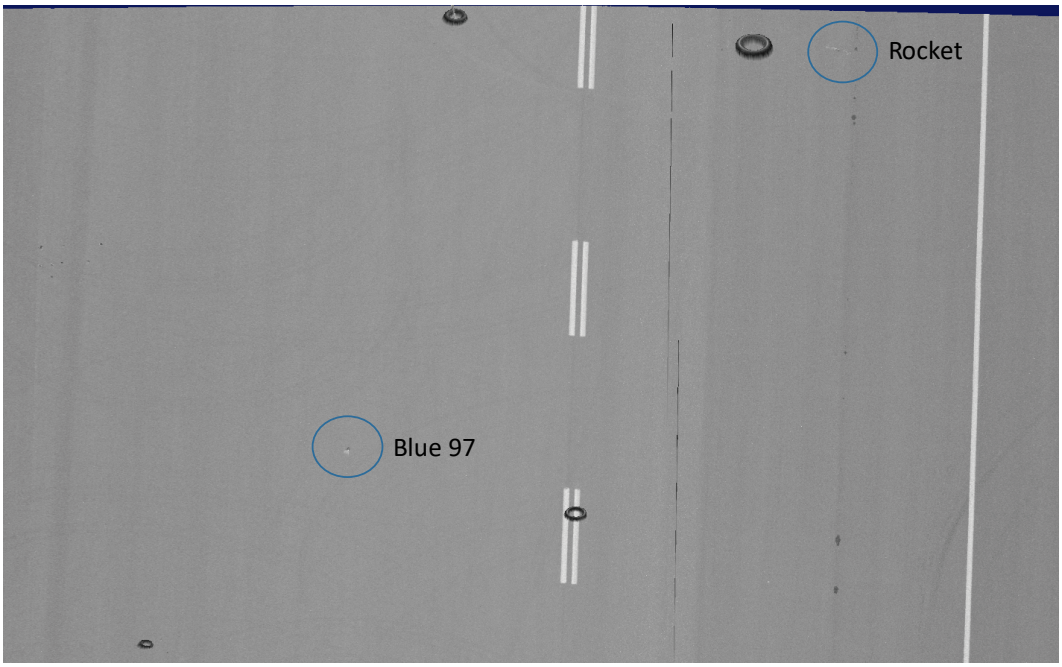


Figure 50. Overhead mosaic of a section of runway with concrete test pad and tires.



A total of three configurations of tires and UXO was setup on the asphalt taxiway, i.e., one setup on the concrete runway and one on the gravel assault strip parallel to the main runway. A LiDAR scan was performed for each of these configurations. This provided data on tires and UXO (asphalt and gravel only) on multiple surfaces for later evaluation for changes in intensity. An example of a mesh generated from the overhead LiDAR data is shown in Figure 51.

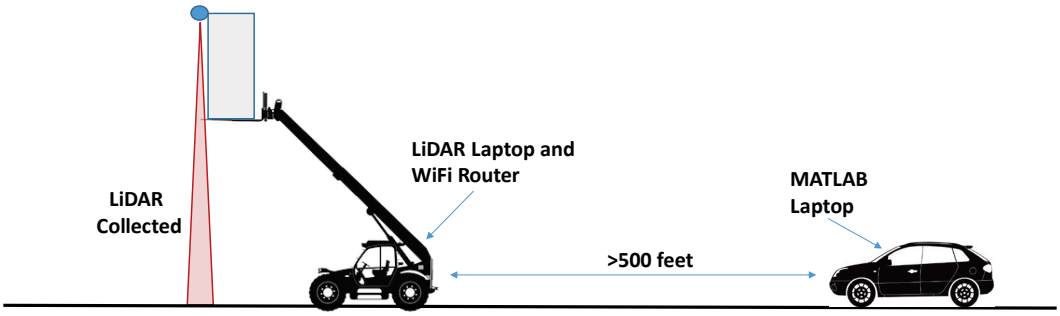
Figure 51. LiDAR mesh of Avon Park tires and UXO on asphalt surface.



3.6.2 Automated data processing

This data collection event marked the first attempt by ERDC to create a real-time data collection procedure. The initial setup consisted of two Windows-based computers connected to each other over a local Wi-Fi network. A Netgear Nighthawk X6 Tri-Band Wi-Fi router was used to facilitate the wireless connection. The first computer served as a remote processor of LiDAR data. The remote computer initiated the data collect using Riegl software. The Riegl software exported geo-rectified LAS files into a network folder. The second (base station) computer searched the network folder for changes and applied researcher-developed algorithms to any LAS data found. The team-developed MATLAB software generates damage files necessary for import into GeoExPT. A graphical representation of this procedure can be seen in Figure 52. The effective range of the Nighthawk router was found to be approximately 500 ft.

Figure 52. Logistics of automated data processing at Avon Park data collect.



3.7 Silver Flag event (October 2018)

In October of 2018, a multi-crater event was planned at the Silver Flag test site at Tyndall Air Force Base. This was a critical event in the ERDC technology development cycle for two reasons. First, the Riegl RiCopter UAS had permission and planned to fly the test event to collect the first UAS LiDAR data for RADAS. Secondly, ERDC was to demonstrate a new portable computer system that integrated the Riegl processing software with team-developed MATLAB algorithms to perform real-time damage detection from the LiDAR data as it was being collected. However, Hurricane Michael struck Tyndall Air Force Base two days after testing was scheduled to begin. Therefore, an abbreviated 1-day data collection was performed to provide the ERDC team with limited UAS LiDAR data to begin analyzing the suitability and limitations of this sensor platform. No craters were detonated so data collection occurred only on tires similar to the Avon Park data collection (Figure 53). UXO were distributed on the test site as well to determine if the RiCopter has sufficient point density to detect these smaller features as shown in Figure 54.

Figure 53. Layout of tires on Silver Flag near North taxiway.

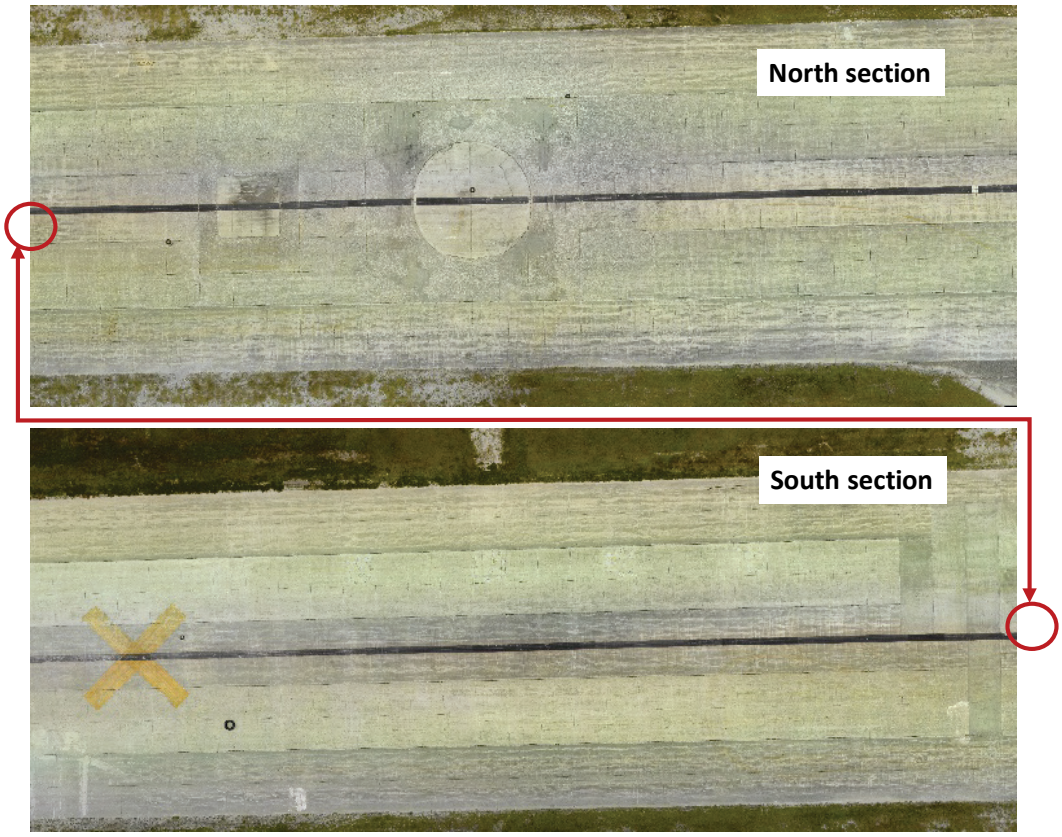


Figure 54. Example of UXO layout on airfield surface (Rockets and Blue 97s visible).



3.7.1 Telehandler LiDAR with NUC and Ubiquiti antennae

ERDC again deployed the VZ-2000 LiDAR system, telehandler mounted, for a two-fold purpose. First, it was used to obtain a high-density point cloud for comparison to the RiCopter and as a legacy sensor for comparison to prior data collections. Secondly, it was the platform on which a small, Windows-based computer and data transmission system was mounted to perform real-time data analysis.

In order to meet the time constraints imposed on data analysis within the RADAS program, ERDC determined that a simultaneous collection of LiDAR data and onboard processing would be required. Data would be passed from the processing platform to GeoExPT to enable the user to visualize damage classification and location concurrent with the data collection. Finally, when LiDAR scanning was completed, the final processing of any remaining data occurred within minutes. This enabled the operator to perform real-time quality control on the incoming damage detection to determine if the collection was valid and correct any inconsistencies in the data.

To accomplish this on-board processing, a Windows-based Intel NUC 7 computer (NUC7i7BNKQ) was used to handle the software requirements of both Riegl and MATLAB (Figure 55). The NUC was chosen for its compact size and lightweight build, measuring approximately 4.5 in. per side and 1 in. in height, and weighing only 690 g not including a battery or AC adapter to power it. This will enable the NUC to be mounted to the RiCopter system easily and still maintain payload capacity for the LiDAR sensor. During testing at Silver Flag, the NUC was connected to an external monitor such that data processing could be initiated. Since permission to mount the NUC on the RiCopter was not approved at the time of testing, it was mounted to the side of the telehandler and connected by an Ethernet cord to the VZ-2000 to allow for data collection and generation of LAS file point clouds (Figure 56). The MATLAB algorithms and Riegl processing software were both installed on the NUC, which analyzed the resultant LAS files, automatically detecting the volumetric damage and then wirelessly transmitting the location, type, and severity of each damage to a ground-based computer for later upload into GeoExPT.

Figure 55. Windows-based Intel NUC 7 computer.



Figure 56. Use of Reigl VZ-2000 mobile LiDAR on telehandler with NUC computer mount.



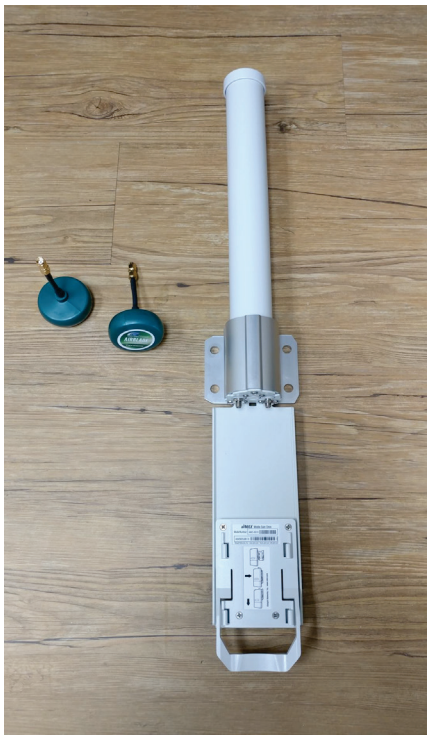
Data transmission from the remotely connected NUC to a land-based computer was accomplished using a series of radios and antennae developed by Ubiquiti Networks. Ubiquiti equipment is typically used by internet service providers (ISPs) for the transmission of data necessary for internet service. The team repurposes this technology to facilitate data transmission from remote and moving sources of data, such as mobile LiDAR and UAS LiDAR platforms. The driving component of this data transfer is the use of two Ubiquiti Rocket AC airMAX 5GHz point-to-point (PtP) radios. These radios provide sufficient power and frequency range to transmit data over distances greater than one mile. The true usefulness of these radios is tied, however, to the style and size of antennae used. For the ground station, the team did not have many size requirements and, therefore, selected a large directional antenna to facilitate data transfer -- Ubiquiti airMAX 2x2 BaseStation Sector (see Figure 57).

Figure 57. Close-up of Ubiquiti airMAX Sector antenna.



The Sector antenna is approximately 30 in. tall, 7 in. wide, and 3 in. thick. The Sector antenna weighs approximately 10.6 lb. Due to the large size of the Sector antenna, a robust tripod is necessary for staging. For this effort, the Sector antenna is mounted atop a TS-110B air-powered aluminum tripod speaker stand. This tripod is used for mounting heavy musical equipment and is more than sufficient to provide a stable base for the Sector antenna. In order to transmit data to and from a remote station, a smaller antenna is necessary. Specifically, for use on UAS platforms it is necessary to minimize impact on aircraft payload. As such, the smallest Ubiquiti antenna was selected for the ground station, i.e., a Ubiquiti airMAX 2x2 Omni Antenna (see Figure 58).

Figure 58. Close-up of Ubiquiti airMAX Omni antenna.

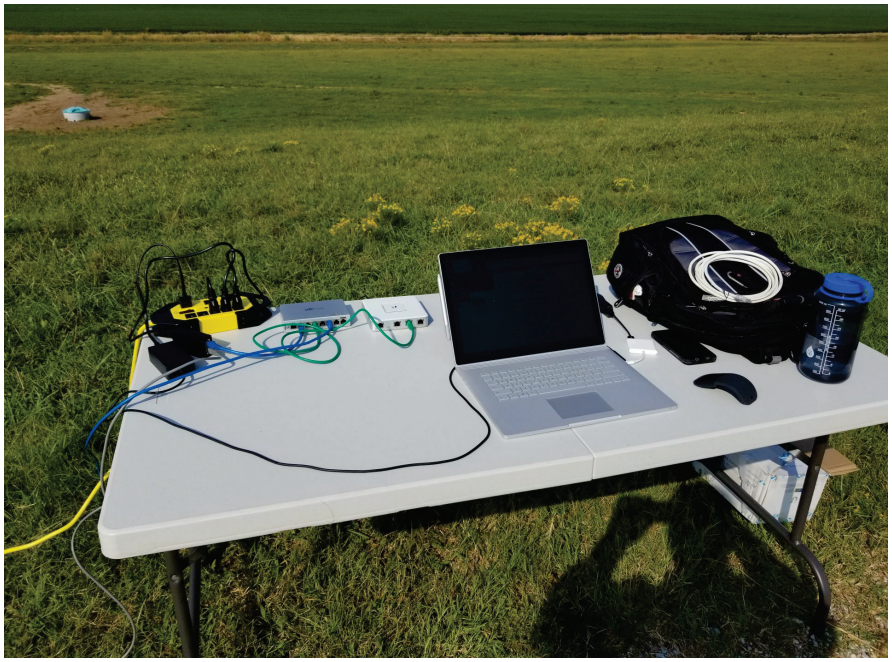


The Omni, unlike the Sector antenna, is an omni-directional antenna that transmits and receives radio waves in all directions normal to the antenna body. The Omni antenna is approximately 23 in. tall, 3.5 in. wide, and 2.6 in. thick. The Omni antenna weighs approximately 1.5 lb. While the dimensions of the Omni are similar to the larger Sector, the weight is significantly lower.

The radios and antenna described above represent the core of team data transmission setup. In order for these devices to interface with two Windows-based computers, other hardware is necessary to facilitate data transfer. Firstly, a modem, or security gateway, is necessary to manage the various internet protocols that each Windows-based computer needs to transmit data over a network. For this effort, researchers stuck to the Ubiquiti Networks equipment ecosystem by choosing a Ubiquiti UniFi Security Gateway. This device manages permissions between the ground station and remote computer. A security gateway is necessary on only one end of the network and is placed at the ground station end. Secondly, in order to allow multiple devices to communicate over radio, a network switch is used on each end. A slightly heavy, robust switch is used on the ground station end, the Ubiquiti UniFi 8-port Ethernet Switch. On the remote station end, a smaller, more lightweight switch is used -- the Ubiquiti

TOUGHSwitch 5-Port PoE Controller. An example of the ground station setup can be seen in Figure 59.

Figure 59. Ubiquiti ground station setup.



In Figure 59, the 8-port switch can be seen next to the yellow surge protector, and next to the switch is the security gateway. While the data transmission setup described above is complex, all necessary equipment can be transported in a single Pelican case (see Figure 60).

Figure 60. Pelican case transport of Ubiquiti equipment.



Using the equipment introduced above (Figure 61), researchers are able to transmit data originated from a remote platform, such as LiDAR data processed on a NUC 7 computer, to a ground-based platform where more network and processor intensive tasks can be completed. In general, the data transmission pipeline works in this order.

1. Data collected using LiDAR sensor (VMZ-2000 or VUX-1LR).
2. Raw LiDAR data transmitted locally to NUC 7 computer over Ethernet.
3. Raw LiDAR data processed on local computer yielding airfield damage shapefiles.
4. Resultant shapefiles transmitted over custom Ubiquiti network to ground station.
5. Ground station computer transmits data locally or over land-based network to a server linked to GeoExPT.

Figure 61. Ubiquiti receiver antenna connected via Ethernet to land-based laptop monitored from a ground station.



3.7.2 Riegl RiCopter UAS deployment

The use of the Riegl RiCopter (Figure 62) provided the first opportunity to collect normal airborne LiDAR data with all the limitations imposed on the data quality based on UAS flight characteristics. The RiCopter utilized both its VUX-1LR LiDAR system along with its twin Sony Alpha cameras to provide LiDAR colorization in post processing. The colorization is not a requirement of the RADAS mission but was used as a research tool.

Operation of the RiCopter required a dedicated pilot who flew missions using both a manual flight controller and a software mission planner (Figure 63 and Figure 64). The flight controller was used for the initial takeoff and IMU initialization performed through a series of figure-8 maneuvers before initiating the mission planner for autonomous flight. The pilot then turned off the autonomous flight and manually landed the RiCopter. A two-man team performed a pre-flight checklist to ensure all systems were operating correctly before flight (Figure 65). Navigation for the RiCopter was provided by an onboard GPS sensor. Post-processing corrections to the actual position of the LiDAR point cloud was provided by a real-time kinetic global positioning system (RTK GPS) that was setup on a tripod on site.

Figure 62. Reigl RiCopter LIDAR UAS with VUX-1LR system mounted underneath.



Figure 63. RiCopter pilot with control station monitoring UAS in the air (left), RiCopter ground station with mission planning software laptop and communication antenna (right).



Figure 64. Image of flight plan for RiCopter along south end of Silver Flag test site.



Figure 65. Pre-flight inspection of RiCopter with two-man team.



Due to the shortened data collection window, only two series of flights were undertaken. Data were collected at two elevations, 50- and 100-ft AGL, and at two speeds, 5 and 10 m/s. This provided four data sets from which to evaluate varying point density and sufficiency to

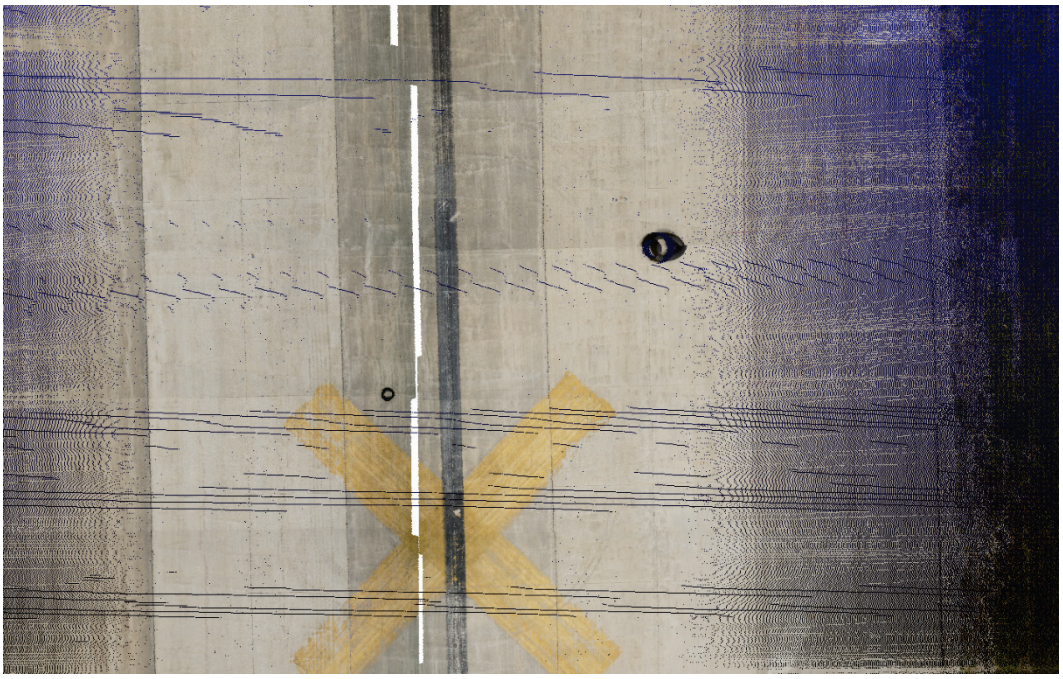
determine tire locations and UXO. The VZ-2000 system scanned the same area as the RiCopter to provide a high-density point cloud to compare to the reduced cloud density from the airborne system. Point densities in points per square meter from the collected clouds are shown in Table 3. This illustrates that the VZ-2000 has far greater point density than the RiCopter; however, as will be discussed, damage detection does not require such dense point clouds.

Table 3. Point densities from RiCopter VUX-1LR and VZ-2000 LiDAR systems.

	Elevation (feet)	Speed (m/s)	Density (pts/sqm)
RiCopter	100	9.0	500
	100	5.0	784
	50	9.0	797
	50	5.0	1353
VZ-2000	40	0.5	3011

An example of LiDAR point cloud colorization for a section of runway with tires is shown in Figure 66. Riegl's semi-automated point cloud colorization method is often inaccurate. Linear features such as paint stripes or slab joints can appear faulted, as well as shadows or blur on static images. This is not indicative of a poor data collect, rather an inability of the software to properly align the photos with the point cloud and some deviations in GPS position can cause shifts in color assignment.

Figure 66. Colored LiDAR image of two tires place on surface of Silver Flag runway.



3.8 Silver Flag event (March 2019)

March 28-30, 2019, a live blast test event was held at the Silver Flag airfield at Tyndall AFB to compensate for the missed opportunity the prior October. During this test event, seven craters, two sets of spalls, and seven camouflets were detonated. An overhead mosaic of the damage features was generated from static imagery taken by a participating RADAS contractor. ERDC decided that its optical technology could be best handled cooperatively with other UAS systems that will simultaneously be collecting data with the LiDAR. The craters formed are shown in Figure 67, and the spalls and camouflets in Figure 68. An additional five camouflets (Figure 69) were located south of the initial blast area for additional cratering operations that were to take place at a later date and included a tire field during day-time operations for CNN training by other RADAS teams. In Figure 70, there are seven rather than the intended six craters in the concrete due to one of the camouflet charges breaking through the surface and initiating a crater rather than a subterranean cavity.

Figure 67. Location of six craters plus cratered camouflet (item 7) at Silver Flag.



Figure 68. Location of two spalls and two camouflets at south end of crater field.

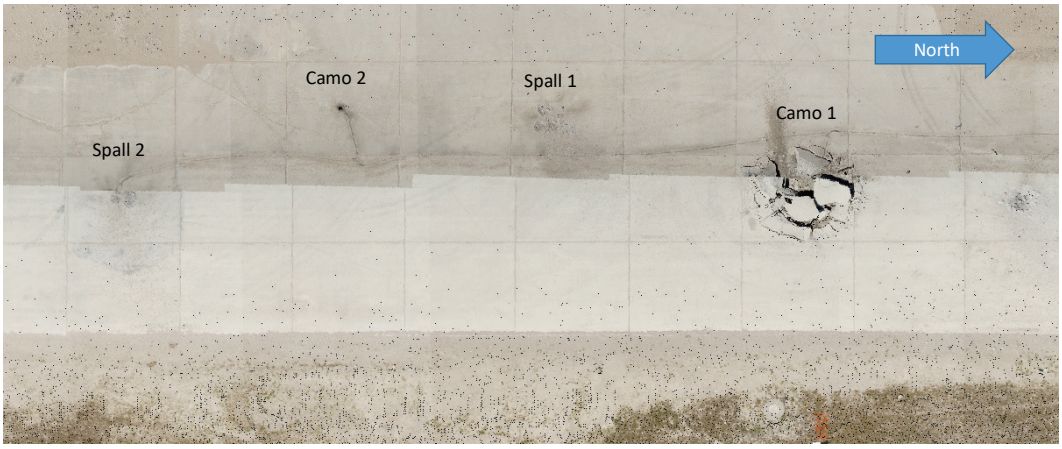


Figure 69. Location of additional camouflets and tire field south of the blast area.

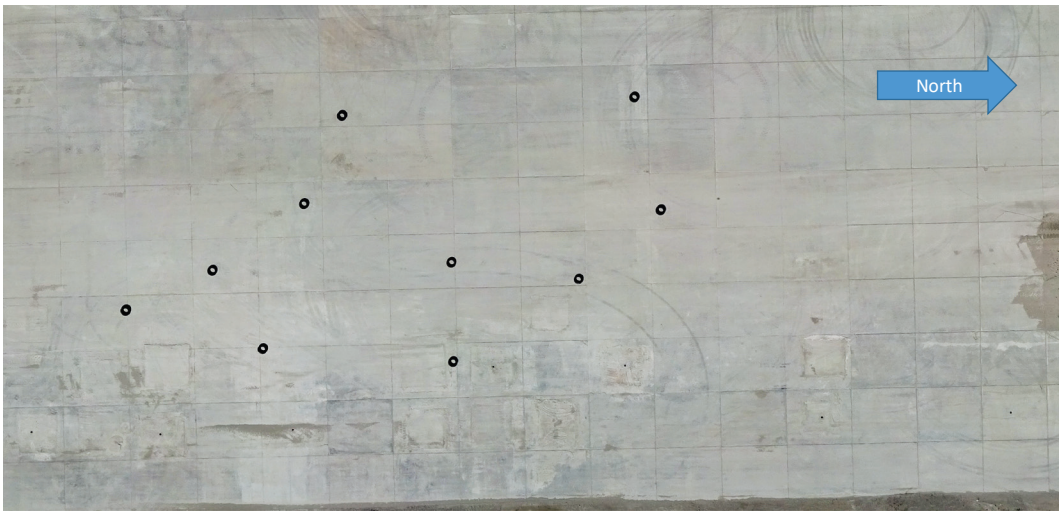
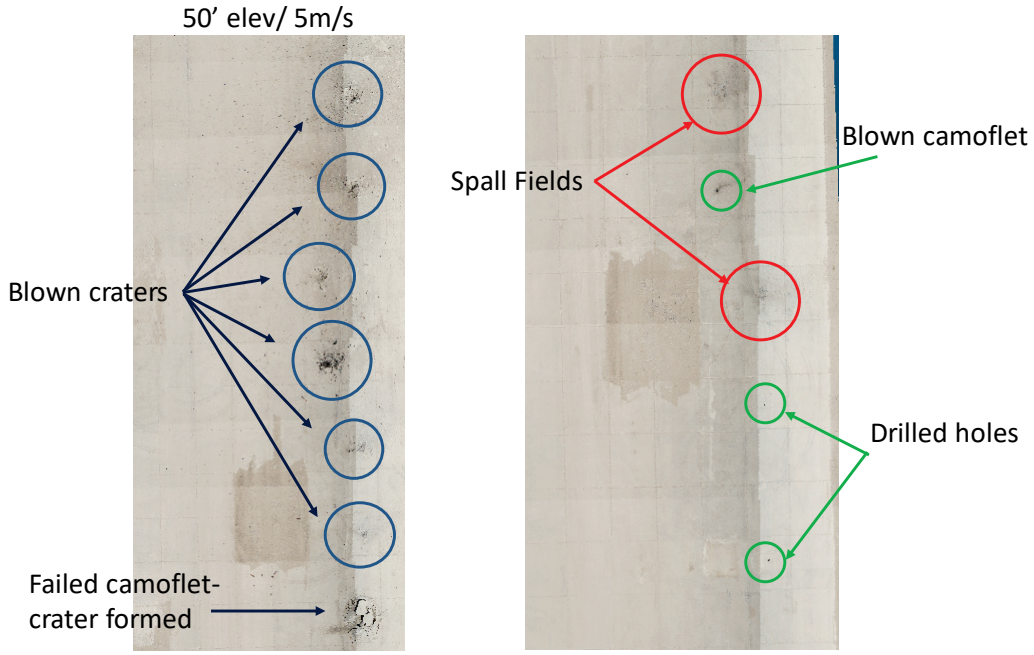


Figure 70. Diagram of intended damage features on Silver Flag runway.



3.8.1 Reigl RiCopter LiDAR collection

The Reigl RiCopter was used as the primary data collection platform for this test event (Figure 71). Until now, no UAS LiDAR data had yet been collected on an actual cratered surface nor with a large debris field in which to distort UXO detection. The mobile VZ-2000 LiDAR on the telehandler was not used at this test event as the RiCopter demonstrated its effectiveness at generating sufficient point cloud density to detect UXO and surrogate tire craters at the October 2018 test event. Further, the automated detection process had

proven to be effective at the prior demonstration. The next stage in demonstrating the ERDC automated detection process is to have the NUC mounted to a UAS for in-flight processing.

During this data collection event, the RiCopter was flown at a number of speeds and heights borrowing on the knowledge gained from the October data collection (see Table 4). A series of flights were also performed in the 30-min window following sunset, the twilight period of the day, to simulate as close to possible night operations. This time window is in accordance with Federal Aviation Administration (FAA) regulations regarding night operations of sUAS aircraft without a waiver. The evening flight was used to validate the effectiveness of LiDAR collection at all times of day or night (Figures 72 and 73).

Table 4. RiCopter flight altitudes and speeds during LiDAR collection.

Flight Line Details		
Flight Line	Flight Altitude (ft. AGL)	Airspeed (m/s)
1	50	5
2	50	9
3	100	5
4	200	5
5	260	5
6	330	5
7	400	5

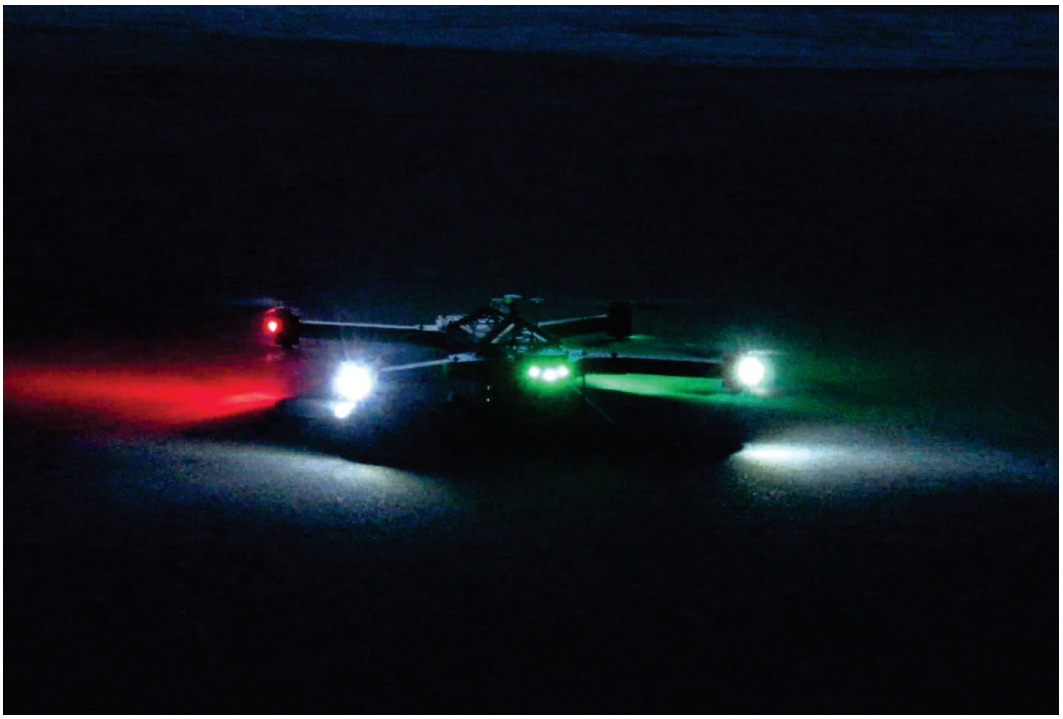
Figure 71. Reigl RiCopter undergoing preflight check list (left) and ground control station with mission planner laptop, radio transmitter, and pilot.



Figure 72. RiCopter prepared for twilight operations (30 min past sunset).



Figure 73. RiCopter lighting for visual tracking during twilight operations.



4 Algorithm Development

This research sought to enable the rapid detection of airfield damage through an analysis of unorganized three-dimensional (3-D) point cloud data collected by mobile and aerial LiDAR sensors. The MATLAB programming language and Integrated Development Environment (IDE) was chosen to implement the methods described in this chapter. MATLAB offers a variety of benefits making it suitable for such an effort. Namely, MATLAB has a number of prebuilt and for-pay functions that are already time-optimized; furthermore, MATLAB has an active community file share from which user-created functions and methods can be utilized. The MATLAB environment is not targeted at any one research area in particular and suits a variety of topics such as signal processing, image processing, point cloud processing, statistical analysis, and more. This wide variety of capabilities is necessary when designing algorithms combining multiple fields, such as will be discussed in the following sections.

4.1 Mesh creation methods

A common theme in the algorithms described here is the need to convert unorganized 3-D point data into an organized rigid surface or mesh. There are a variety of methods capable of producing such a surface from scattered 3-D data.

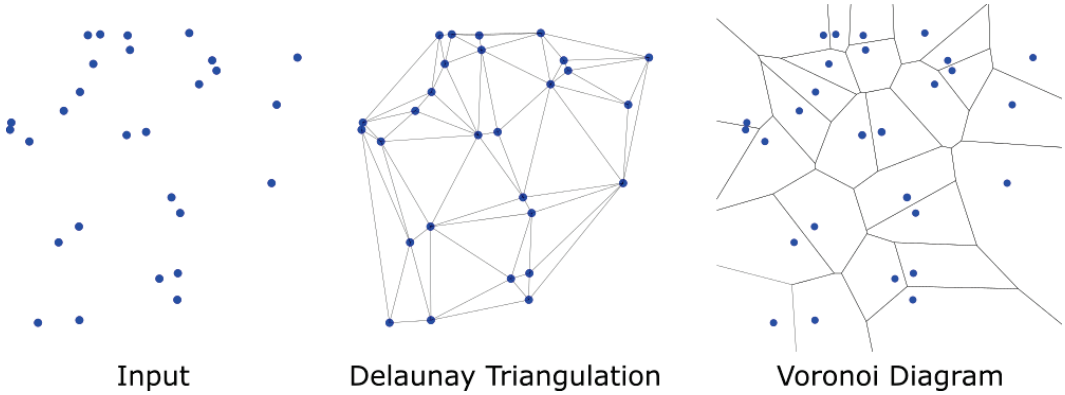
1. Interpolation
    a. Linear
    b. Nearest neighbor
    c. Natural neighbor
2. Extrapolation
    a. Linear
    b. Nearest neighbor
3. Grid Averaging

Interpolation and extrapolation techniques are closely related and are typically used in combination to fully render a mesh from scattered data. Their shared methods, linear and nearest neighbor, are the same. The principal difference between the two techniques is that interpolation is used when the neighborhood of points is known and sufficient to estimate encompassed empty space, and extrapolation is used when this is not the case and empty space must be extrapolated, or extended, from any points

nearby. As an example, interpolation works best to fill in gaps between LiDAR points where missing data may have occurred due to the presence of water or interference from objects or grazing angle of the sensor. Extrapolation works best to estimate the behavior of points outside of the collected area such as the furthest extents of the LiDAR or edges of the scan.

Linear interpolation in the case of surface reconstruction refers to a triangulation-based linear fitting method. In this method, a Delaunay triangulation (Delaunay 1934) is performed over the scattered points, resulting in all points becoming the vertex of a triangle with the restriction that no points lie inside the circumcircle of any triangle (Figure 74). A plane is then fit to the three linear edges of each triangle making up the reconstructed surface. The reconstructed surface described here is C^0 continuous; as in, all functions making up the surface are continuous. Nearest neighbor interpolation approximates an unknown area of empty space by giving it the same value as the nearest point. While simple in nature, this method produces a discontinuous, or piecewise, surface that does not appear smooth. Natural neighbor interpolation is the middle ground between linear and nearest neighbor interpolations. The natural neighbor method first calculates the Voronoi diagram (Voronoi 1908) of the scattered points. That is, polygonal areas are constructed around points of similar value where the boundary represents the passing of a pre-determined difference threshold. The natural neighbor method then calculates the value of a query point, or point in empty space, by measuring the weighted area of overlap in neighboring polygonal Voronoi areas. This process is then repeated until the desired surface grid spacing is reached. The benefits of natural neighbor interpolation are that the surface is continuous at all points and C^1 continuous at most points; that is, most bounding functions making up the surface are differenceable. This provides a much smoother surface than that produced by the nearest neighbor method.

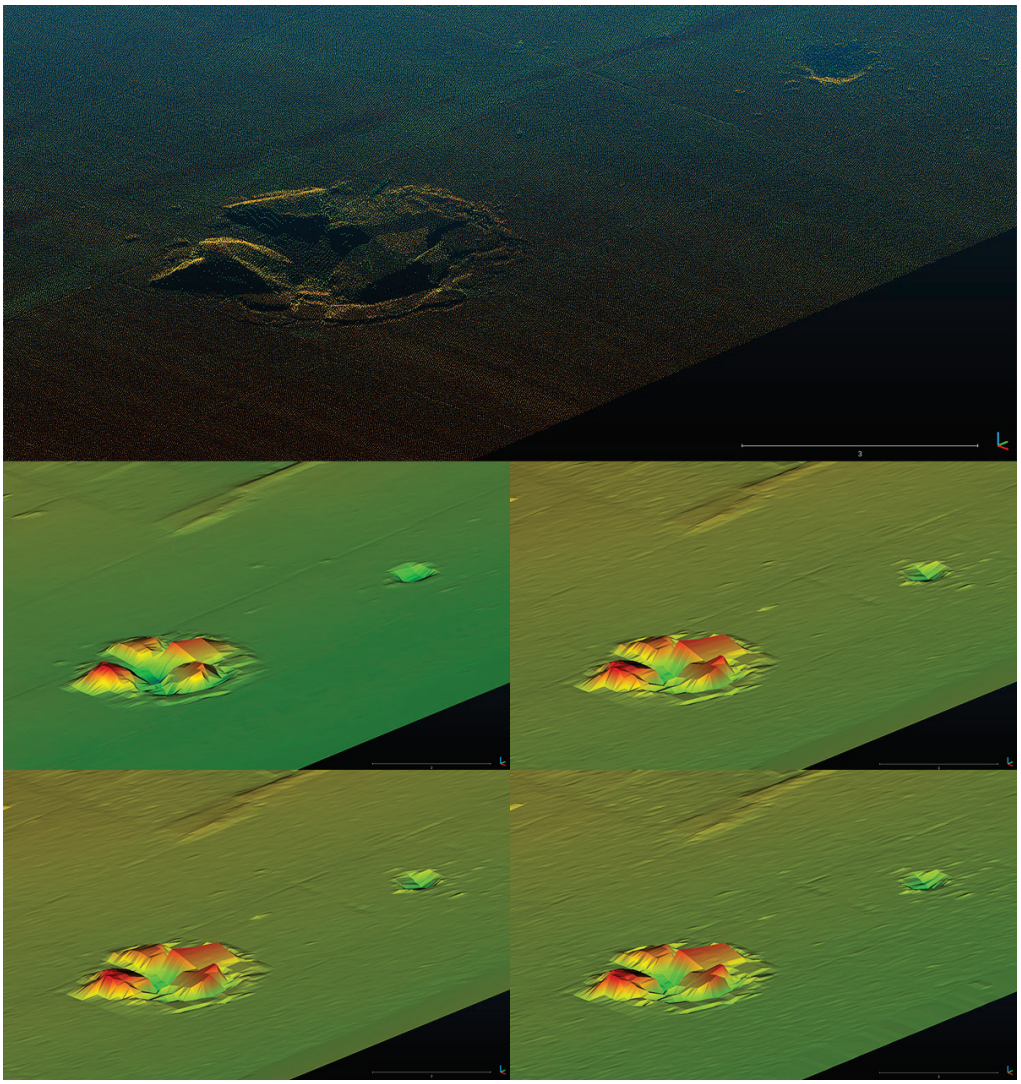
Figure 74. Delaunay triangulation and Voronoi Diagram for a random 3-D point cloud input.



While the above methods are principally simple, computationally, they rely on complex triangulations and many weighted calculations to be made at each grid point. For the sake of time-efficiency, the following method of grid averaging is also considered. The grid-averaging method could be considered a mix of extrapolation and interpolation techniques. It is simply the creation of a surface by use of the mean of nearby points. The produced surface cannot claim to be C^0 or C^1 continuous at any point. Furthermore, should the grid spacing encompass an area of completely empty space, i.e., no sample points, then the grid is returned empty. This must be remedied after completion of grid averaging by further filling empty grids with the average value of nearby grids.

Interpolation and extrapolation methods will typically yield a more accurate surface. The drawback to more accurate methods like this is longer processing times for large point clouds. Further, at low mesh resolutions, the accuracy differences between the meshes is negligible (see Figure 75). In point cloud applications relevant to the present work, it is estimated that the various interpolation and extrapolation methods are roughly 60% slower than the grid-averaging method, which is a critical time savings when considering the RADAS time frame. It can be seen in Figure 75 that despite the longer processing time of the more complex methods, the reconstructed surfaces are largely similar. Furthermore, for the current application to airfield pavement damage detection, it is beneficial that the grid-averaging method tends to smooth rough surfaces such that they appear flat. This prevents the creation of surface artifacts that may be picked up in surface volume detection techniques.

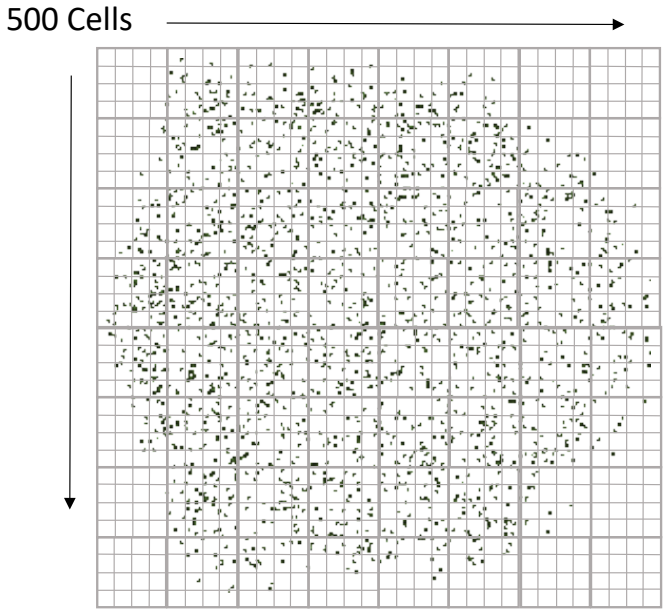
Figure 75. Comparison of various mesh creation techniques.



* Top, point cloud. Center, left, Grid Average. Center, right, Linear Interpolation. Bottom, left, Natural Neighbor Interpolation. Bottom, right, Nearest Neighbor Interpolation.

Once a surface of fixed grid size is constructed from the 3-D scattered data, the grid points are input into damage detection algorithms. Due to the time-sensitive nature of military airfield damage remediation efforts, a grid size must be selected that yields sufficient detail to register pavement damage but also small enough to prevent time-costly processing. Initial testing on typical LiDAR-generated point clouds revealed that, in general, a grid size of 500 × 500 cells when applied to a square point cloud was sufficient to give detail to large and small damage (Figure 76).

Figure 76. Example of 500 x 500 grid applied to point cloud for grid estimation.



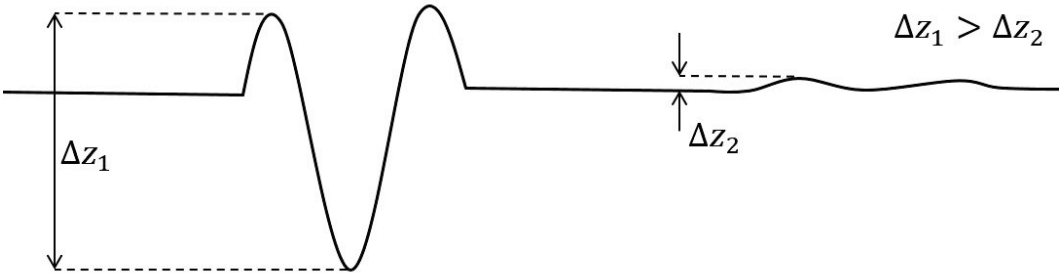
4.2 MATLAB functions for damage detection

As already mentioned, the MATLAB environment provides a number of unique and proprietary functions. Of those, many are community developed. Three community-developed functions are of note for the present work, i.e., *lasdata*, *bin2mat*, and *inpaintn*. The *lasdata* function provides fast LAS import capability. The LAS format is widely used with LiDAR data and is necessary for use in the present effort. The *lasdata* function also provides support for all LAS formats from 1.0 to 1.4, making it suitable for the presently used LiDAR data collection software. The *bin2mat* function is a time-optimized implementation of the grid-averaging method described in Section 4.1. The *bin2mat* function takes as input three $1 \times N$ arrays of scattered, unorganized 3-D data (xyz), and two $N \times N$ matrices representing the mesh x and y variables. The *bin2mat* output is an $N \times N$ matrix representing the elevation, or z, values of the mesh. The output mesh inherits the geo-referenced coordinate system of the scattered 3-D data. Lastly, the *inpaintn* function provides a fast method for filling the NaN (not a number) values that may result from the grid-averaging method. The NaN values result from application of the grid-averaging method on empty space.

4.3 Z-deviation method

The initial approach to detection and classification of airfield pavement damage considered the problem to be one of measuring changes in elevation, z-deviation. That is, airfield pavement damage such as craters, spalls, and camouflets could be described by variations in z-deviation (see Figure 77).

Figure 77. Changing z-deviation with changing surface geometry.



4.3.1 Algorithm introduction

Principally speaking, the process of measuring and reporting z-deviations on a surface is simple; however, there are present, in real-world circumstances, anomalies that must be accounted for in order to reduce false positives. Such a measurement would be sensitive not only to pavement damage resulting from unwanted detonations but also to variations in surface geometry common on airfields.

1. Drop-off from pavement shoulder to the infield area.
2. Water drainage systems.
3. Surface curvature resulting from changes in the underlying soil structure.

A method relying on changes in elevation must account for these instances in order to isolate only pavement damage. As such, a difference surface method is considered here in combination with the aforementioned z-deviation measure. In order to construct a difference surface, two surfaces must be defined in the process, i.e., the pre-surface that exists prior to any damage and the post-surface following ordnance detonation. The data resulting from aerial LiDAR scans following ordnance detonation make up a post-surface. It is then necessary to construct or directly measure a pre-surface, which is to be removed from the post-surface. The following sections will detail how, absent secondary LiDAR scans, a pre-surface is constructed and used to remove the common surface geometry referenced above.

4.3.2 Initialization

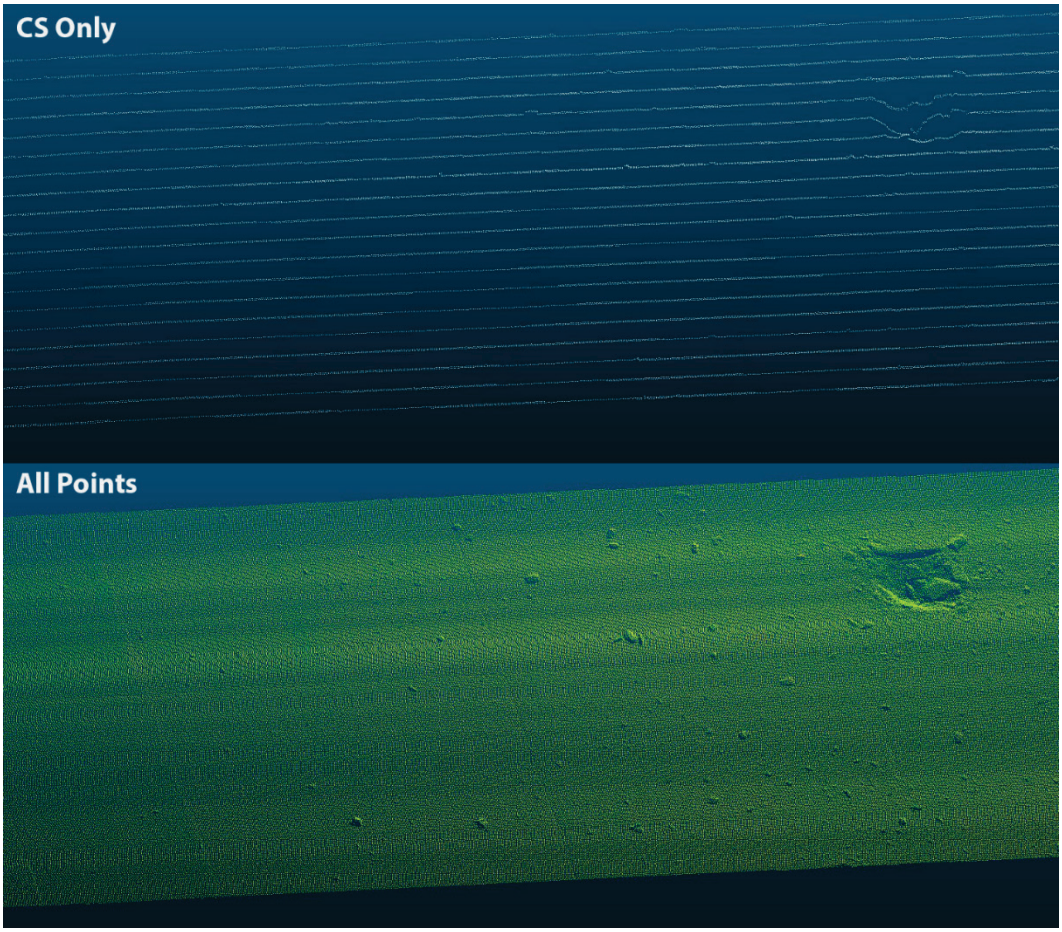
Generally, one can make z-deviation measurements directly on a 3-D unorganized point cloud. It is, however, difficult to make such measurements on large unorganized point clouds as the process devolves, computationally speaking, into the inefficient searching of multiple large 1-D arrays. It is with this in mind that researchers initialize the z-deviation algorithm by converting the large 3-D unorganized point clouds resulting from LiDAR data collection into a rigid surface. This is done using the grid-averaging method detailed in Section 4.1.

4.3.3 Generate pre-surface

In an effort to remove surface geometry trends from the post-surface, a pre-surface is constructed. If a pre-scan of the surface of interest exists, then that should be used as the pre-surface. Should no pre-scan data be available, then a pre-surface can be approximated by selecting an undamaged portion of the post-surface. Let us call this undamaged portion the "good area." The good area should be selected such that any repeating surface geometry trends are captured. The good area is then replicated until it matches the size and shape of the post-surface.

It is noteworthy here that the problem of surface damage remediation is a time-critical one. Optimizations must be made to all algorithms to ensure timely analysis. With this in mind, instead of creating the good area from the original unorganized 3-D points, it is created using a series of pre-cut cross sections. In this way, the problem of removing the good area from the post-surface is reduced from searching tens of millions of un-organized points to searching tens of thousands. A drawback with this approach is that the generated pre-surface is much less representative of the actual good area from the post-surface (see Figure 78). This drawback is negligible when the surface geometry trends are extensive enough to be picked up on successive cross sections.

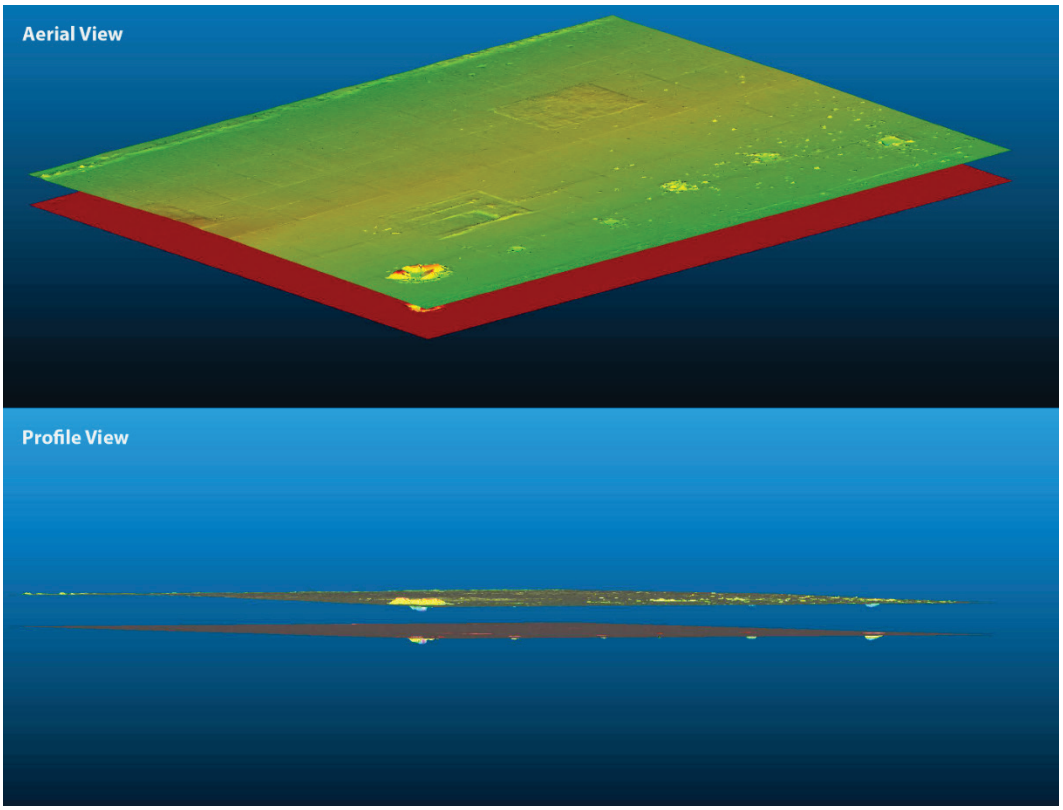
Figure 78. Comparison of cross section and all-points good area selections for z-deviation method.



4.3.4 Generate difference-surface

The generation of a difference-surface follows the creation of a pre-surface matching the size of the post-surface. That is, an $N \times N$ post-surface mesh exists at some real-world scale and a pre-surface exists with the same mesh spacing, $N \times N$, and same scale. Once the pre-surface elevations are removed from the post-surface mesh, a difference mesh is created. This mesh maintains the same XY properties as the post-surface; however, the elevation values are now shifted (see Figure 79). The elevation values in the difference mesh do maintain the same real-world scale as the post-surface.

Figure 79. Difference-surface elevation change for z-deviation method.

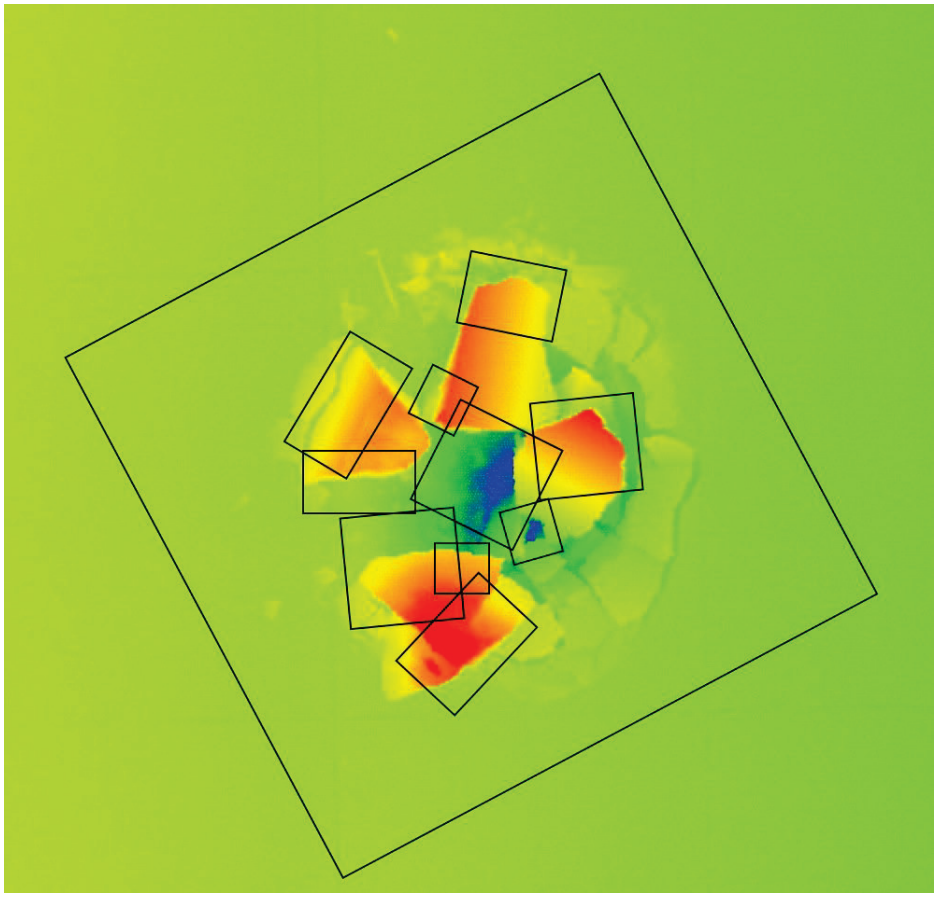


The difference surface should maintain all large pavement damage and remove repeatable surface geometry. Small damage such as spalls may be smoothed out by the good area removal.

4.3.5 Z-deviation scan

With the preparation of a difference-surface completed, it is necessary to scan for pavement damage. The individual grids of the $N \times N$ difference-surface mesh are analyzed one by one. A z-deviation threshold is defined that represents the boundary between elevation changes representing pavement damage and elevation changes representing flat ground or debris. Any grid meeting or exceeding this threshold is marked as pavement damage. Given the high likelihood that a single crater will consist of many such grids, those nearby are merged. This is done by combining marked grids with similar centroid locations (see Figure 80).

Figure 80. Combination of marked grids into one larger grid for the z-deviation method.



4.3.6 Damage classification

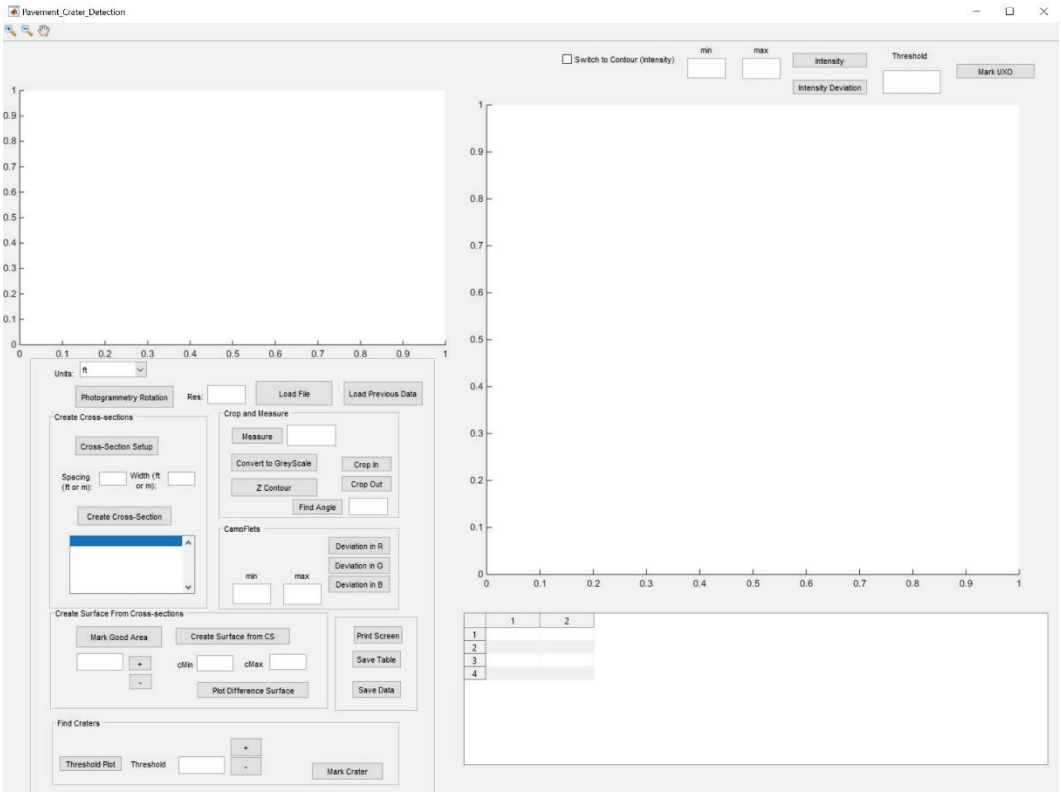
Of the three types of pavement surface damage, the z-deviation technique is suitable only to detect craters. Craters provide enough opportunity to detect grid cells with sufficient elevation change to be detected by the aforementioned z-deviation measurement. Spalls are already artificially shallow as a result of the grid-averaging mesh creation method and are made even shallower by the removal of an approximated pre-surface. Camouflets are similarly smoothed and unnoticeable by a z-deviation scan following the difference-surface method.

4.3.7 MATLAB implementation

GUI design for the z-deviation method required a vast amount of user interaction and automated very little of the process. There are many options for processing, and they require experienced personnel for execution (see Figure 81). It should be noted here that there are a number

of features present in Figure 81 that will not be discussed directly by this report. This is due to the fact that some of the options are developer tools for debugging purposes and not of interest to end users.

Figure 81. MATLAB-developed GUI for z-deviation method.



The first step to initialize the software is to load the scanned surface of interest. This is done by first selecting a coordinate system scale (Universal Transverse Mercator [UTM] meters or feet) in the "Units" dropdown box (see Figure 82). Following this, the user must select a resolution for the pre- and post-surface meshes. This is done by inputting an integer into the "Res" textbox (see Figure 82). This resolution value represents the total number of grids in one dimension of the mesh, N, such that the mesh size is $N \times N$ for both the pre- and post-surface. Following the above two inputs, the "Load File" button can be pressed to import LiDAR data (see Figure 82). Imported data formats are restricted to the LAS format. Following import, the point data are displayed as a 500×500 mesh on the right side of the software (see Figure 83). This plot is updated throughout this process as the user makes selections and visualizes data.

Figure 82. Load file portion from the z-deviation method GUI.

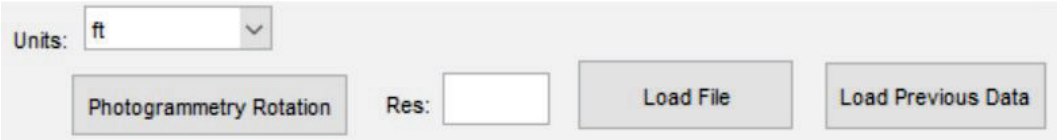
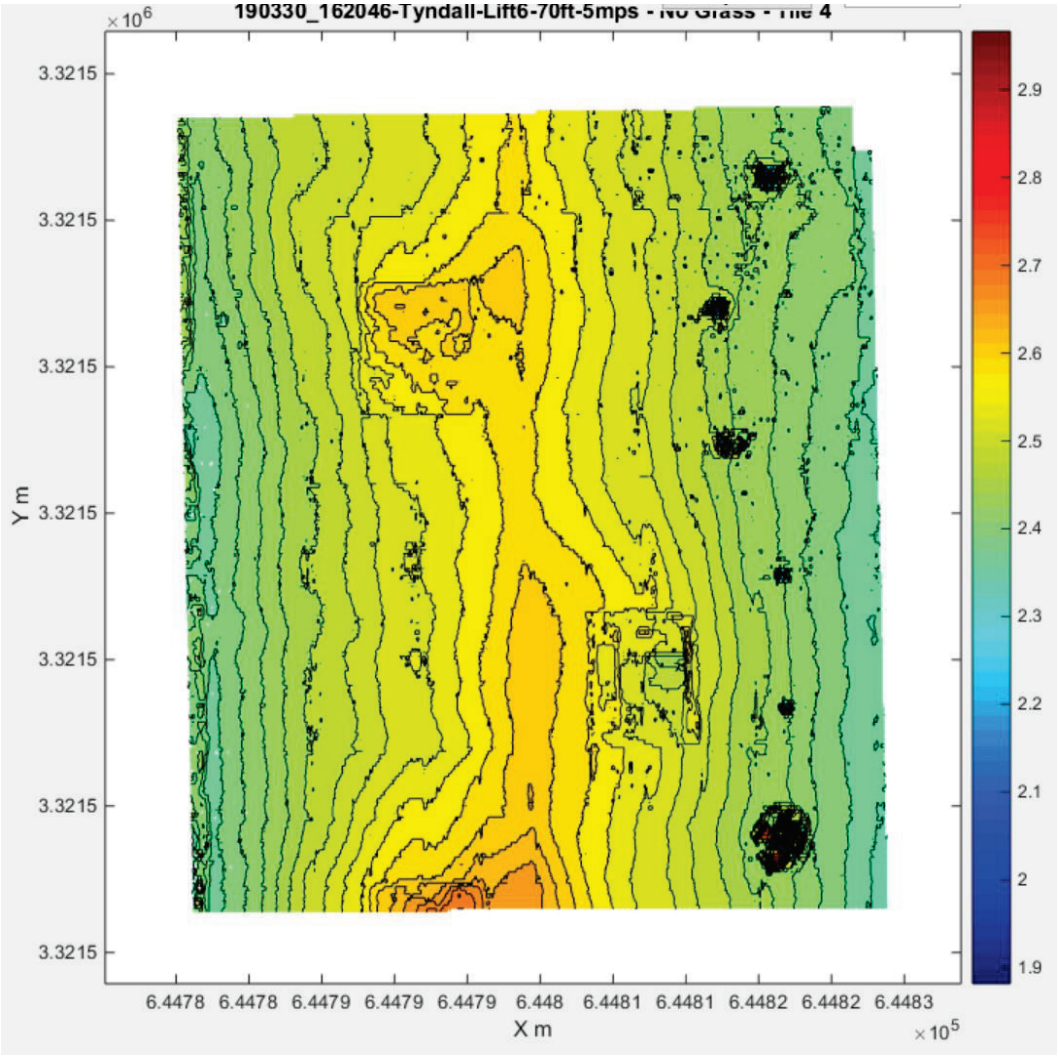
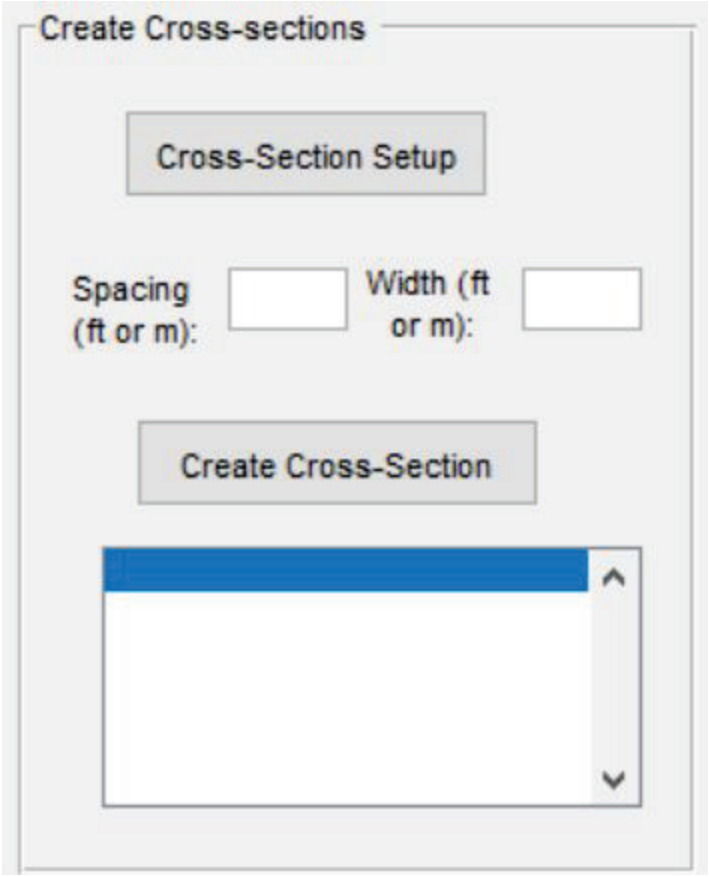


Figure 83. Mesh plot of imported post-surface from the z-deviation method GUI.



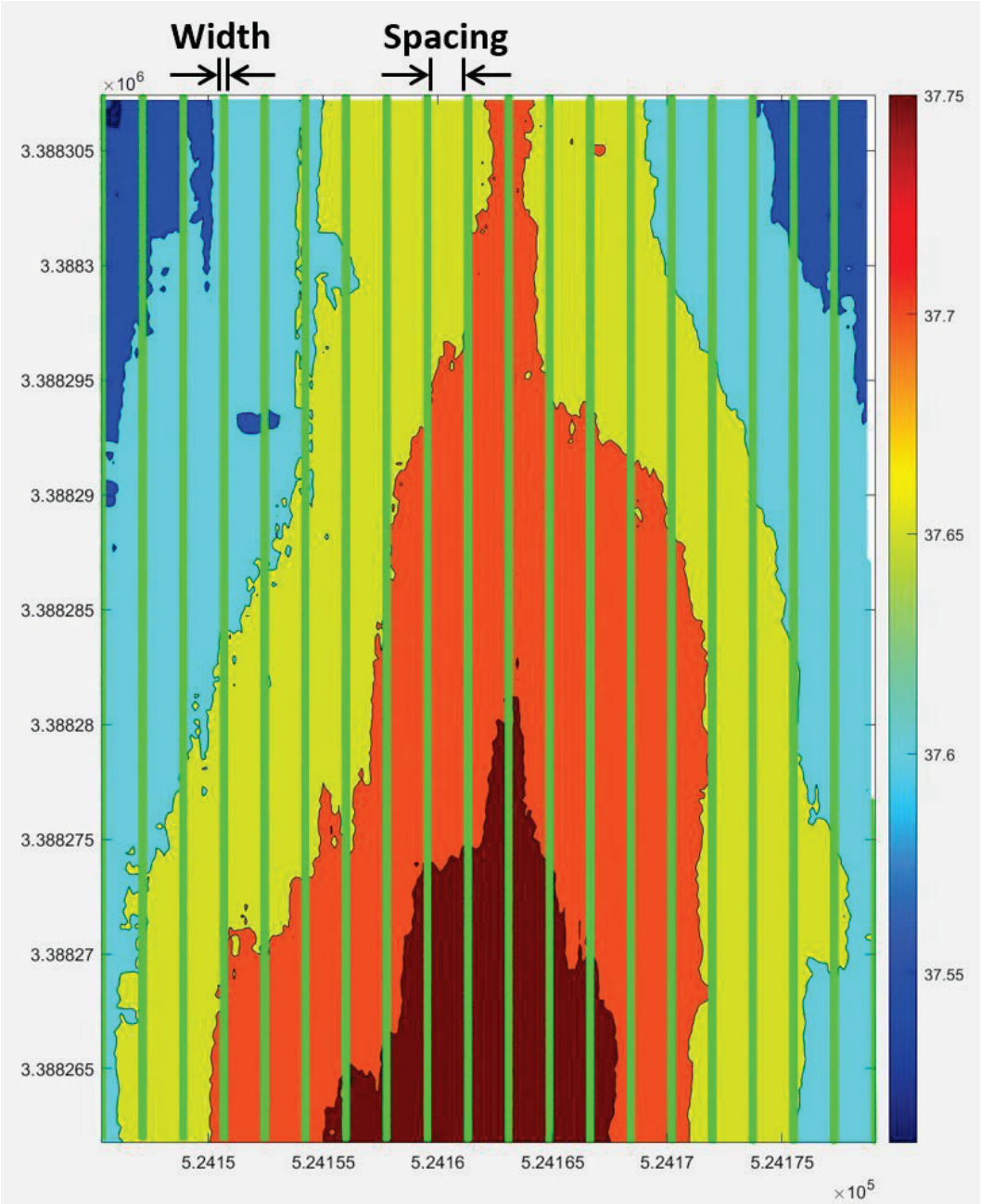
Once the 3-D point data are imported into the software, the user is able to generate custom cross sections under the "Create Cross-sections" menu (see Figure 84).

Figure 84. Cross-section menu from the z-deviation method GUI.



First, the user needs to specify spacing and width parameters before the cross sections are made. The spacing parameter defines the distance between successive cross sections, and the width parameter defines the width of individual cross sections (see Figure 85). Once that is done, the user should press the "Cross-Section Setup" button. This button allows the user to select a custom size and orientation cross section on the displayed post-surface. The GUI handles this process by converting the mouse cursor into a crosshair that the user controls. The user clicks the post-surface where they want the first cross section to start and stop. Finally, the user should press the "Create Cross-Section" button in order to generate cross sections along the entire surface. Upon pressing the "Create Cross-Section" button, the user is prompted to enter a name for the current series of cross sections. This name will be used to save the cross sections to file.

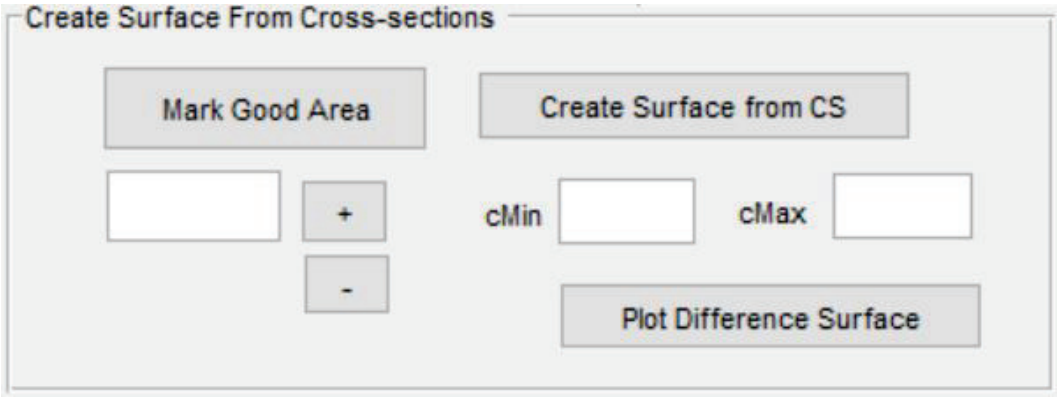
Figure 85. Width and spacing parameters for the z-deviation method GUI.



With the above process completed, the software is now ready to generate a pre-surface. This can be done in the "Create Surface From Cross-Sections" menu (see Figure 86). First the user presses the "Mark Good Area" button. This prompts the user to select the four corners of a clean area on the post-surface. This is done with the same crosshair used in the cross-section setup step. Once the four corners of a good area are selected, the user presses the "Create Surface from CS" button. This completes the pre-surface generation step. At this point, the software has access to an

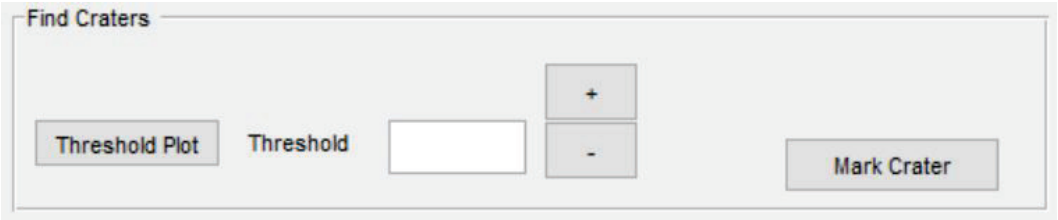
imported post-surface and a generated pre-surface of the same grid size, $N \times N$. With the generation of a pre-surface complete, the user is now able to plot and display a difference-surface mesh by pressing the "Plot Difference Surface" button (see Figure 86). This process automatically fills the "cMin" and "cMax" text boxes. These values represent the difference-surface min and max elevations. The values cMin and cMax can be modified; however, they are initialized at the absolute min and absolute max of the entire difference-surface mesh.

Figure 86. Create surface from cross-sections menu from the z-deviation method GUI.



The final step of the damage detection process is to pick a z-deviation threshold and apply it to the difference-surface mesh. This is first done by inputting a numerical threshold value for z-deviation in the "Threshold" text box in the "Find Craters" menu (see Figure 87).

Figure 87. Find craters menu from the z-deviation method GUI.



The z-deviation threshold value can be incremented by one tenth upon pressing the plus and minus buttons. By pressing the incrementing buttons, any detected craters are visualized by adding a black square at that location in the difference-surface plot (see Figure 88). At the same time, an Excel table is displayed and updated just below the surface plot area (see Figure 89).

Figure 88. Mesh plot with damage indicator boxes from the z-deviation method GUI.

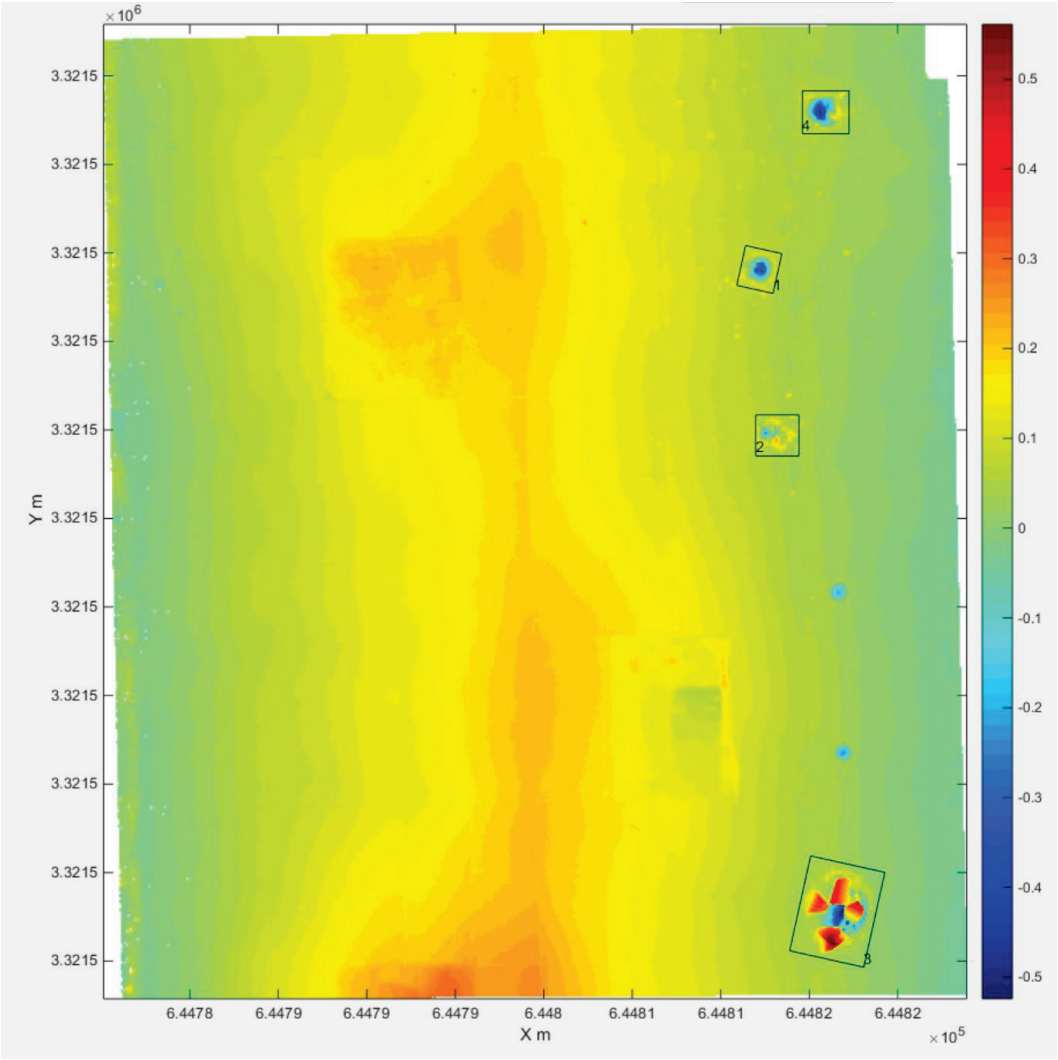


Figure 89. Excel table with damage locations from the z-deviation method GUI.

	Crater Number	Units	Width	Length	Area	Volume	Corner Point 1 X	Corner Point 1 Y	Corner Point 2 X	Corner Point 2 Y	Corner Poir
1	Crater 1	m	2.3035	2.0991	4.8353	4.2575	6.4481e+05	3.3215e+06	6.4481e+05	3.3215e+06	6.4481
2	Crater 2	m	2.4432	2.3160	5.6585	5.3860	6.4481e+05	3.3215e+06	6.4481e+05	3.3215e+06	6.4481
3	Crater 3	m	5.4875	4.2936	23.5614	46.0917	6.4482e+05	3.3215e+06	6.4482e+05	3.3215e+06	6.4482
4	Crater 4	m	2.6386	2.4263	6.4022	6.4853	6.4481e+05	3.3215e+06	6.4482e+05	3.3215e+06	6.4482

An appropriate threshold is found by targeting the z-deviation values associated with obvious craters in the difference-surface plot. Once a satisfactory threshold is found, various data can be saved using the save data functions to the right of the “Create Surface From Cross-sections” menu (see Figure 90). The “Print Screen” button saves the current GUI

screen as an image file with a user-input filename. The "Save Table" button saves the current damage location information to an Excel file with a user-input filename. The "Save Data" button is largely for developer use and saves the current software parameters into a MATLAB-compatible access database file.

Figure 90. Save data functions from the z-deviation method GUI.



4.3.8 Future improvements

While the above method is superseded by later work in regards to this effort, it is still important to provide potential improvements should this method be adopted for other future research needs.

A core limitation of the above-mentioned z-deviation method is the reliance on elevation change within a single grid. While this method is sufficient for larger craters with noticeable upheaval, it fails with shallow craters and craters with little upheaval. This failure occurs, largely, for two reasons, i.e., the smoothing effect of the grid-averaging mesh creation technique and the coarseness of the reconstructed mesh. Without adopting more time-consuming solutions such as using an interpolative mesh creation technique or increasing the mesh size, the method can be improved with better z-deviation scanning. Instead of scanning for elevation change within a single grid, one can scan a moving window of grids. This is difficult to do with large, unorganized 3-D point clouds, but relatively easy to do with an organized coarse mesh such as the ones representing the pre- and post-surface. Such a change in z-deviation scanning would have a number of effects.

1. The larger elevation change present in deep and shallow craters would be more noticeable.
2. More false positives would be picked up as surface geometry can change elevation over larger areas.
3. The z-deviation threshold used can be made stricter, as the difference between the lip and bottom of a crater is much more than that of coarse surface geometry.

With the above-mentioned improvement, the z-deviation technique could be used to measure surface changes for a variety of applications. The technique is principally simple and easy to implement computationally. However, without the use of a very accurate pre-surface, this technique will continue to have poor performance with finding smaller surface damage such as camouflets and spalls.

4.4 Depression-filling method

The problem of measuring pavement surface damage can be summarized as an exercise in quantifying surface depressions. For the project at hand, team members take these surface depressions to be the result of unwanted detonations on an airfield pavement surface. As such, a method for efficiently detecting and classifying surface depressions is desired. Based on this and the inherent limitations of the z-deviation method detailed in the preceding section, a second approach to damage detection was considered that targeted surface depressions.

Depression-filling algorithms are used in a variety of fields including geosciences and astrophysics. In the geosciences community, depression-filling algorithms are used to label watersheds (Vincent and Soille 1991), determine flood risk (Barnes et al. 2014), and measure surface depression storage capacity (Huang and Bradford 1990). In the astrophysics community, depression-filling algorithms are used to monitor cratering on distant astronomical bodies (Yin et al. 2013). In effect, the problem of detecting and classifying very large craters on distant bodies is relatively the same as detecting and classifying small craters on nearby surfaces.

4.4.1 Algorithm introduction

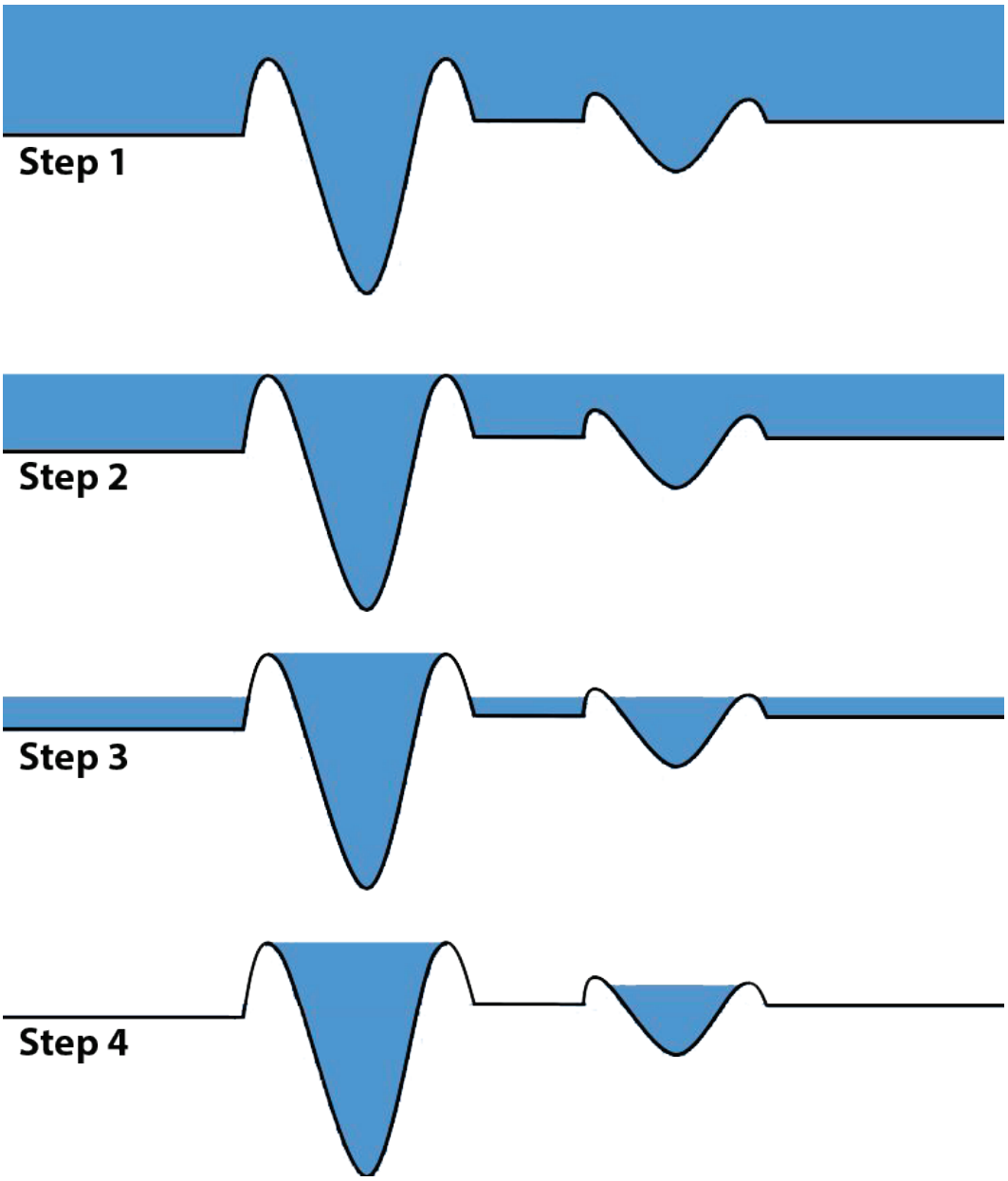
Following a rigorous literature review, it was found that while many proposed methods exist for the detection of negative volumes on a surface, few converge to a unique solution for all surface types. A foundational

paper by Planchon and Darboux (2001) provides a depression-filling algorithm that both converges to a unique solution for all surfaces and does so with time complexity suitable for modern, high-resolution meshes. While the seminal work done by Planchon and Darboux (2001) influenced others to modernize and speed up their method, it is found that the original Planchon and Darboux (2001) algorithm is efficient enough for the present work. For the remainder of this section, the original Planchon and Darboux algorithm will be referred to as "no-sink."

4.4.2 Initialization

The no-sink algorithm works, in a subjective sense, by flooding a surface with water and draining the water in a controlled manner (see Figure 91). It is imperative then that the surface has no holes in it. This presents a problem as the output of the LiDAR sensor is a series of unorganized three-dimensional point clouds with unfilled gaps between successive points. For input into no-sink, these 3-D point clouds must be converted into a rigid surface with all gaps filled. This is done using the grid-averaging method detailed in Section 4.1.

Figure 91. No-sink algorithm flooding example.



4.4.3 Surface flooding

The no-sink depression-filling approach in effect floods a surface with "water" and iteratively decreases the height of the "water" until all depressions are filled. While there are numerous implementations of depression-filling type algorithms, all typically follow three basic rules while flooding the surface (Planchon and Darboux 2001).

1. The final surface, S_2, is at all grid cells greater than or equal to the original surface, S_1, i.e., at all points $S_2 \geq S_1$.

2. For all grid cells in the final surface, there exists a path that leads to the boundaries in which there is a minimum descent of ε.
3. The final surface is the lowest elevation surface by which the above two properties apply.

The descent value, ε, is taken by default to be zero. That is, all depressions should be filled with flat "water." The final surface, S_2, is initialized at infinity everywhere but at the boundary. Computationally speaking, this means that the elevation of S_2 is initially set to a number higher than the maximum z-value of the original surface, S_1. The boundary region of S_2 is set to the same elevation as the boundary region of S_1. In this way, the border region of S_2 starts off fully drained and, therefore, its final elevation is already known. With that, S_2 now represents a fully flooded surface and must be iteratively drained to reveal a depression-filled surface defined by the three rules above.

4.4.4 Surface draining

Keeping with the analogy of flooding a surface with water, the surface draining step can also be seen as the drying of water. The surface draining step consists of iteratively decreasing the elevation of the final surface, S_2, until all three rules are met. As the border region of S_2 is known and fixed to the values from S_1, the iterative scan can begin at any point in the border region. In order to ensure complete coverage of the surface for proper draining, eight directional scans are conducted (see Table 5).

Table 5. No-sink surface draining scan directions.

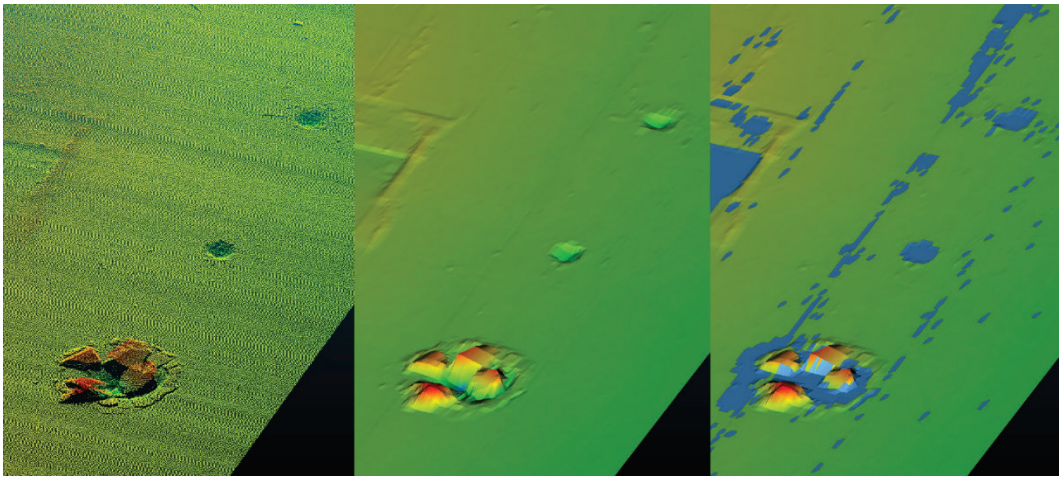
Starting Position	Direction	Reset Position
Top Left	Rightwards	Next Down, First
Bottom Right	Leftwards	Next Up, Last
Top Right	Down	First, Next Left
Bottom Left	Up	Last, Next Right
Top Right	Leftwards	Next Down, Last
Bottom Left	Rightwards	Next Up, First
Top Left	Down	First, Next Right
Bottom Right	Up	Last, Next Left

The scan directions detailed in Table 5 yield various scan lines one grid wide with a length stretching between two boundaries. In order to satisfy rule two, each scan must create or find an ε-descending path between the two boundaries. This can be done by ensuring that a smaller ε-descending path exists between every query grid and its eight neighbors.

The first drying action takes place during the eight-neighbor search. If any neighbor of the query grid contains water of a lower elevation than the original surface of the query grid, then that neighbor grid is dried of water back to its original surface elevation. Should this not be the case, an attempt is made to level the water of the query grid and the neighboring grids. If the water level of the query grid is greater than that of any neighboring grid, then the query grid's water level is reduced to the level of the neighboring grid. This process is repeated for all grids in the current scan stopping just short of the boundary. Once the scan is complete, another is chosen from Table 5.

There is a directional dependence on the individual scans referenced in Table 5. For instance, if a crater has high upheaval on the side of the scan direction but lower upheaval on the side perpendicular to the scan direction, then the first scan would result in a higher water level than the scan in the perpendicular direction. It is for this reason that the surface is scanned eight times. The result of these scans is the lowest possible surface satisfying rules one and two (see Figure 92).

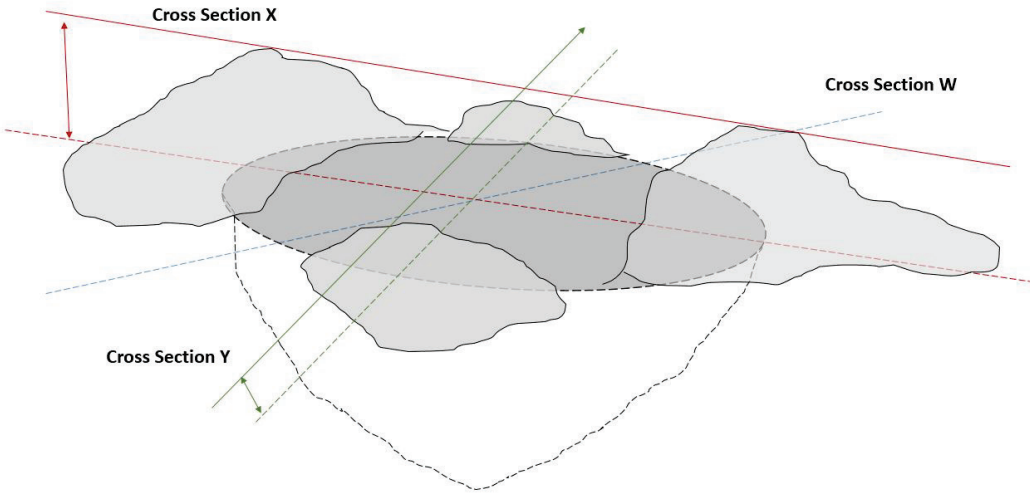
Figure 92. Depression-filling example on damaged pavement surface.



*Left, point cloud; middle, 500x500 mesh; right, depression-filled mesh

It can be seen in Figure 92 that, despite cratering with uneven upheaval, the no-sink algorithm filled the inside of the craters with water such that the water is allowed to "drain" out following subsequent scans (see Table 5). Smaller depressions on the surface are also filled with water; however, they are much shallower than the damage of interest and can be easily ignored. In order to better visualize this, let us consider a fictional crater with larger upheaval in the $\vec{x}$-direction than in the $\vec{y}$-direction with a vector $\vec{w}$ representing the pavement surface ground level (see Figure 93). With such a crater, a cross section can be drawn from the top of the $\vec{x}$-direction upheaval to $\vec{w}$ and another from the $\vec{y}$-direction upheaval to $\vec{w}$. Given a single surface draining scan in the $\vec{x}$-direction, the water level would be as high as the $\vec{x}$ cross section upheaval. This level of water is not indicative of the actual damage depression. Given a second scan in the $\vec{y}$-direction, the water level would be drained down to the $\vec{y}$ cross section upheaval. Subsequent scans in other directions would further drain the water to a level more indicative of the actual surface damage.

Figure 93. Cross-section view of a fictional crater for the no-sink method.

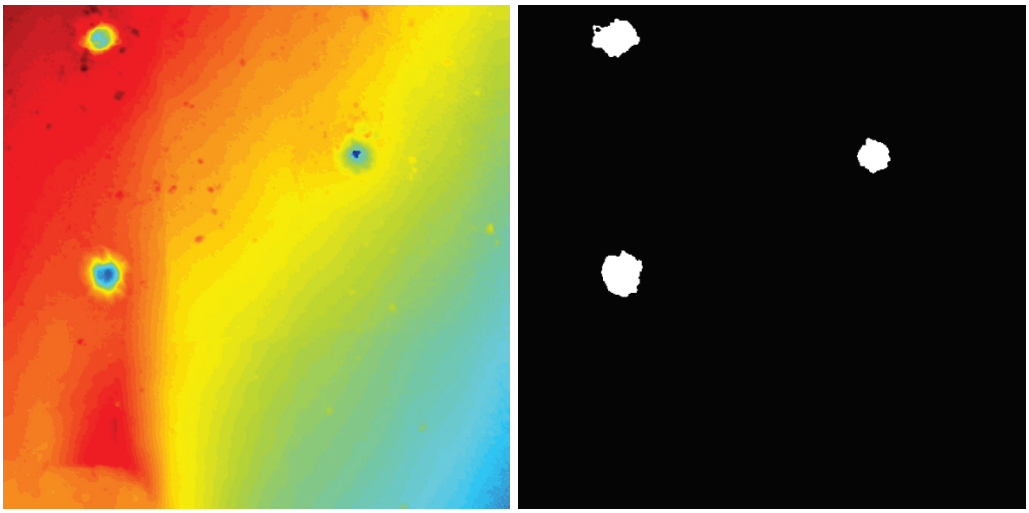


4.4.5 Isolating depressions

Once the no-sink method converges to a final depression-filled surface, the original surface is removed leaving only the depressions. One way to visualize this is through the use of a binary filter. A depth threshold is set and any grid not meeting this threshold is colored black and the rest, namely depressions, are colored white (see Figure 94). It is noteworthy that in Figure 94 the surface surrounding the three depressions was not

flat; however, the binary image resulting from the binary filter is perfectly flat everywhere but at the depressions. The result of this difference surface is a very well-defined ground plane that is featureless except for surface depressions. This ground plane serves as a reference plane from which damage-depth measurements are made.

Figure 94. Before and after crater field using a depression-filling algorithm.



a) Before depression-filling method b) After depression-filling method

4.4.6 Damage classification

The various damage seen following detonation on a pavement surface can be summarized in the following three categories, i.e., craters, camouflets, and spalls. Craters present as bowl-shaped depressions in the surface. It is common for craters to appear with debris present in the bowl along with ejecta surrounding the lip or upheaval. Due to the acute angularity of the crater's bowl shape, it is expected that aerial LiDAR measurements will fully map the extent of the crater. On the other hand, camouflets present when ordnance penetrates the pavement surface and creates an artificial void or cavern below the surface. In this way, the only visible sign of a camouflet on the surface is a small hole the size of the penetrating projectile. As such, an aerial LiDAR data collection system would map only the projectile's entrance cavity and not the larger void below the surface. Finally, spalls are present with the same acute angularity of craters but at much shallower depths. It is common in spalls for most debris to be ejected from the surface depression.

Considering the appearance of airfield damage described above, the following three statements can be made.

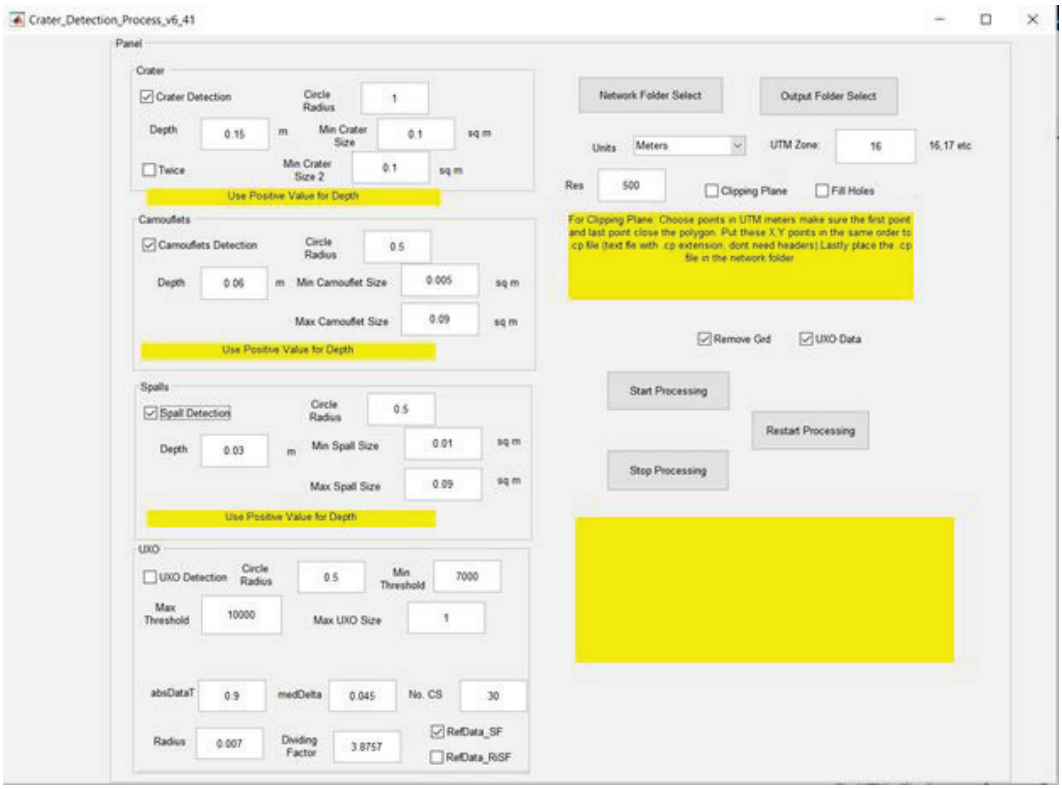
1. Of the three types of damage, craters will present with the largest cross-sectional surface area and will be deeper than or as deep as camouflets.
2. Camouflets will have a much smaller cross-sectional surface area than craters, but with as much depth.
3. Spalls will have the least depth and cross-sectional surface area of all three types of damage.

It is with these considerations in mind that the search for damage using the no-sink method described in the preceding sections can begin. The result of the no-sink algorithm is a well-defined ground plane and well-defined depressions. As such, pavement surface damage revealed by the no-sink method can first be delineated by depth from the ground plane and second by the cross-sectional surface area of isolated depressions.

4.4.7 MATLAB implementation

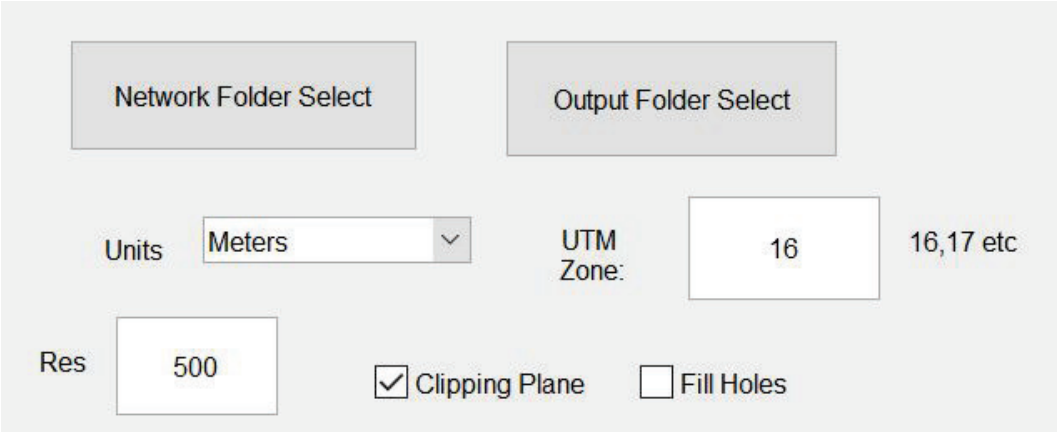
The GUI developed to demonstrate this approach (see Figure 95) was designed to provide users with the ability to customize the definitions of craters, spalls, and camouflets. In this way, airfields with varying pavement layer thickness can meet their damage definition requirements without the need to recompile the software.

Figure 95. MATLAB-developed GUI for method three.



The first step in damage detection initialization is defining data input/output streams. In the damage detection GUI, output and input data streams are defined by local and/or network folders. LAS data is input in the GUI by the selection of a watch folder. The software searches the input watch folder for added files and processes them in order. The output folder is selected as the destination for output shape files. The user selects the LAS input data folder using the "Network Folder Select" button (see Figure 96). This presumes that the LAS data is collected and processed on another machine and is transferred to the processing machine over a network. The user selects the output data folder using the "Output Folder Select" button (see Figure 96).

Figure 96. GUI options for initialization steps one, two, and three.



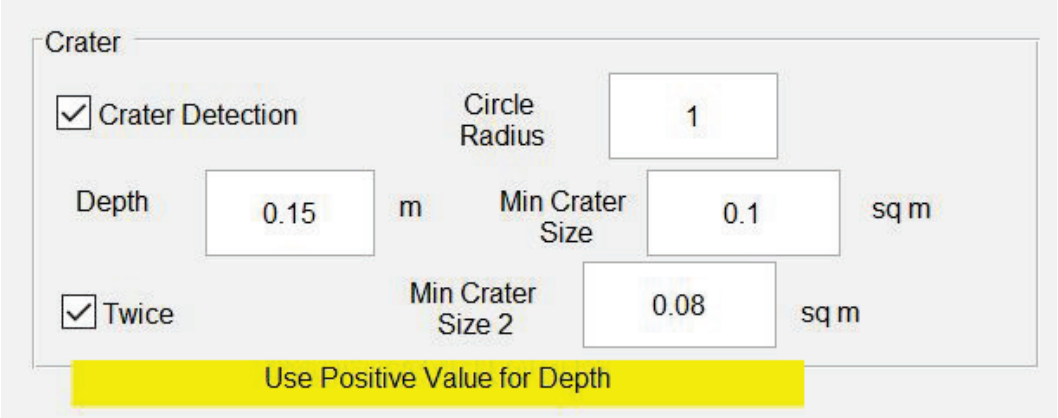
Secondly, the user must consider the coordinate system in which the LiDAR data is collected. The processing algorithms within the damage detection software require the use of a projected global coordinate system such as UTM. The UTM coordinate system estimates Earth's surface as an ideal ellipsoid and breaks it into 60 zones. These zones are then projected onto a flat plane. This system is ideal for the purposes of damage detection and classification as it allows for the use of length units (feet, meters) directly. As such, LiDAR data collected using the UTM coordinate system allows for engineering measurements to be made directly on easting and northing data. The user has the option of selecting UTM meters or UTM feet as well as choosing the specific northern UTM zone (see Figure 96).

The third step of the initialization process requires the user to select a mesh resolution to be used during the mesh creation step described in Section 4.4.2. It is at this time that the user has the option of selecting a clipping plane that describes the exact polygonal area to be scanned. This provides the user with the option of ignoring infield or grassy areas that are not of interest. The clipping plane file must consist of clockwise sequential x-y pairs that enclose the surface of interest. The clipping plane file can be arbitrarily named; however, it must be saved using the ".cp" extension and placed in the input folder in order for the software to find it. To enable the clipping plane feature, the user must select the "Clipping Plane" checkbox (see Figure 96).

The fourth step of initialization is the most important and encompasses the defining of damage feature variables. The user first selects the left-side checkboxes for each damage feature for which they intend to search. Crater detection has an optional second run in the event the user wants to

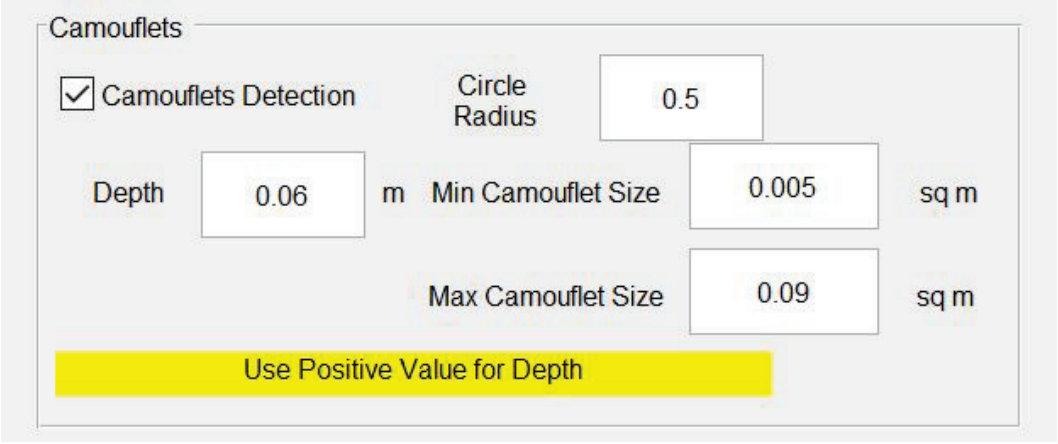
define two distinct sizes of craters as opposed to one. Should this be the case, the user must select the "Twice" checkbox and define a second max crater area. In the "Crater" section of the GUI (Figure 97), the user must input a depth value in units of meters. This is the depth at which the software will search for contours representing the crater. The user must also input the minimum crater area. Finally, a circle radius, in units of meters, is input by the user that represents the radius of a circle to be drawn around the damage when generating shape files.

Figure 97. Crater section of the damage detection GUI.



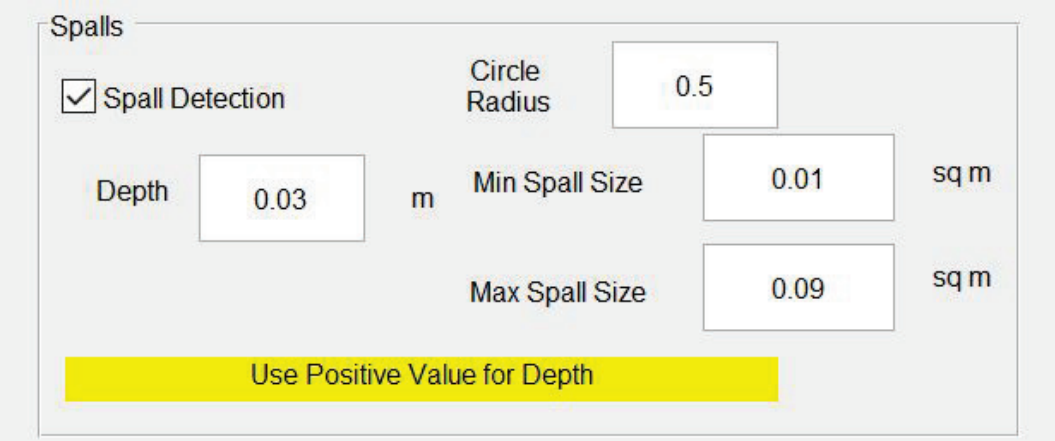
In order to detect camouflets, the user must select the "Camouflets Detection" checkbox and input some initial values in that section of the GUI (see Figure 98). Much like the crater detection step, a depth value and crater circle radius must be input. Additionally, the min and max area of the camouflet must be input into their respective text boxes.

Figure 98. Camouflet section of the damage detection GUI.



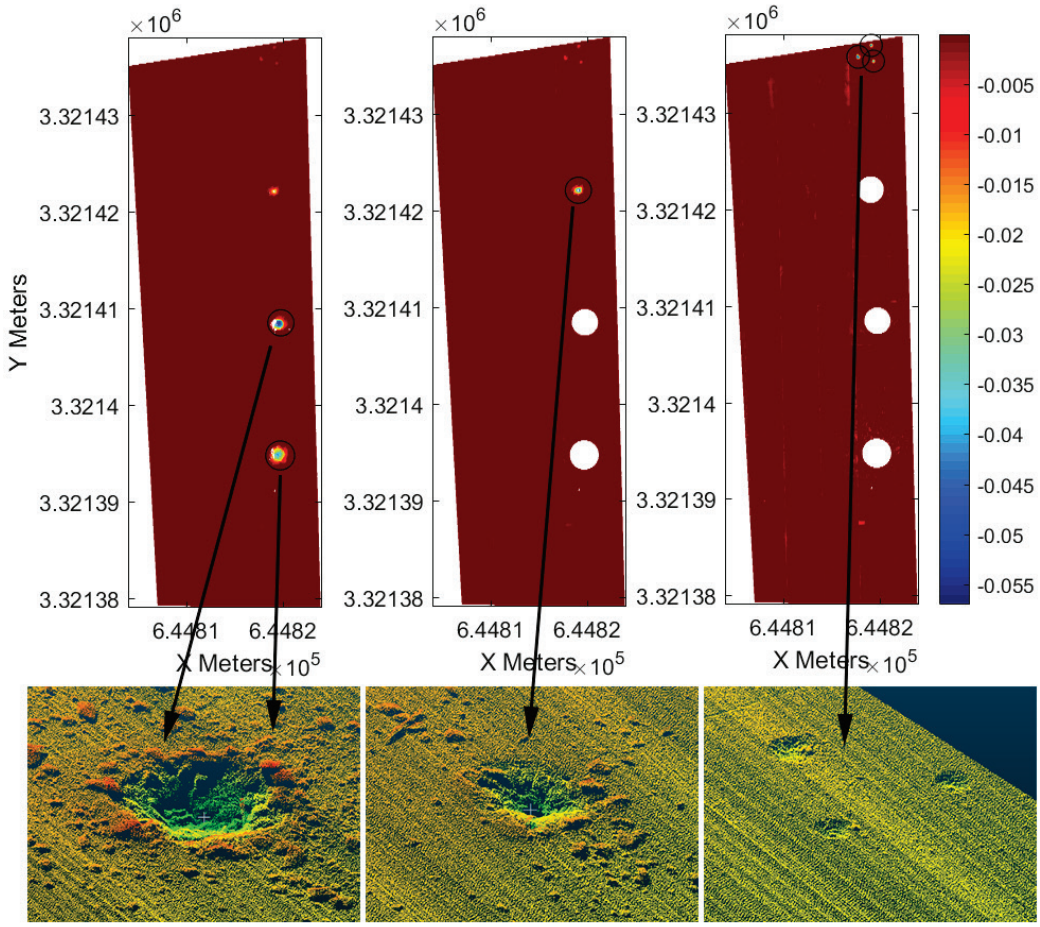
Finally, spall detection is done by selecting the "Spall Detection" checkbox and inputting the same initial values described above, i.e., depth, circle radius, and min and max areas (see Figure 99).

Figure 99. Spall section of the damage detection GUI.



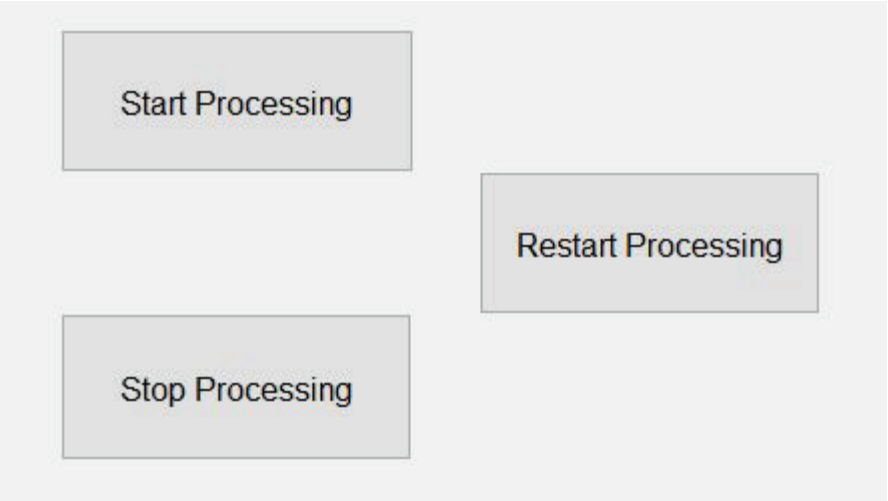
As one may infer from the initialization process, the damage detection software works in order of largest damage to smallest (see Figure 100). First, craters detected using the aforementioned initialization parameters are documented and removed from the surface. The new surface is then used as the input for the remaining steps, i.e., optional crater run, camouflet run, and spall run.

Figure 100. Large to small depression-filling damage classification.



Once users have input the necessary initialization values, they need only press the "Start Processing" button to begin the damage detection process (see Figure 101). The starting process can take place before or after data are available. If completed LiDAR data are available in the input folder, the software will start processing them in order. Should no data be available at start, the software iteratively searches the input folder until a completed LAS file exists. The damage detection process can be stopped by pressing the "Stop Processing" button; however, the stopping action may be delayed should the program be busy with an underlying function. Should the user choose to restart all processing and rerun the program on existing LAS files, the "Restart Processing" button can be pressed.

Figure 101. Start/stop section of the damage detection GUI.



Upon starting the damage detection process, the software will save, as it becomes available, the difference surface plots described in Section 4.4.5 and seen in Figure 100 as well as multiple Excel files (see Figure 102) where the damage location and various shape information is stored.

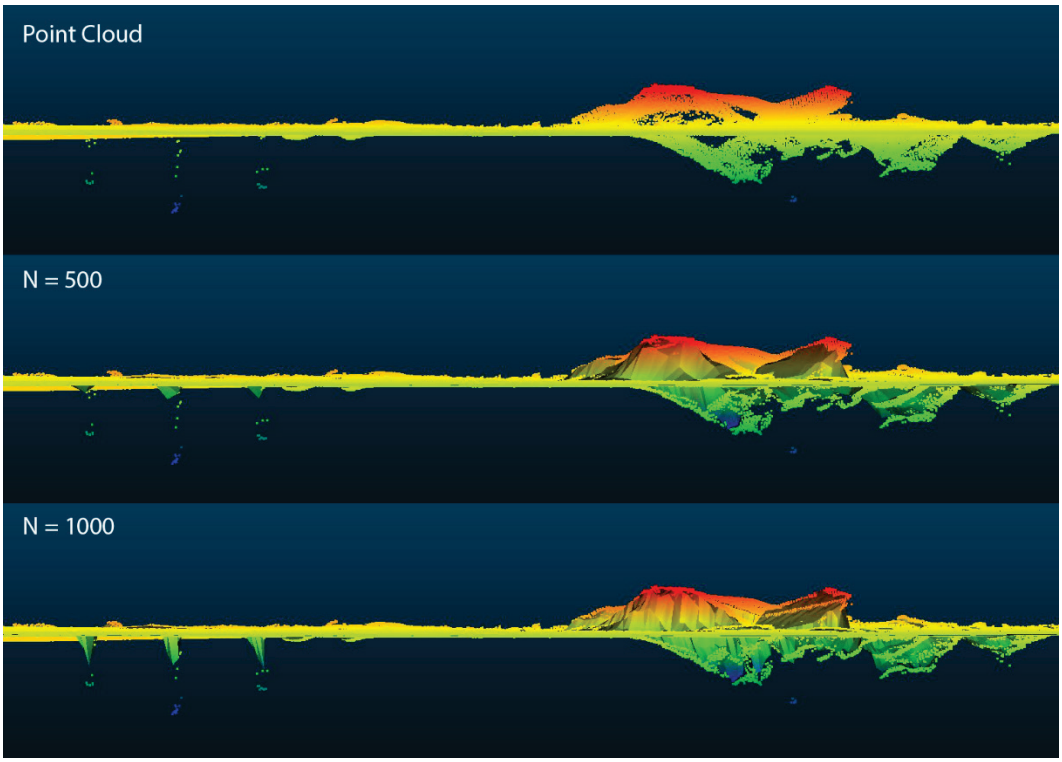
Figure 102. Excel file output from the damage detection GUI.

	A	B	C	D	E	F	G	H
1	Crater Nu	Units	Position X	Position Y	Radius	Area	Volume	
2	Crater 1	Meters	644819.684064	3321394.787615	1.500818	7.076299	8.496192	
3	Crater 2	Meters	644819.757209	3321408.512328	1.358176	5.795118	6.296634	
4								

4.4.8 Future improvements

One detriment to the surface reconstruction techniques referenced in Section 4.4.2 is their insensitivity to areas of low point density. For instance, if a camouflet exists where few points are located at the true depth of the damage, the aforementioned techniques will largely ignore these points and reconstruct a much shallower surface where the majority of the points are located. A comparison of 500 × 500 and 1000 × 1000 grid sizes can be seen in Figure 103.

Figure 103. Profile view of a point cloud and corresponding surface reconstructions.



*In the figure, N denotes grid size, $N \times N$

The figure shows a profile view of a sample point cloud with three camouflets on the left side and three craters on the right side. The true depth of the various damage can be noted by the lowest (green and blue) points in the image. When the surface is reconstructed with a 500 × 500 grid, the camouflets appear much shallower than they actually are in the point cloud view. This is a result of not only the grid-averaging method used but also the coarseness of the mesh grid size. The bottom of the figure shows a reconstruction on the same point cloud with a 1000 × 1000 grid size. It can be seen from the 1000 × 1000 reconstruction that the true depth of the camouflet is more closely modeled.

It is from this limitation that improvements can be made to the damage detection algorithm. First, larger grid sizes can be used in order to better simulate the true depth of damage at the expense of processing time. Second, a small grid size can be used to locate damage, and the original point cloud could be queried at these locations to make measurements at the true depth.

5 Validation of Damage Detection

5.1 Depression-filling analysis for damage detection

5.1.1 Development of LiDAR tiles for analysis

As described in Chapter 4, the depression-filling method is the most reliable pavement damage detection system currently developed by ERDC. Its development matured over the intervening data collection test events between 2016 and 2019. To validate the overall performance of the final depression-filling algorithm, a series of LiDAR tiles were constructed from the various test events in which LiDAR data were collected on airfield damage summarized in Table 6.

Table 6. LiDAR data collection events used in validation study.

Date of Event		LiDAR Technique	Altitude (ft)	Craters	Number of Spalls	Camoflets
August	2016	Mobile-Ground/Historical	8	6	0	0
December	2016	Mobile-Ground	8	6	2	2
August	2017	Mobile-Telehandler	50	12	4	4
December	2017	Mobile-Telehandler [Smoke]	50	2	0	1
March	2019	UAV-RiCopter	50	7	2	6
		UAV-RiCopter	100	7	2	6
		UAV-RiCopter	200	7	2	6
		UAV-RiCopter	260	7	2	6
		UAV-RiCopter	330	7	2	6
		UAV-RiCopter	400	7	2	6
		UAV-RiCopter [Night]	50	7	2	7

The depression-filling algorithm achieves its optimal performance when the 500 x 500 grid mesh is able to be distributed evenly over the entire LiDAR surface, maximizing the number of points available within each square of the mesh. Therefore, each runway test section scanned with LiDAR was manually subdivided into square tiles using a COTS point cloud visualization software called CloudCompare with a length approximately equal to the width of the scan or width of the runway, whichever was smaller. This resulted in each test event being described in 4 to 6 tiles. Further, each LiDAR data set was trimmed to remove vegetation on the side of the runway. At least one tile from each LiDAR collection event was created where no damage was present to represent a null case to test the algorithms ability to avoid false positives.

The following figures illustrate the LiDAR tiles created from each test event with labels as to the approximate location of each key damage for reference. Circles are crater locations, squares are spall locations, and triangles are camouflet locations.

The MATLAB DFA was run on each tile to determine the success of identifying the three damage features present. A table is presented in each subsequent section summarizing the number of features detected against the number of features actually present in each tile. Appendix A displays the actual screen captures taken from the MATLAB software from which these detection locations were observed and denoted. This allows the data in each table to be cross-referenced with the screen shots shown in Appendix A to visualize the accuracy of both the location and geopositioning of each damage feature.

5.1.2 August 2016 – artificial airfield

The LiDAR point clouds merged from the five individual crater blasts were segmented into five analysis tiles with each tile containing one crater (Figure 104). The DFA was run, and the results are shown in Table 7. The detection of craters was the primary purpose for this artificial surface, and the DFA was able to successfully identify the crater in each tile with no false positives. However, in Tiles 2, 3, and 4, there were a number of cored holes on the test site but whose origin and intent were unknown. An estimate of the number of camouflet holes was made from the LiDAR point cloud since no documentation of these test sites is available. When the DFA was applied to these tiles, the hole placement was significantly affected by crater upheaval, causing multiple false positives and missed locations. Given the complex layout of the cored holes, the DFA identify the majority of the camouflet holes that in a real scenario would guide mitigation teams to the correct locations of damage. The DFA also returned a number of spall fields from various damaged portions of the test sites from prior testing. The significance of Table 7 is that craters were successfully detected with no false positives.

Figure 104. August 2016 reference LiDAR point cloud tiles for analysis.

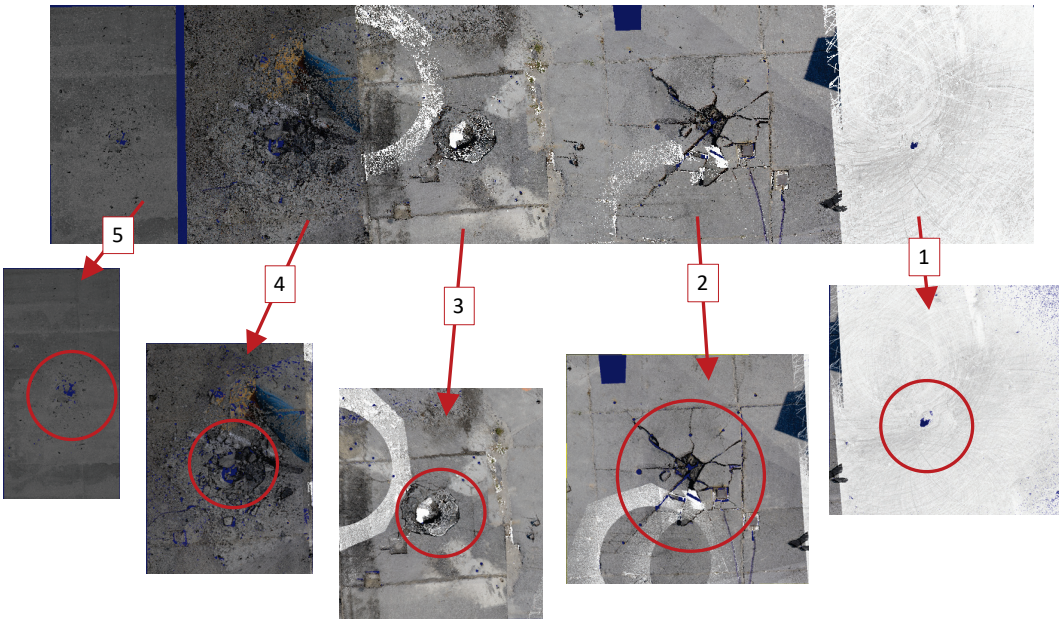


Table 7. Damage detection summary for August 2016 – artificial airfield.

August 2016	Number of Craters			Number of Camoflets			Number of Spall Fields		
	Actual	Detected	False Positives	Actual	Detected	False Positives	Actual	Detected	False Positives
Tile 1	1	1	No	0	2	2	0	0	0
Tile 2	1	1	No	10	25	Multiple	0	15	15
Tile 3	1	1	No	24	18	Multiple	0	14	14
Tile 4	1	1	No	12	17	Multiple	0	13	13
Tile 5	1	1	No	0	1	1	0	3	3

5.1.3 December 2016 – 6 crater event

This test event represented LiDAR taken at a very low grazing angle to the runway surface as the VZ-2000 LiDAR was suspended about 8 ft. off the ground surface. Significant LiDAR shadowing was observed in the scenes as any debris or crater upheaval blocked the penetration of LiDAR into the crater, spall, and camouflet locations. The test section was divided into four tiles with Tiles 2 and 3 containing pavement damage and 1 and 4 being empty (Figure 105). The DFA was applied to these tiles, and the resultant detections are shown in Table 8. What is readily apparent is that very little damage could be detected. There is not enough vertical relief that can be detected by the DFA to provide depression filling in the craters, spalls, and camouflets due to the low grazing angle. The DFA did not pick up any false positives on the clean area suggesting that, in areas where no damage is present, the DFA will not provide false positives reducing the mitigation area for repair. This

data justifies the need for a more normal LiDAR collection to penetrate below the pavement surface and be unobstructed by debris and upheaval.

Figure 105. December 2016 reference LiDAR point cloud tiles for analysis.

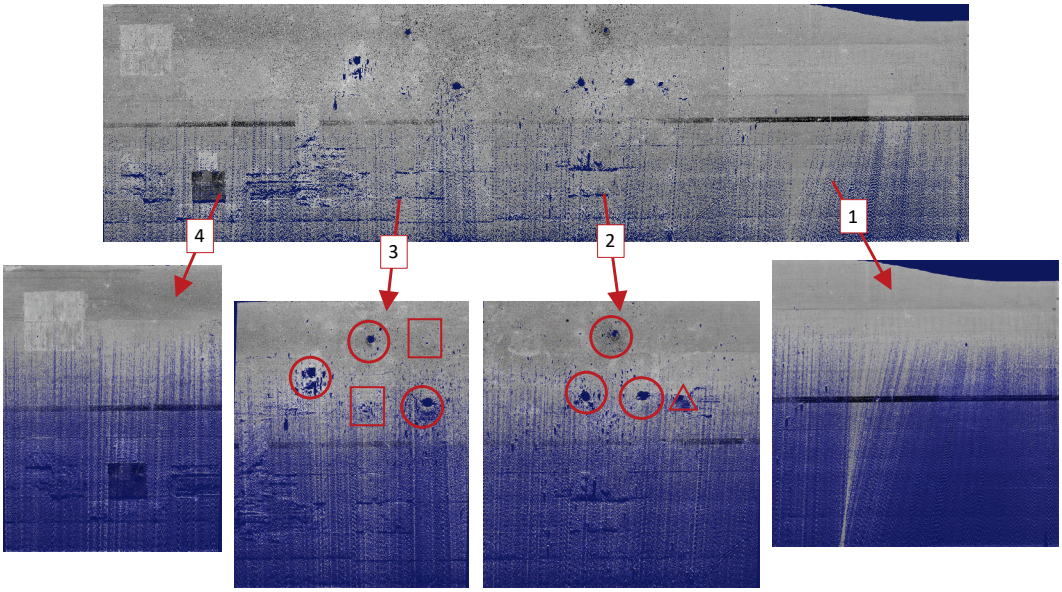


Table 8. Damage detection summary for December 2016.

December 2016	Number of Craters			Number of Camoflets			Number of Spall Fields		
	Actual	Detected	False Positives	Actual	Detected	False Positives	Actual	Detected	False Positives
Tile 1	0	0	No	0	0	No	0	0	No
Tile 2	3	0	No	1	1	1	0	0	1
Tile 3	3	1	No	0	0	No	2	1	1
Tile 4	0	0	No	0	0	No	0	0	No

5.1.4 August 2017 – 12 crater event

The August 2017 test event was the first to use an overhead LiDAR sensor, with the telehandler providing a near normal perspective of the runway surface. The blast event covered a much larger area of the runway than prior data collection efforts and required dividing the analysis effort into 6 square tiles (Figure 106). Tile 3 was reserved for testing the capability of the DFA to not create false positives on undamaged pavement. The remaining tiles all consisted of at least one damage type and, in some instances, all three. From the analysis results shown in Table 9 it can be quickly seen the advantage that a normal LiDAR data collect can have on the success of the DFA. Nearly every intended damage feature was detected with no false positives with the exception of Tile 1 where one of the spall fields was misplaced, resulting in a miss and a false positive. This

level of accuracy exceeds that from the prior data collections, which relied on low height LiDAR collection and limited the functionality of the DFA.

Figure 106. August 2017 reference LiDAR point cloud tiles for analysis.

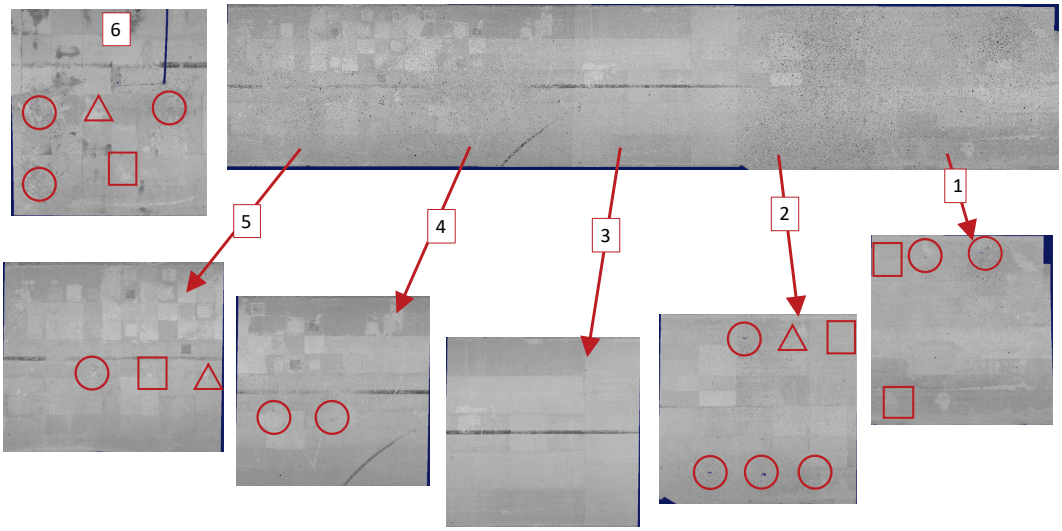


Table 9. Damage detection summary for August 2017.

August 2017	Number of Craters			Number of Camoflets			Number of Spall Fields		
	Actual	Detected	False Positives	Actual	Detected	False Positives	Actual	Detected	False Positives
Tile 1	2	2	No	0	0	No	2	2	1
Tile 2	4	4	No	1	1	No	1	1	No
Tile 3	0	0	No	0	0	No	0	0	No
Tile 4	2	2	No	0	0	No	0	0	No
Tile 5	1	1	No	1	1	No	1	1	No
Tile 6	3	3	No	1	1	No	1	1	No

5.1.5 December 2017 – 2 crater obscurant event

The December 2017 test event was the first collection to involve the use of obscurants and represented a test of the elevated LiDAR sensor to penetrate a smoke cloud and still detect surface damage. The blast event covered a smaller area of the runway than the prior data collection efforts, requiring only a single tile for analysis (Figure 107). Tile 1 contains two craters and single camouflet from the second day of testing at the site. From the analysis results shown in Table 10, it can be seen that the sample LiDAR point cloud beneath the smoke cloud enabled detection of each damage feature with no false positives. This suggests that during an operational attack on an airfield when cratering produces copious amounts of obscurants, a LiDAR sensor can still receive enough laser returns to map the surface accurately for the DFA to function properly.

Figure 107. December 2017 reference LiDAR point cloud tile for analysis (with smoke).

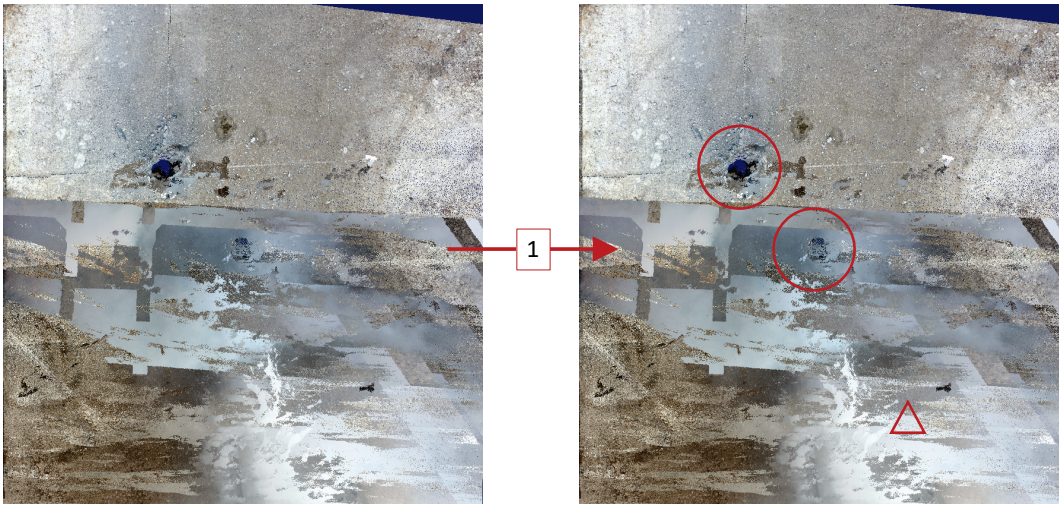


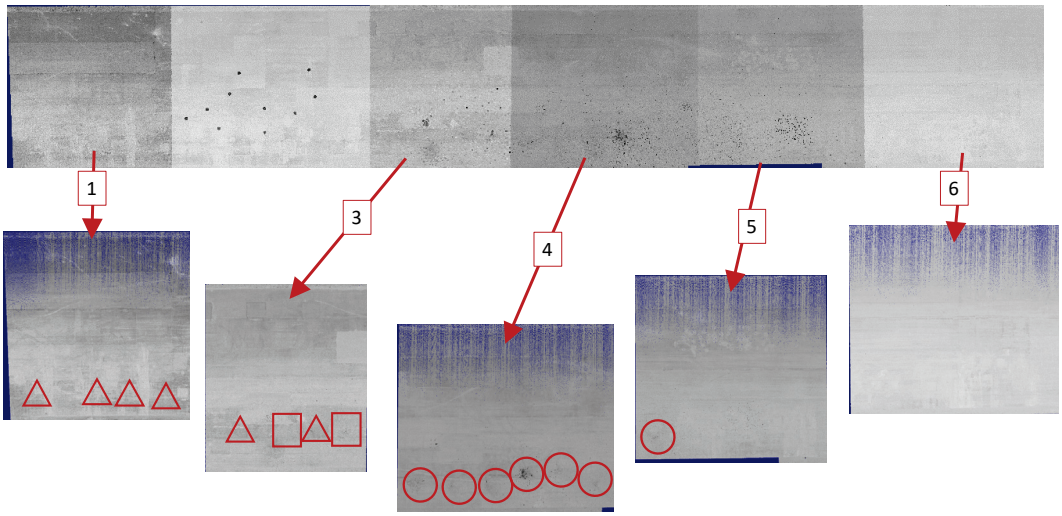
Table 10. Damage detection summary for December 2017 obscurant event.

Dec-17 Smoke	Number of Craters			Number of Camoflets			Number of Spall Fields		
	Actual	Detected	False Positives	Actual	Detected	False Positives	Actual	Detected	False Positives
Tile 1	2	2	No	1	1	No	0	0	No

5.1.6 March 2019 – 7 crater event

The March 2019 data collection was the first event in which UAS LiDAR was collected providing the valuable normal data collection perspective identified as critical in the August 2017 event. This test event covered a large area of the runway requiring six tiles to capture the damaged area (Figure 108). Tile 6 represented the undamaged tile to validate the DFA's ability to minimize false positives from UAS LiDAR. One tile was left off from analysis (the unlabeled Tile 2) due to the fact it had multiple tires placed on the runway surface along with a series of camouflets that were not representative of a damaged runway surface.

Figure 108. March 2019 reference LiDAR point cloud tiles for analysis.



The data analysis was performed on each combination of flight altitude and speed as shown earlier in Table 4. The first combination of 50-ft altitude and 5 m/s had the highest point density and resulted in accurately capturing all the damage with no false positives (Table 11). Subsequent combinations of altitude and air speed resulted in lower point cloud densities and an increase in the error in identifying damage.

Table 11. Damage detection summary for March 2019 (50ft-5m/s).

March 2019	Number of Craters			Number of Camoflets			Number of Spall Fields		
50ft 5m/s	Actual	Detected	False Positives	Actual	Detected	False Positives	Actual	Detected	False Positives
Tile 1	0	0	No	4	4	No	0	0	No
Tile 3	0	0	No	2	2	No	2	2	No
Tile 4	6	6	No	0	0	No	0	0	No
Tile 5	1	1	No	0	0	No	0	0	No
Tile 6	0	0	No	0	0	No	0	0	No

5.1.6.1 Crater detection with altitude

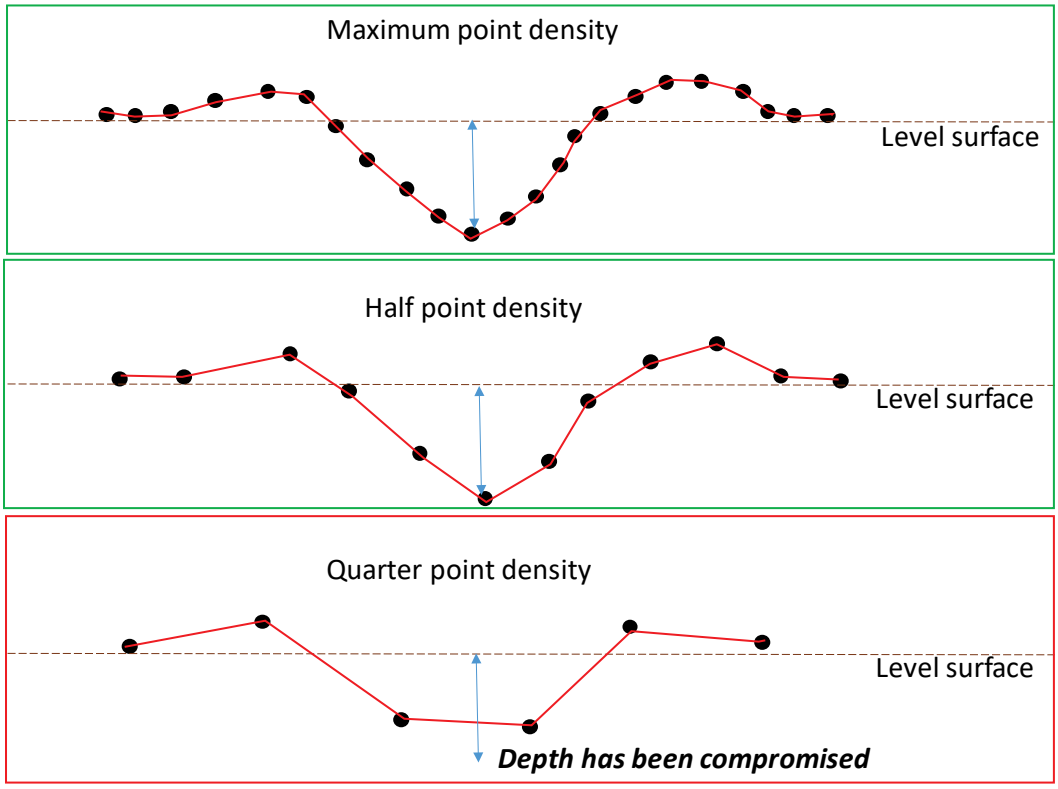
Table 12 shows that for crater detection, no false positives occurred owing to the robustness of the DFA to identify only large damage features. As the point density decreased coinciding with increase in UAS altitude, the number of craters that could be detected decreased. However, this effect is not due to any error in the DFA, simply that, as the number of points is reduced in the point cloud, the less data present in which to generate an accurate mesh from which to measure depth. Figure 109 illustrates what occurs in the mesh generation that lessens the crater depth with reduction in point density. In the MATLAB GUI, the user selects the absolute depth to which a damage feature is labeled a crater. If this depth criteria is not

met, then the algorithm assumes no crater is present. This behavior can be seen in the 330- and 400-ft elevation flights. By reducing the depth to which a crater is classified, then all craters can again be found.

Table 12. Summary of crater detection for all RiCopter flight lines.

March 2019	Number of Craters													
		50ft-5m/s		50ft-9m/s		100ft-5m/s		200ft-5m/s		260ft-5m/s		330ft-5m/s		400ft-5m/s	
	Actual	DET	FP	DET	FP	DET	FP	DET	FP	DET	FP	DET	FP	DET	FP
Tile 1	0	0		0		0		0		0		0		0	
Tile 3	0	0		0		0		0		0		0		0	
Tile 4	6	6		6		6		6		5		6		4	
Tile 5	1	1		1		1		1		1		0		0	
Tile 6	0	0		0		0		0		0		0		0	

Figure 109. Lessening of crater depth during mesh development as point cloud resolution decreases.



5.1.6.2 Camouflet and spall detection with altitude

While crater detection was straightforward with the DFA, camouflet and spall detection was more difficult as the point cloud became coarser. Table 13 and Table 14 show that at the lowest altitude, all camouflets and spalls were detected, but as the altitude increased, more of these were missed and more surface damage was mistaken for a collection of camouflets rather than a crater or a spall or conversely a spall instead of

a crater. This is due to the same mesh smoothing that results in missed craters. As the number of points defining a damage feature decreases, eventually the damage feature will become invisible to the detection algorithm. These damage features can be obtained by changing the depth thresholds in the MATLAB GUI, but this defeats the purpose of a simple, one-setting-fits-all approach to detection.

Table 13. Summary of camouflet detection for all RiCopter flight lines.

March 2019	Number of Camoflets													
		50ft-5m/s		50ft-9m/s		100ft-5m/s		200ft-5m/s		260ft-5m/s		330ft-5m/s		400ft-5m/s	
	Actual	DET	FP	DET	FP	DET	FP	DET	FP	DET	FP	DET	FP	DET	FP
Tile 1	4	4		3		3		3		3		3		1	1
Tile 3	2	2		1		2		2		2		3	2	1	
Tile 4	0	0		0		0		1	1	3	3	2	2	2	2
Tile 5	0	0		0		0		0		0		1	1	0	
Tile 6	0	0		0		0		0		0		0		0	

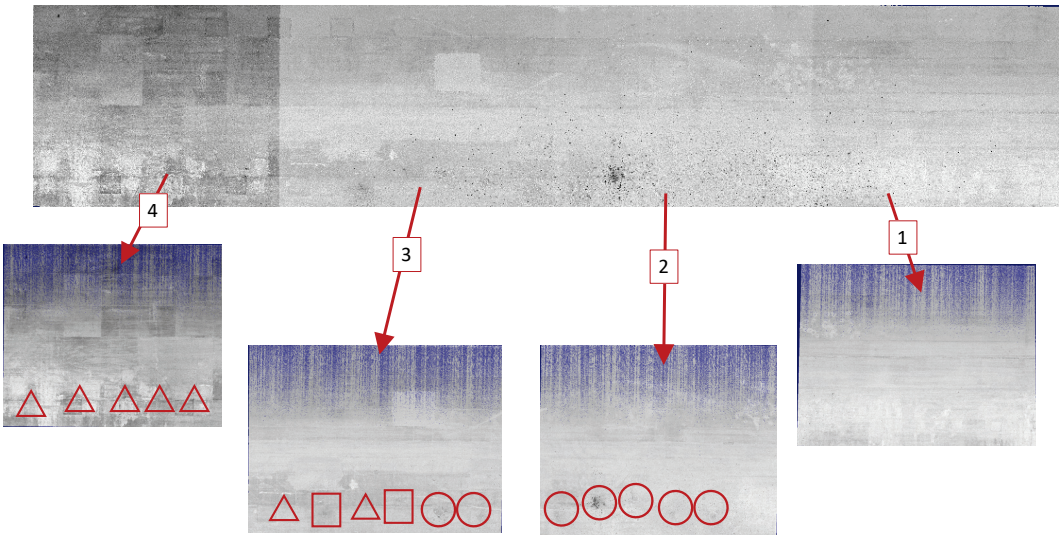
Table 14. Summary of spall field detection for all RiCopter flight lines.

March 2019	Number of Spall Fields													
		50ft-5m/s		50ft-9m/s		100ft-5m/s		200ft-5m/s		260ft-5m/s		330ft-5m/s		400ft-5m/s	
	Actual	DET	FP	DET	FP	DET	FP	DET	FP	DET	FP	DET	FP	DET	FP
Tile 1	0	0		0		0		0		0		0		0	
Tile 3	2	2		2		2		2		2		1		0	
Tile 4	0	0		1	1	2	2	1	1	1	1	2	2	1	1
Tile 5	0	0		0		0		0		0		0		0	
Tile 6	0	0		0		0		0		0		0		1	1

5.1.6.3 March 2019 – night detection

To define the effectiveness of LiDAR during nighttime operations, a flight was performed over the damaged runway at twilight, which provided very low light conditions as shown in Chapter 3. This data collect occurred over a smaller area of the runway than the daytime runs but was still able to capture all the damage features as shown in Figure 110. Tile 4 is the section of runway that was later covered with tires, but at night, no tires were deployed, allowing that area to be included and increasing the number of camouflet holes to detect. Tile 1 represented the undamaged tile to validate the DFA's ability to minimize false positives.

Figure 110. March 2019 nighttime reference LiDAR point cloud tiles for analysis.



The detection analysis used the 50-ft altitude 5 m/s flight combination so that results could be directly compared to Table 11. The results shown in Table 15 indicate that the lack of ambient lighting on the airfield had no impact on the ability to detect damage features. All damage features were detected with no false positives using the DFA. Since LiDAR is an active source sensor, it has no dependency on lighting conditions, and the runway point cloud appears exactly the same during the night as the day. This validates the appropriateness of LiDAR to be used as a 24-hr capable sensor package for physical description of the runway surface.

Table 15. Damage detection summary for March 2019 night flight (50ft-5m/s).

March 2019 50ft 5m/s Night	Number of Craters			Number of Camoflets			Number of Spall Fields		
	Actual	Detected	False Positives	Actual	Detected	False Positives	Actual	Detected	False Positives
Tile 1	0	0	No	0	0	No	0	0	No
Tile 2	5	5	No	0	0	No	0	0	No
Tile 3	2	2	No	2	2	No	2	2	No
Tile 4	0	0	No	5	5	No	0	0	No

5.1.7 Summary of DFA damage detection

When considering the overall success of the DFA system in its present configuration, the following data are considered for evaluation.

1. Only the crater detection from August 2016 as the camouflet/spall conflicts in the tiles were not clearly understood.

2. No data were taken from the December 2016 test event due to the poor quality LiDAR results from the low grazing angle, not representative of a real data collection scenario.
3. All data collected from the 50-ft telehandler altitude in the August 2017 event.
4. The 2-crater, 1-camouflet layout from the December 2017 obscurant test series was selected so comparison between scale of detection could be provided.
5. Only the 50 ft-5 m/s data collect from the March 2019 collect represents the best quality aerial data and the recommended flight altitude for runway data collection to maximize point cloud density. This was repeated for the night flight for direct comparison of LiDAR capability.

On the basis of these data sets, the DFA combined with the MATLAB mesh and detection settings were able to detect 26 of 26 craters with no false positives, 11 of 11 camouflet holes with no false positives, and 7 of 7 spall fields with one false positive. The ability to assign a single MATLAB setting and provide results that are uniform across various test events and various data collection methodologies illustrates the robustness of the analysis system. However, as can be seen, the mesh system begins to break down as the point cloud is reduced, which can be due to altitude, angle of incidence, or obscurant factors. Therefore, a refinement to the process is still required whereby the initial analysis system defines the local areas of interest where damage may exist. Once an area of interest is identified, a more detailed analysis of the raw point cloud in those regions can be performed to better approximate the damage feature rather than relying on mesh depths.

5.2 Z-deviation analysis for December 2016 test event

In Chapter 4 it was noted that there were two principal methodologies used in the development of damage analysis. While the DFA is the most current and successful, it was the z-deviation method first introduced in the fall of 2016 that convinced the RADAS program that rapid analysis of LiDAR was possible and had the potential to address damage detection. A cursory review of its use and output is shown to illustrate the genesis of the research program afforded to ERDC to continue with the DFA development.

The data collection team drove the mobile VZ-2000 LiDAR across the site in a single pass to collect the required data. These data were manually converted into a LAS file. A hard drive was then used to transfer the LAS

file to the analysis laptop housing CloudCompare and MATLAB. The LAS data were brought into CloudCompare and the data cropped to isolate only the region in which damage was present. Then, the MATLAB GUI was initiated by loading in the cropped LAS file. A clean area of runway surface was selected to identify the pre-surface. Following, two different MATALB GUI settings were implemented in Figure 111 and Figure 112 whereby the minimum (cMin) and maximum (cMax) extents of upheaval above the ground surface were modified. From Figure 105, there were six distinct craters formed in the concrete. As shown in the resultant plots of Figure 113, either 7 or 6 craters were detected in the mesh as a result of a small adjustment in the Threshold box. Both sets of spalls can be seen in the height map but cannot be detected with the z-deviation algorithm as no upheaval is present. Likewise, the settings can be modified to identify the upheaved camouflet present at the top of the crater locations if the Threshold criteria is altered, but it is recorded as a crater.

Figure 111. Z-deviation MATLAB settings for first trial.

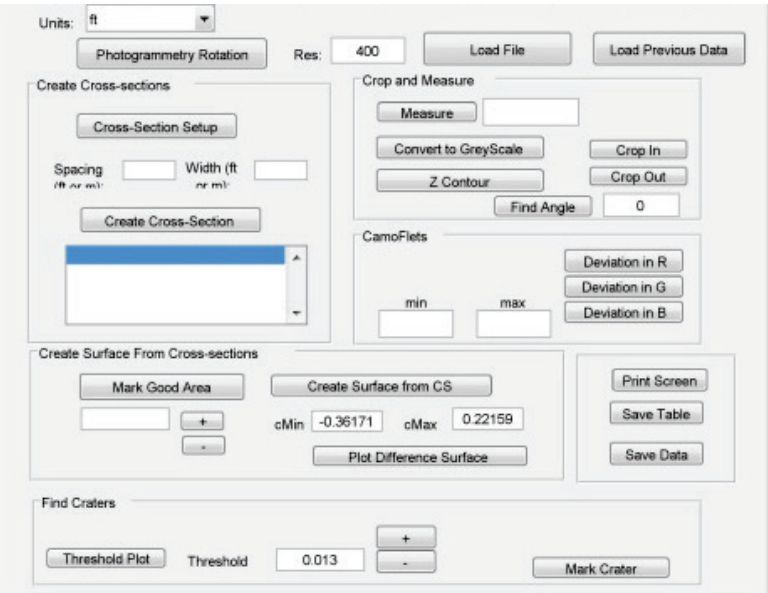


Figure 112. Z-deviation MATLAB settings for second trial.

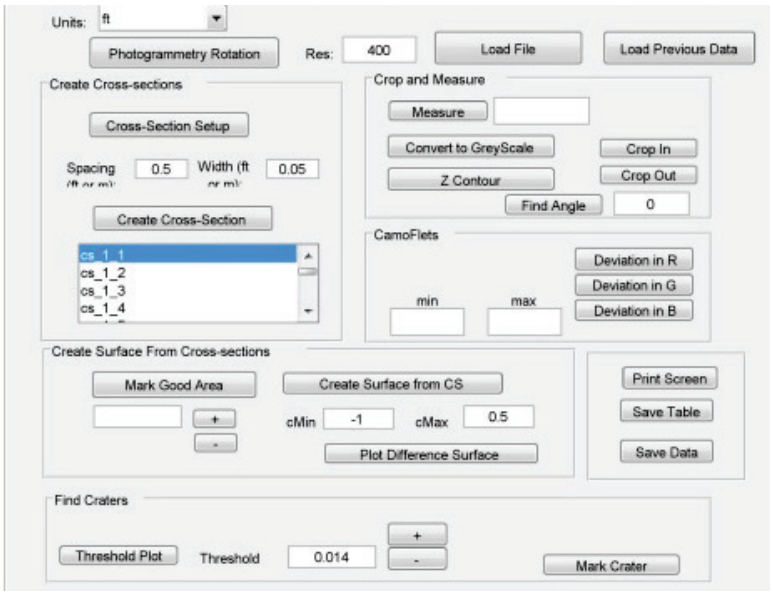
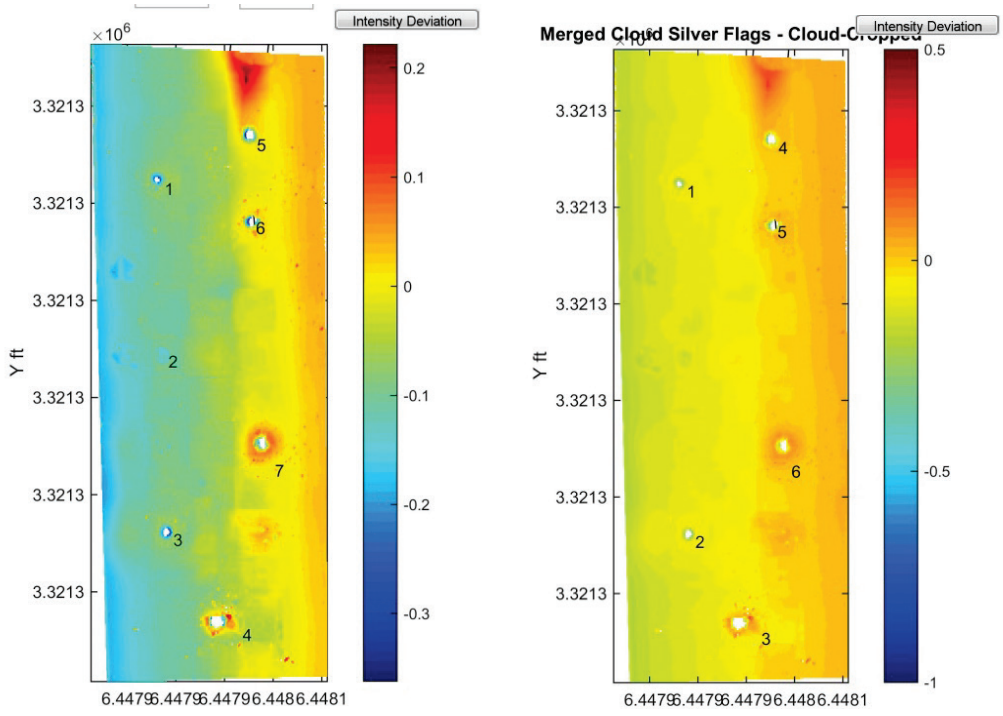


Figure 113. Results of crater detection from first trial (left) and second trial (right).



The z-deviation methodology was successful where the DFA was not on this data set because it relied only on looking at deviations in surface elevation above the pre-surface and did not concern itself with crater depth. While not the ideal software system to find all the requisite damage features, it was successful enough to demonstrate to the RADAS team that LiDAR can be collected and processed in a compressed timeframe and provide resultant

values of pavement damage with little user input. The results of this analysis identified the weaknesses of its generation yet inspired continued research from the Air Force to develop the current DFA methodology.

6 Data Delivery

The RADAS program chose as its data delivery system to the Airmen, the GeoExPT as described in Section 2.2.3. GeoExPT in its current configuration is an add-on to the AutoCAD 3D software environment, which is installed first in order to operate the GeoExPT software. GeoExPT ingests either an AutoCAD shapefile or a data element that can be read by GeoExPT containing the classification and location of damage features.

6.1 AutoCAD shapefiles for GeoExPT

To satisfy the RADAS task of integrating with GeoExPT, the output from the MATLAB analysis was designed to be a small (kB) size file that contains the classification (crater, spall, camouflet), the UTM positioning of the centroid of the damage object, and a radius of the extent of upheaval to distinguish damage size. A folder is set up on the processing laptop that contains a series of small shapefiles for each damage feature detected during the LiDAR collection. Data objects within this folder are then uploaded into GeoExPT to be graphically displayed to the user. The location and type of each damage feature can then be analyzed in GeoExPT to provide the MAOS.

GeoExPT was designed as a visually intensive tool and functions best for the user if a post-blast mosaic of the airfield is displayed so projection of the damage features can overlay a visual reference on the airfield for the airmen to assess accuracy. This mosaic can be generated from either compressed video or still images taken of the runway. In the current RADAS configuration, a video stream is displayed in real time to provide the airmen with an immediate first look at the extent of damage. Later in the data collection, more detailed imagery from still photos are used to augment blurry video and to provide a more accurate GPS positioning of the damage features. Objects displayed using the GPS supplied with the real-time video feed can be off by several meters, while still images are within about 1 to 2 m of the actual position. The ERDC shapefiles are generally within about 1 m or less from the actual position and within a few inches if an RTK-GPS system is employed. Therefore, a mosaic generated from still images with better GPS positioning can make the most sense to the airmen performing the quality assessment of the data analysis.

The GeoExPT integration with the ERDC shapefiles was validated during the August 2017 data collection. A mosaic of the runway taken from the

RADAS team UAS imagery system was provided as a background image for ERDC to use. The folder of shapefiles that were generated from the damage analysis discussed in Chapter 5 was then uploaded into GeoExPT. Figure 114 shows how damage features were identified by size and color of circle to provide a visual discernment between damage types.

Figure 114. GeoExPT from August 2017 showing damage features.

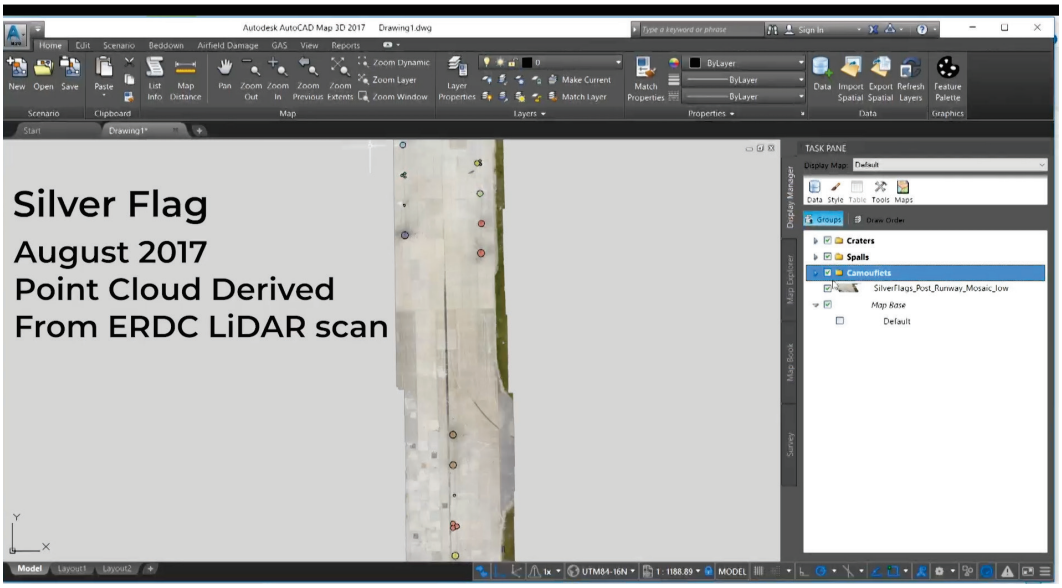
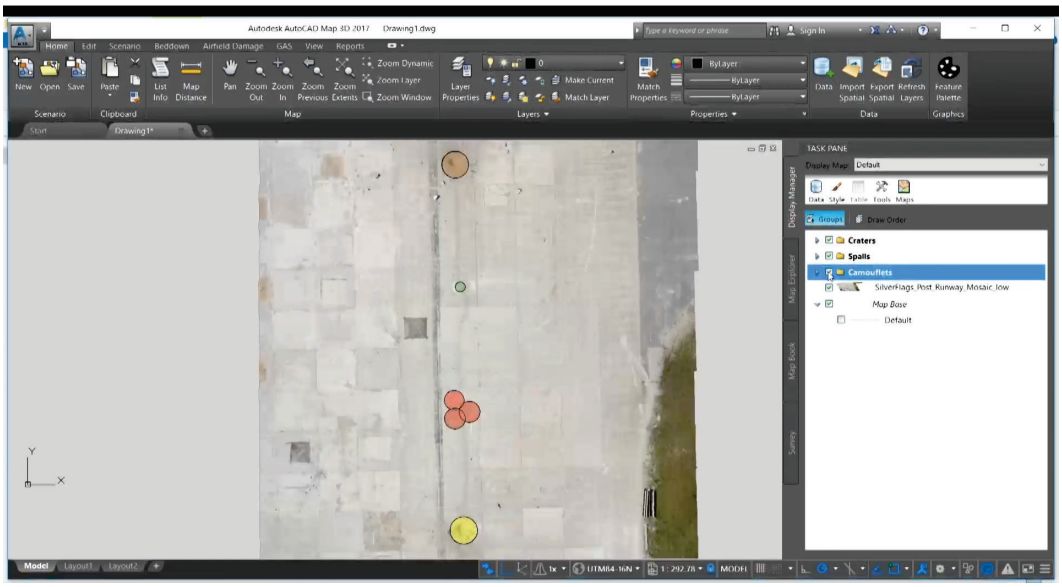


Figure 115 demonstrates how a close inspection of the damage features can be performed. ERDC chose to provide a transparency to the colorization such that the mosaic can still be viewed underneath the damage icon. In the figure it can be seen that the detected damage presented from the shapefiles overlays the actual location of the damage very accurately. Even the camouflet is within 12 in. or less from the actual location, which is sufficient for MAOS calculation. For repair, a construction team can now travel directly to the location of the damage and visually identify the exact location of the feature quickly.

Figure 115. Close up of GeoExPT from August 2017 showing damage features.



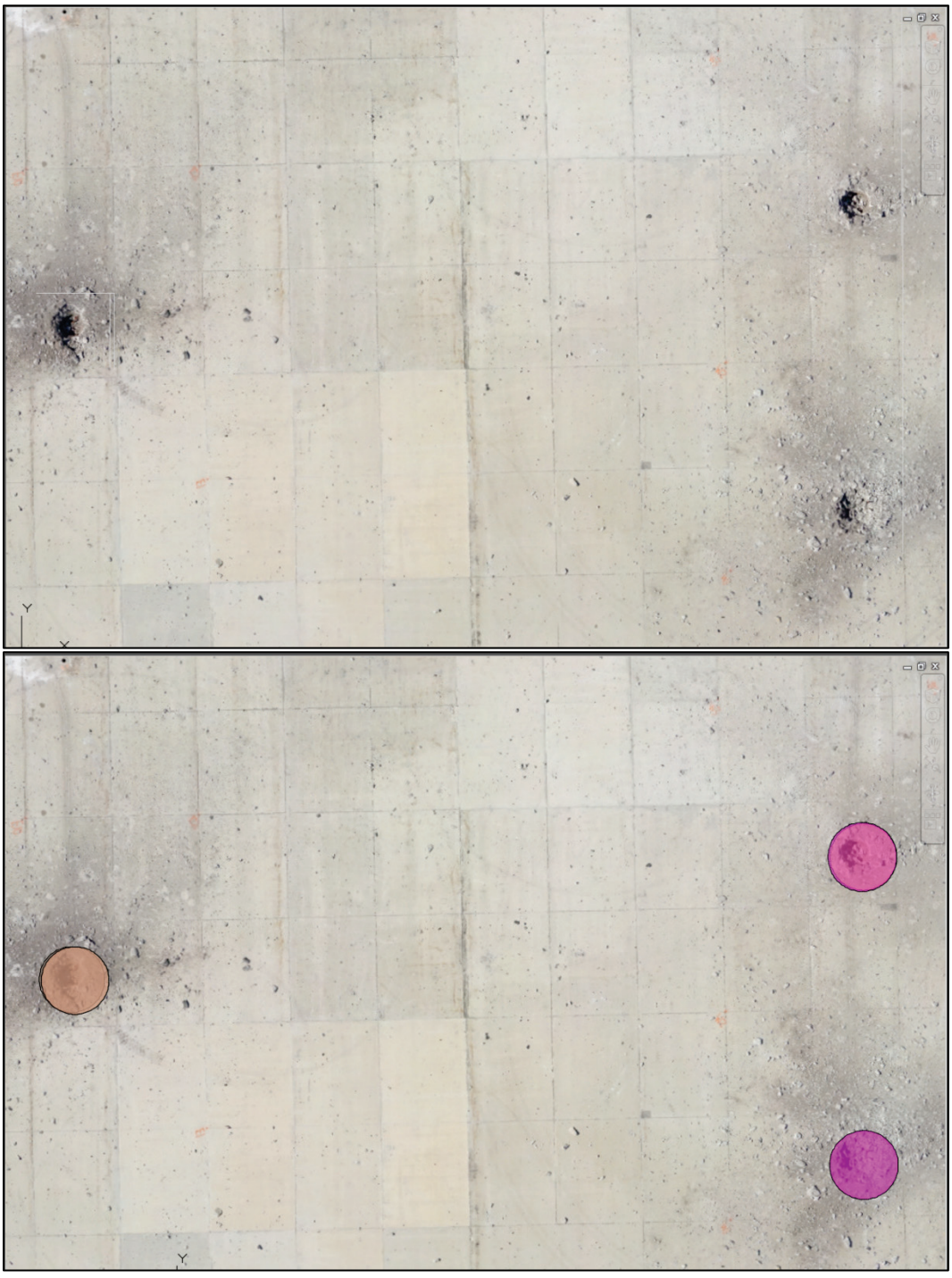
6.2 Identification of damage features in GeoExPT

Using the data from the August 2017 data collection, the following discussion illustrates the accuracy of the LiDAR point cloud MATLAB analysis to identify the location and size of each damage feature.

6.2.1 Crater identification

Figure 116 shows the presence of three craters on a segment of airfield obtained from a mosaic of images created from the Matrix-E UAS. For scale, this image is 41.2 m across. The segment of LiDAR from which data analysis occurred is Tile 2 from Figure 106. Shapefiles were created for each damage feature detected and displayed in GeoExPT. Figure 116 shows a snapshot of that section of GeoExPT where the identified location and extent of the crater lip based on the mesh generated for those crater locations. Notice the close association of the crater and circle dimensions indicating little error from the positioning and dimensioning of the DFA analysis.

Figure 116. Image mosaic of three craters locations from August 2017 event (Tile 2).



6.2.2 Spall identification

Figure 117 shows the presence of two spall fields for a total of six individual spalls on a segment of airfield obtained from a mosaic of images created from the Matrix-E UAS. The segment of LiDAR from which data analysis occurred

is Tile 1 from Figure 106. For scale, this image is 41.2 m across. The lower half of the figure shows the identified location of each spall based on the point cloud generated for those spall locations. Each spall field is actually represented by three individual spalls making up each field providing visual clarification of a damage feature not associated with a crater. Notice the close association of the spall and circle dimensions indicating little error from the positioning and dimensioning of the DFA analysis.

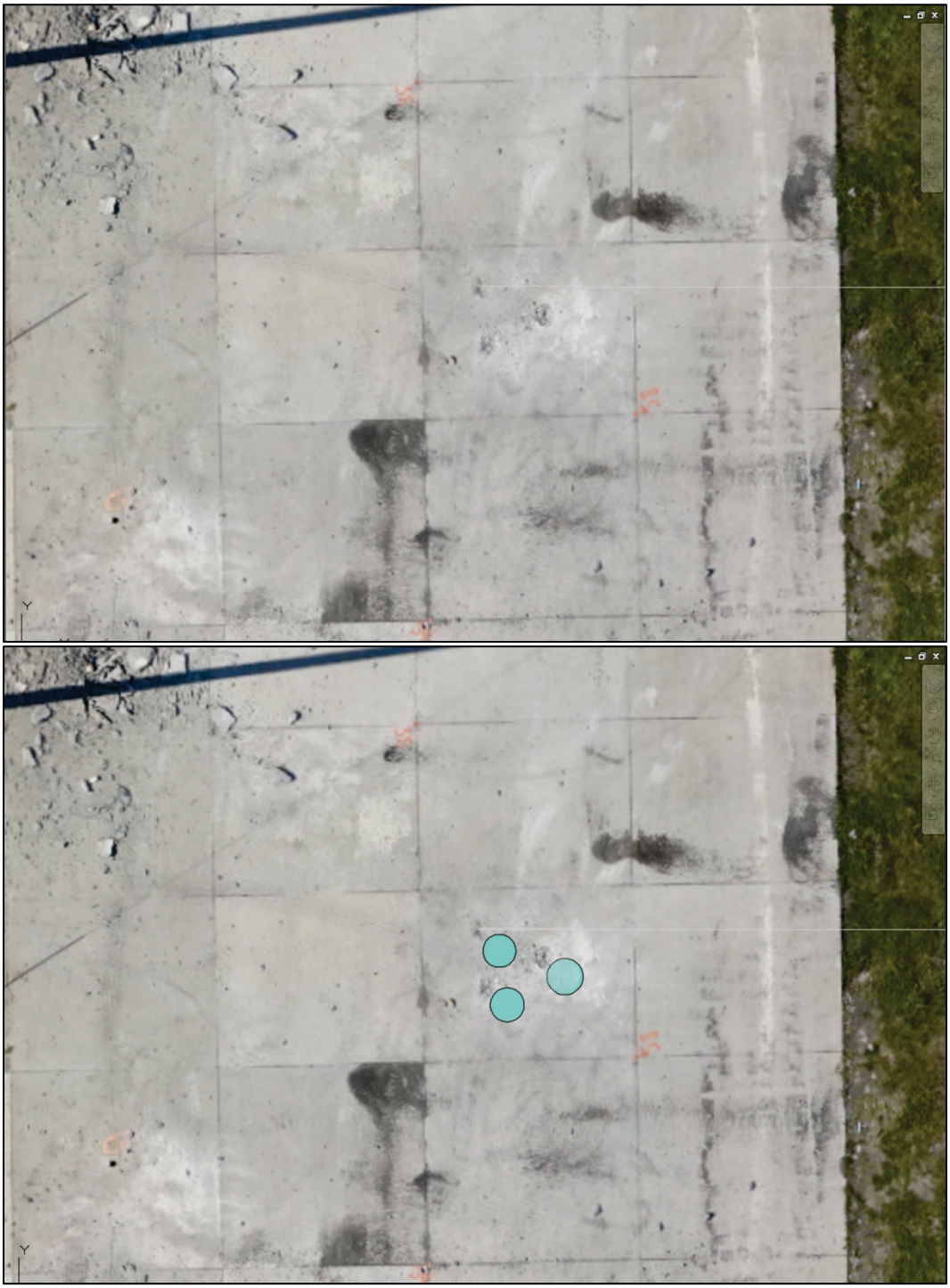
Figure 117. Image mosaic of two spall locations from August 2017 event (Tile 1).



The next set of spall illustrations in Figure 118 is taken from Figure 106, Tile 6. For scale, this image is 21.5 m across. This LiDAR data set shows the ability of the DFA to detect spalls when there is considerable noise and other debris on the runway. Notice again, in the lower half of the image, the close association of the location of each individual blast in the three-spall field. In

this instance, a discrepancy in the GPS positioning can be observed by about 1 to 2 ft but is within an acceptable range for the RADAS effort.

Figure 118. Image mosaic of two spall locations from August 2017 event (Tile 6).



6.2.3 Camouflet identification

Figure 119 shows the presence of a single camouflet on a segment of airfield obtained from a mosaic of images created from the Matrix-E UAS. The segment of LiDAR from which data analysis occurred is Tile 5 from Figure 106. For scale, this image is 21 m across. The lower half of the figure shows the identified location of the camouflet based on the mesh generated for that damage feature. Notice the close association of the position to the actual hole being within about 1 to 2 ft from the actual location indicating little error from the positioning of the DFA analysis.

Figure 119 Image mosaic of one camouflet location from August 2017 event (Tile 5).



7 Conclusions and Recommendations

7.1 Conclusions

Beginning in 2016, the U.S. Air Force initiated work with ERDC under their Rapid Airfield Damage Assessment System (RADAS) program to develop an expedient technology to identify, classify, and map damage features on a runway surface following a bombing attack. The ERDC chose a Light Detection and Ranging (LiDAR) system as its primary sensor to accomplish this task. LiDAR provides an immediate data product by representing the ground surface as a three-dimensional point cloud and emits an intensity return that can be used as a filter for the detection of unexploded ordnance. LiDAR collection can be conducted in both day and night operations, being independent of surface lighting conditions, and can penetrate airborne obscurants such as smoke. For the detection of damage features, the optimum deployment of LiDAR is from a near normal perspective to the runway surface from an elevated position. ERDC began their research using a mobile LiDAR sensor suspended 50 ft in the air on a telehandler before deploying a LiDAR system mounted to an unmanned aerial system (UAS) flying at altitudes between 50 and 100 ft above ground level.

ERDC developed a computational methodology whereby LiDAR sensor data were converted into georeferenced point clouds (LAS files) incorporating the inertial measurement unit of the UAS and the geospatial positioning of the sensor. The LiDAR collection software segments each scan into a series of smaller LAS files at a user-specified time period for easier data analysis. ERDC then developed a series of algorithms using MATLAB that converts the LAS files into meshes on which a depression-filling algorithm is used to identify locations on the runway where penetration beneath the pavement surface has occurred due to cratering. The ERDC-developed MATLAB algorithms then identify in a stepwise fashion the location, radius, and centroidal position of each crater, camouflet, and spall/spall field occurring on the runway surface. The detected damage is saved as a series of small shapefiles or data objects that can be displayed within the Geospatial Expeditionary Planning Tool (GeoExPT) used by the Air Force to display runway damage. GeoExPT is then used to calculate the Minimum Airfield Operating Surface from the location of all damage features. This entire computational process has

been automated such that the user's only responsibility is to activate the software, deploy the LiDAR sensor, and view the resultant damage data.

LiDAR data were collected during a series of field-collection events from August 2016 until March 2019 at Silver Flag Airfield, Tyndall Air Force Base. These data collections simulated a damaged runway using explosives to generate craters, spalls, and camouflets in the pavement surface. LiDAR point clouds were initially generated from a Riegl VZ-2000 mobile LiDAR sensor mounted on an elevated lift to simulate aerial data collection and later from a Riegl VUX-1LR LiDAR sensor mounted on a Riegl RiCopter UAS at multiple altitudes and ground speeds. The optimum deployment of the LiDAR was found to be at 50 to 100 ft above ground level. Refinements to the detection algorithms and the technology to automate and transmit the data analysis were developed during these data collections.

On the basis of data sets with representative aerial LiDAR, the DFA combined with fixed detection settings was able to correctly identify and classify 26 of 26 craters with no false positives, 11 of 11 camouflet holes with no false positives, and 7 of 7 spall fields with one false positive. Detections of these damage features were made during daytime and nighttime operations, and through obscurants validating the use of LiDAR sensor technology in all times of day and during low visibility dust emissions during the initial stages of an airfield attack. The ability to assign a single MATLAB setting and provide results that are uniform across various test events and various data collection methodologies illustrates the robustness of the analysis system. Regarding processing time, the ERDC methodology was able to identify damage features with a time lag of only 2 min behind the moment of LAS file creation in flight. This indicates that with a series of between 2 and 4 UAS-based LiDAR systems, the entire airfield could be scanned and processed within a 30-min time window as dictated in the RADAS objectives.

7.2 Recommendations

7.2.1 LiDAR as a sensor platform

Given the success of LiDAR in accurately modeling the runway in a compressed timeframe while enabling engineering analysis, it is recommended that continued investment in LiDAR sensor technology occurs as it represents the future of surface scanning. As LiDAR technology becomes more useful to commercial parties, the technology

will become more cost-effective for DoD purposes. LiDAR also continues to be adapted to myriad military problems creating a more robust infrastructure to leverage within the DoD.

7.2.2 Integration of hardware

The ability to automate the detection process was successfully demonstrated using the NUC processing computer mounted to a telehandler. However, to provide a field-ready solution to the Air Force to detect surface damage features, a comprehensive program is required. Such a program would enable integration of the processing computer and network hardware into the UAS to fully air deploy the solution. This will require coordination with the Air Force in defining a suitable UAS platform and the FAA and USACE Aviation to obtain Air Worthiness Certification. Further, software logistics must be developed with the agreed-upon merged platform to enable ease of operation to the airmen, training requirements, and storage and operation criteria.

7.2.3 Algorithm advances

The developed DFA is robust and accurate; however, improvements can still be made to make it more time efficient and user friendly. The coarseness of speedily constructed meshes makes accurate depth and upheaval extents difficult to measure accurately. Refinements to the analysis process include using the DFA to identify damage features from which a secondary, detailed analysis of the point cloud can occur. This detailed analysis can identify more accurate depths for damage distinction or upheaval extents for crater repair criteria.

Integration with GeoExPT is a key component in delivering an effective solution to the Air Force. The ERDC needs to continue integration with the GeoExPT environment and other RADAS team products to provide a more comprehensive view of the airfield with layers of redundancy to ensure confidence to the user. The data product provided by ERDC algorithms focuses mainly on the detection of pavement damage and requires a visualization component provided by other means of varying accuracy. It is suggested that the LiDAR intensity image serve as a replacement for background mosaics to shorten decision timeframes.

7.2.4 Continued data collection

The best way to continue to refine the ERDC RADAS detection process is to demonstrate RADAS solutions at actual blast test events and improve integration with other RADAS partners. While this report demonstrates the ERDC's ability to develop and validate damage detection methodologies, blast events always provide unique issues that may present in real world runway detonations.

Further, much of this research has focused on the performance of damage detection on concrete surfaces. Many airfields around the world are asphalt surfaced and represent contingency landing strips during conflict. It is suggested that significant testing occur on asphalt surfaces in an effort to contrast data collection on concrete surfaces. The inherent difference in flexible and rigid pavement structures may lead to varied crater geometry.

References

Barnes, R., C. Lehman, and D. Mulla. 2014. Priority-flood: an optimal depression-filling and watershed-labeling algorithm for digital elevation models. *Computers and Geosciences* 62:117-127.

Delaunay, Boris. 1934. Bulletin de l'Académie des Sciences de l'URSS. *Classe des Sciences Mathématiques et Naturelles* 6:793–800.

Filler, Lance. 2011. *Rapid Airfield Damage Assessment System (RADAS) final report*. Panama City, FL: HQ Air Force Civil Engineering Support Agency, Tyndall AFB.

Huang, C., and J. M. Bradford. 1990. Portable laser scanner for measuring soil surface roughness. *Soil Science Society of America Journal* 54:1402-1406.

Planchon, O., and F. Darboux. 2001. A fast, simple and versatile algorithm to fill the depressions of digital elevation models. *Catena* 46(2-3):159-176.

USAF TTP 3-32.11. 2016. *Airfield damage assessment after attack*. Air Force tactics, techniques and procedures 3-32.11. Washington, DC: U.S. Department of the Air Force.

Vincent, L., and P. Soille. 1991. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13(6):583-598.

Voronoi, Georgy. 1908. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. *Journal für die Reine und Angewandte Mathematik* 1908(133):97–178.

Yin, J., Y. Xu, H. Li, and Y. Liu. 2013. *A novel method of crater detection on digital elevation models*. International Geoscience and Remote Sensing Symposium. Melbourne, Australia.

Appendix A: MATLAB Damage Detection

The following section documents the screen captures obtained from the MATLAB DFA results for each LiDAR tile that indicated a detection from the tiles detailed in Chapter 5. If no detection of a damage feature was made, then no screen display is returned to the user, which is why some tiles shown have screen shots for all three damage types (crater, camouflet, and spall field) while others have between zero and two screen shots when only partial data are available. No screen shots are presented for the blank tile included in each test series as the DFA did not detect any false positive damage features in these tiles. The data summarized in the tables presented in Chapter 5 can be cross referenced with the detections indicated in each screen capture presented.

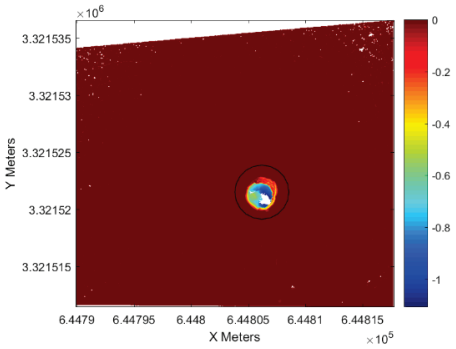
Note that the original screen captures denote a damage feature with a small black circle. In the DFA, craters are detected first and, if detected, a radius around the centroid is determined; the points making up the crater are removed from the tile leaving a white circle or a void space in the LiDAR point cloud as seen on subsequent screen captures. Camouflets are detected next and, in the illustrations below, are highlighted with a yellow circle to make visualization easier in the report. These are then removed leaving white circles where the camouflets had been; spalls are detected last and are also highlighted with yellow circles in the report to make visualization easier.

Aug 2016: MATLAB screen captures for Artificial Runway

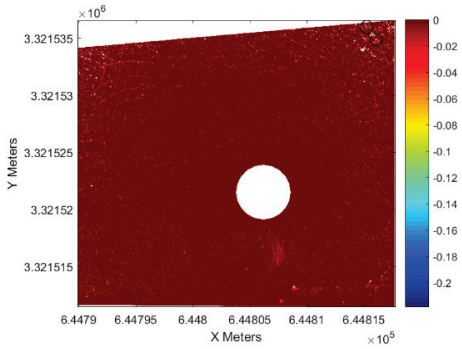
Due to the large number of camouflet and spall field detection in these very damaged runway sections, many of which are incorrect, no additional yellow highlighting is provided so as not to complicate the scene.

Tile 1

Crater
1/1 found

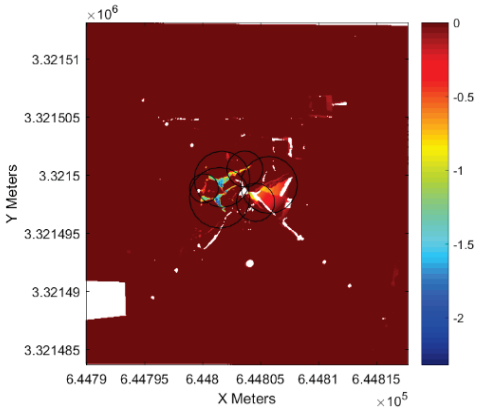


Camoflet
2 found, 2 FP

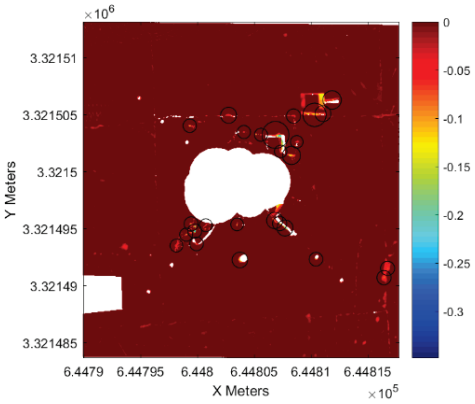


Tile 2

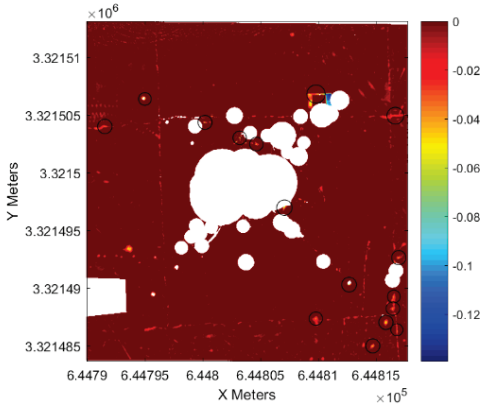
Crater
1/1 found



Camoflet
Multiple found

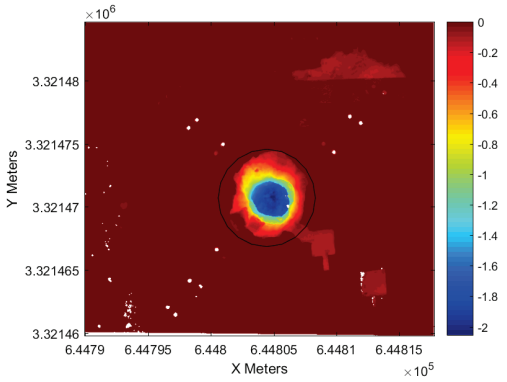


Spall Field
Multiple found

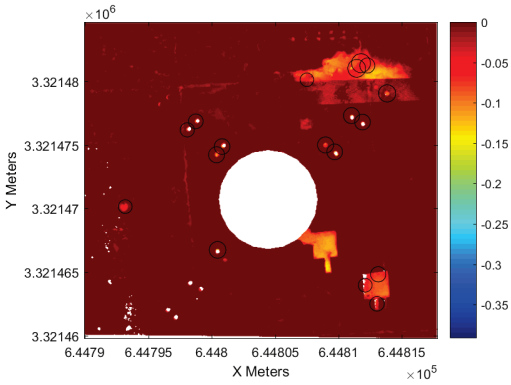


Tile 3

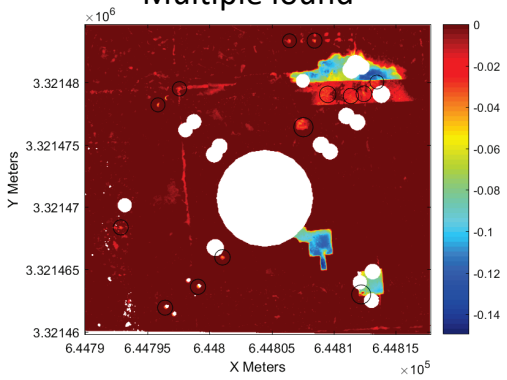
Crater
1/1 found



Camoflet
Multiple found

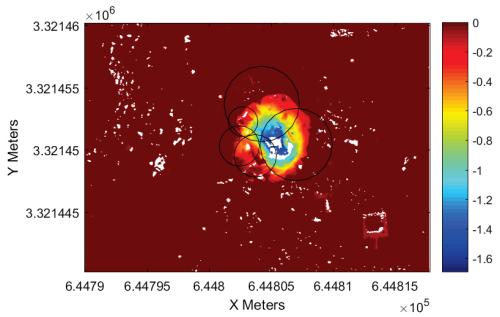


Spall Field
Multiple found

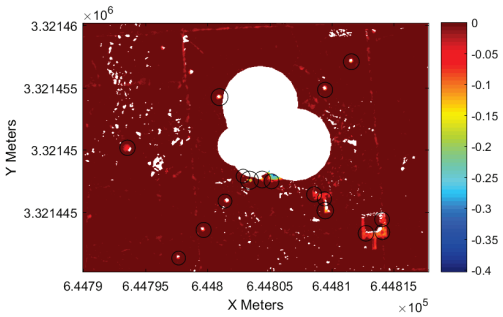


Tile 4

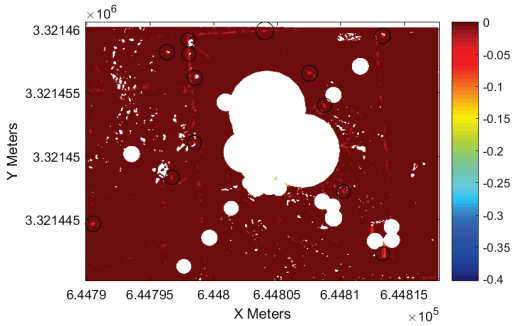
Crater
1/1 found



Camoflet
Multiple found

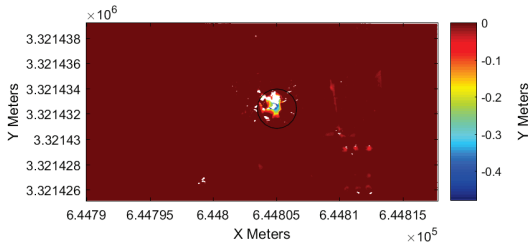


Spall Field
Multiple found

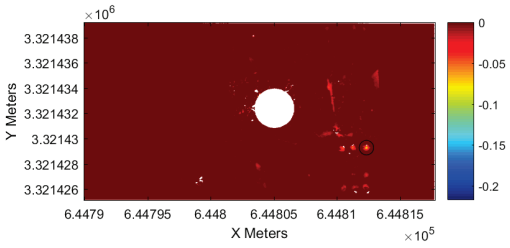


Tile 5

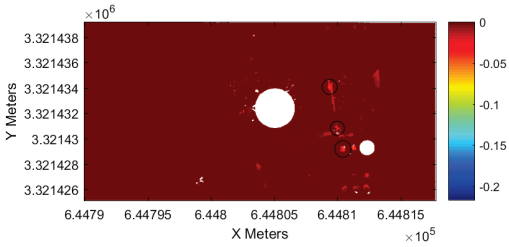
Crater
1/1 found



Camoflet
1 found, 1 FP



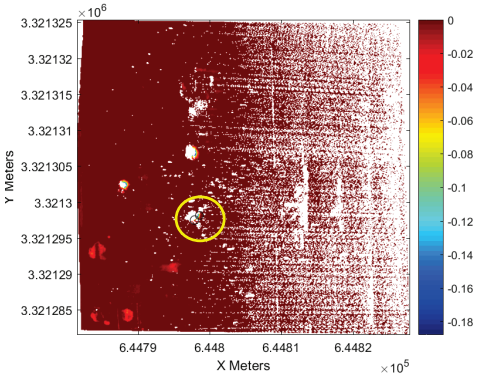
Spall Field
3 found, 3 FP



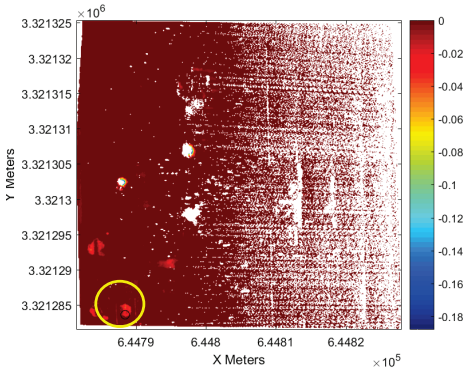
Dec 2016: MATLAB screen captures for December 2016 event

Tile 2

Camoflets
1/1 Found, 1FP

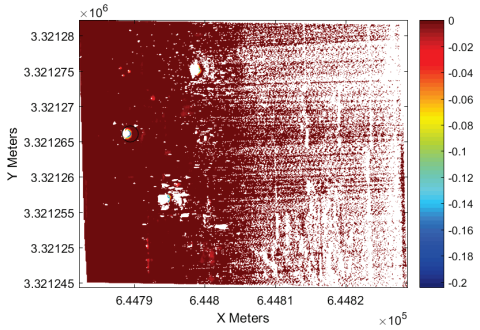


Spall Fields
1 FP

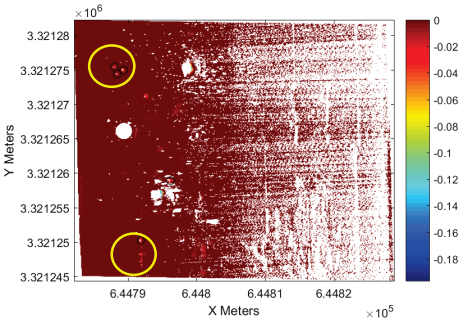


Tile 3

Craters
1/3 Found



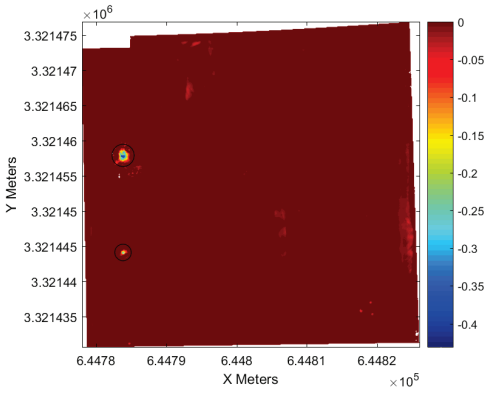
Spall Fields
1/2 Found, 1 FP



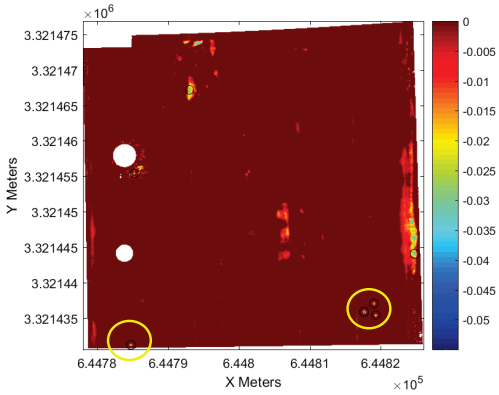
Aug 2017: MATLAB screen captures for August 2017 data collect

Tile 1

Craters
2/2 Found

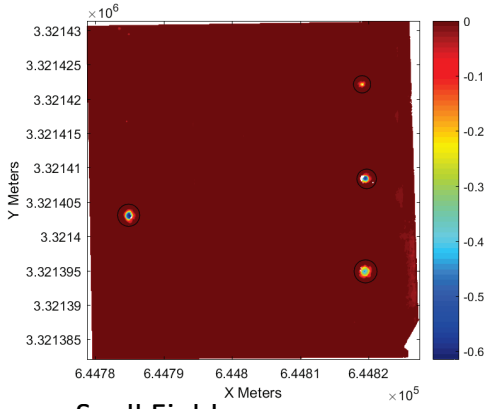


Spall Fields
2/2 Found, 1 FP

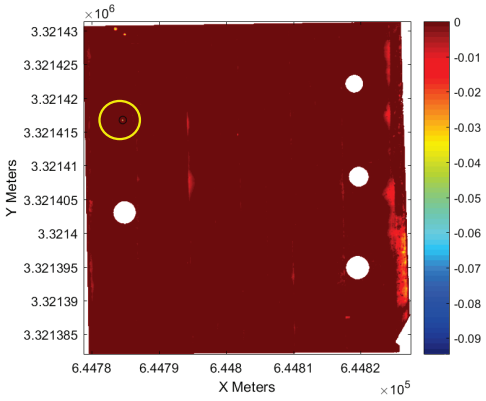


Tile 2

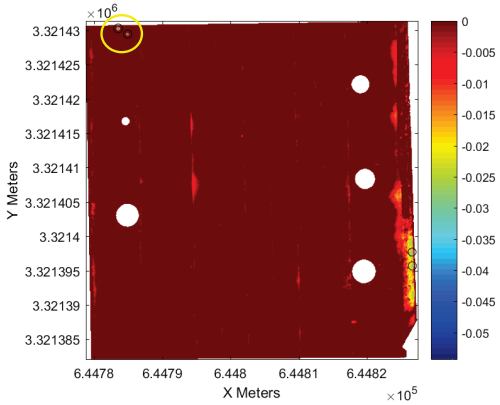
Craters
4/4 Found



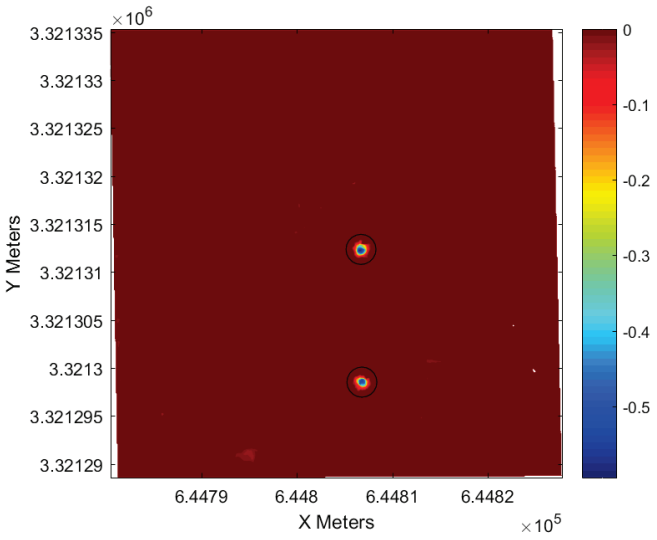
Camoflets
1/1 Found



Spall Fields
1/1 Found

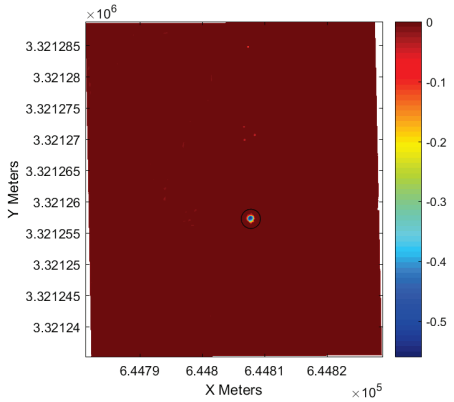


Tile 4
Craters
2/2 Found

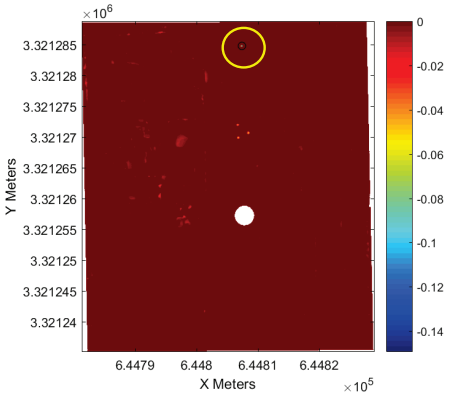


Tile 5

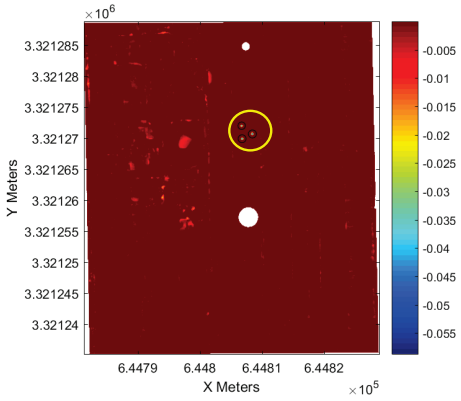
Craters
1/1 Found



Camoflets
1/1 Found

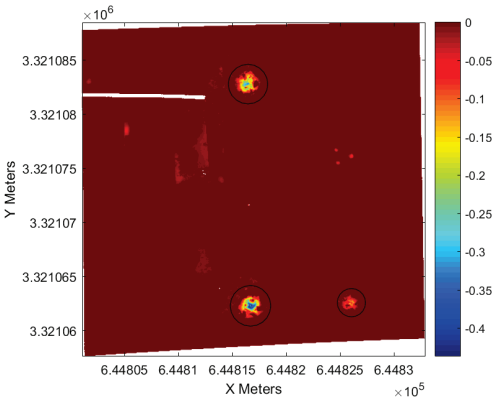


Spall Fields
1/1 Found

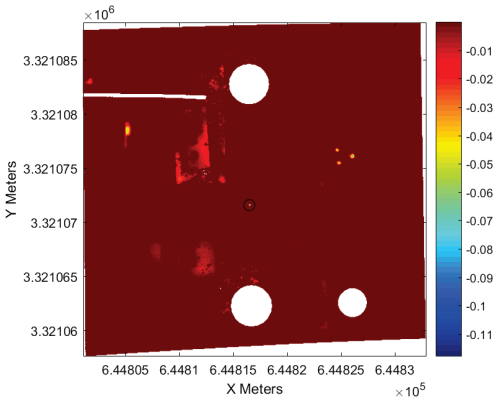


Tile 6

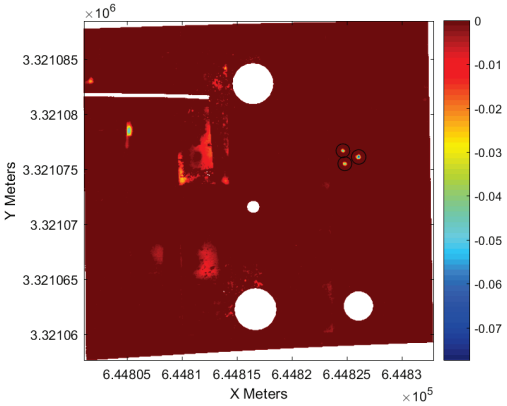
Craters
3/3 Found



Camoflets
1/1 Found



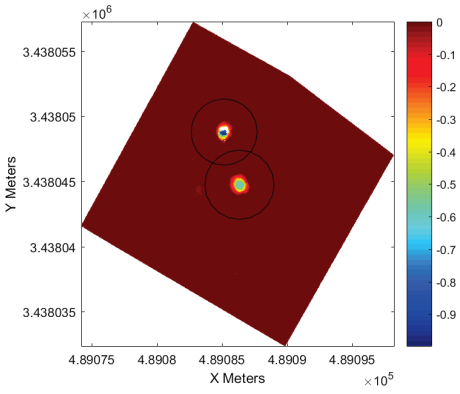
Spall Fields
1/1 Found



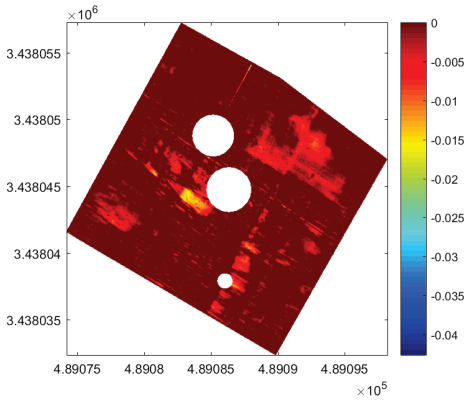
Dec 2017: MATLAB screen captures for Obscurant data collect

Tile 1

Craters
2/2 Found



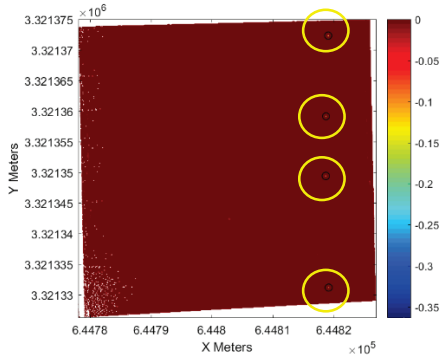
Camoflets
1/1 Found



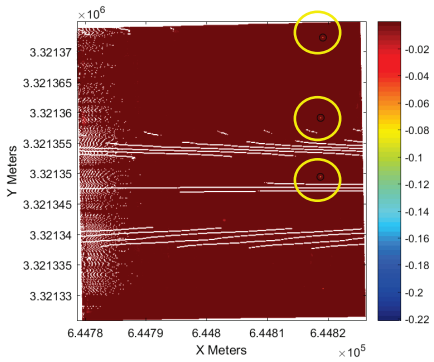
Mar 2019: MATLAB screen captures for March 2019 data collect

Tile 1

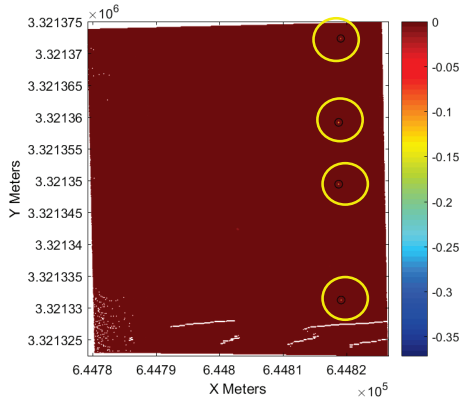
Camoflets
50’ 5m/s
4/4 Found



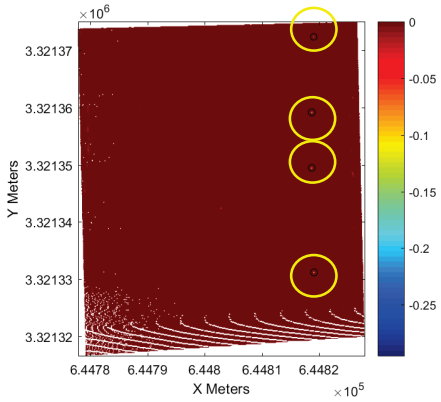
50’ 9m/s
3/4 Found



100’
3/4 Found



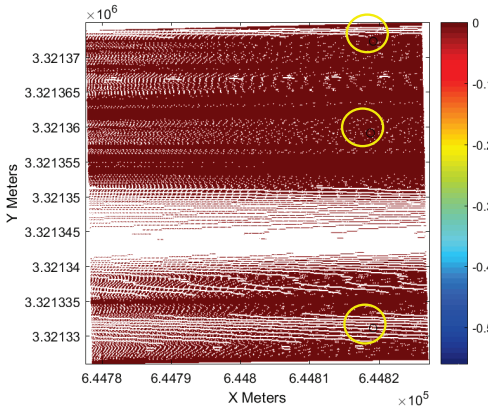
200’
3/4 Found



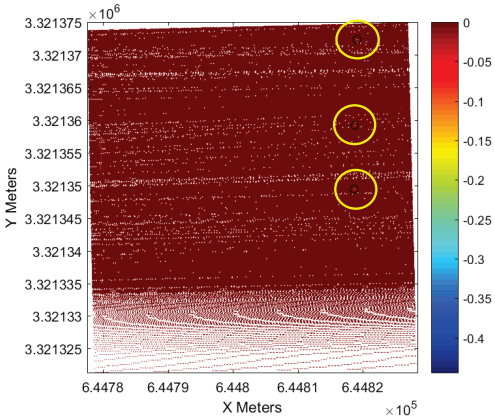
Tile 1

Camoflets

260’
3/4 Found



330’
3/4 Found



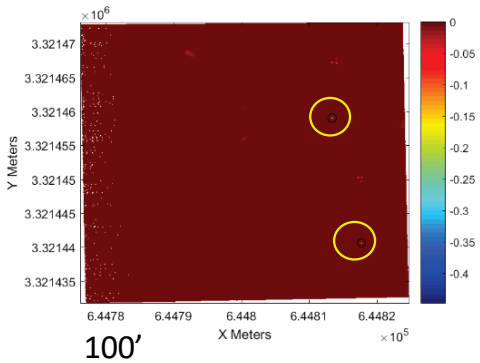
400’
0/4 Found, 1 FP

NA

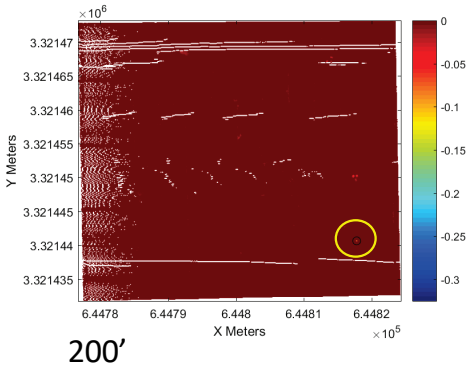
Tile 3

Camoflets

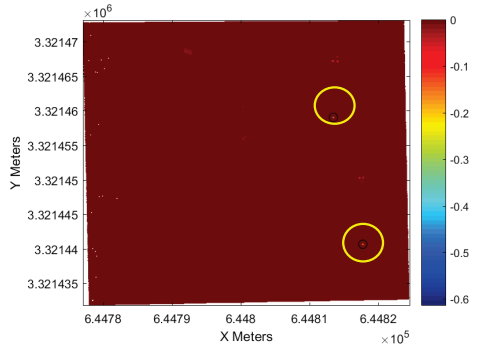
50’ 5 m/s
2/2 Found



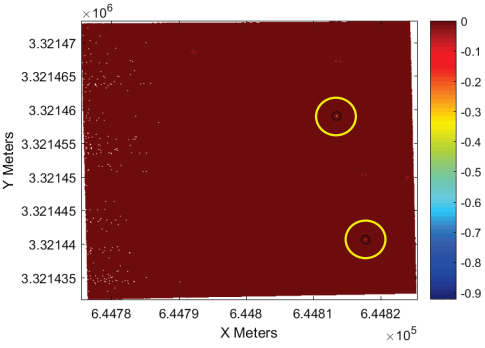
50’ 9 m/s
1/2 Found



100’
2/2 Found



200’
2/2 Found

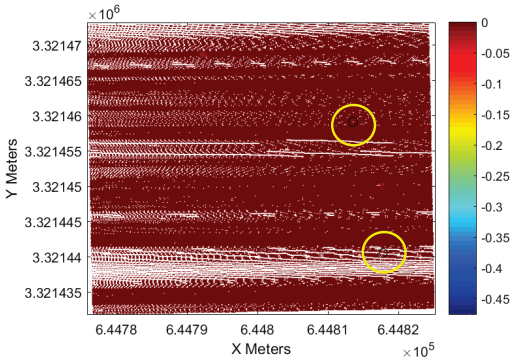


Tile 3

Camoflets

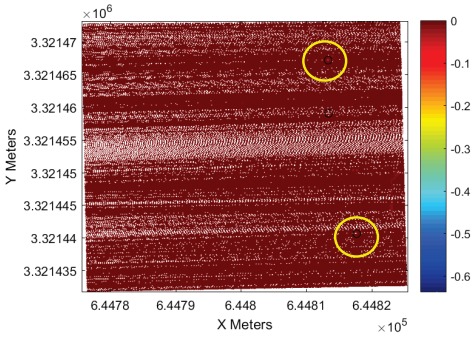
260’

2/2 Found



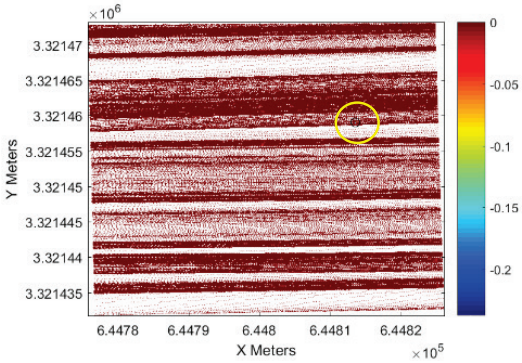
330’

1/2 Found, 2 FP



400’

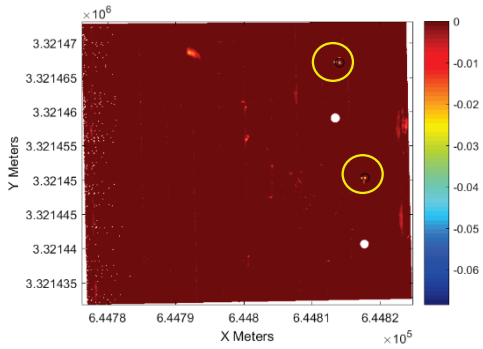
1/2 Found



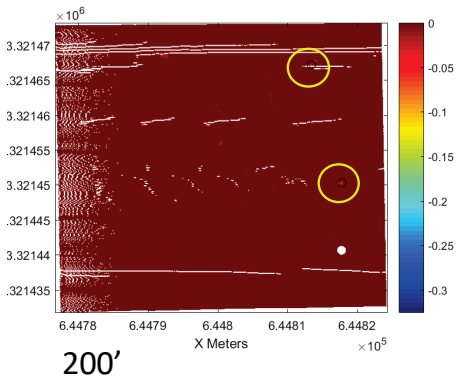
Tile 3

Spall Fields

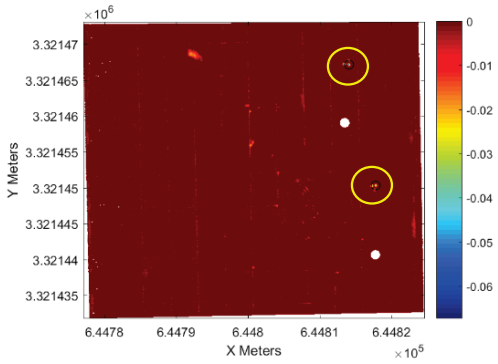
50’ 5 m/s
2/2 Found



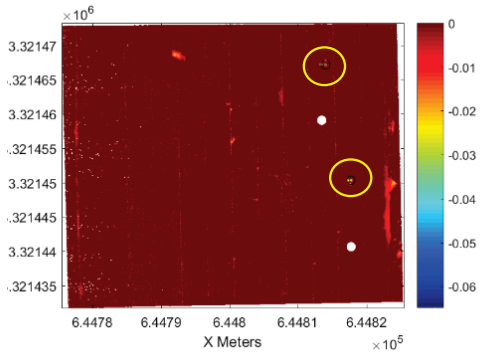
50’ 9 m/s
2/2 Found



100’
2/2 Found



200’
2/2 Found



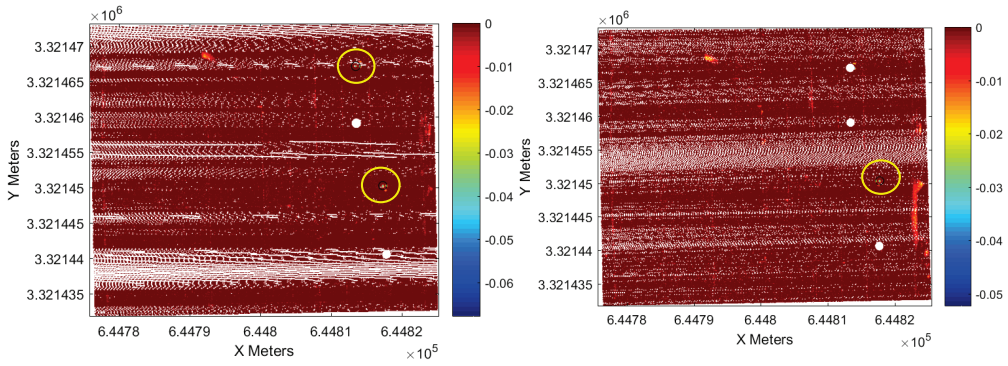
Tile 3

Spall Fields

260’

330’

1/2 Found



400’

0/2 Found

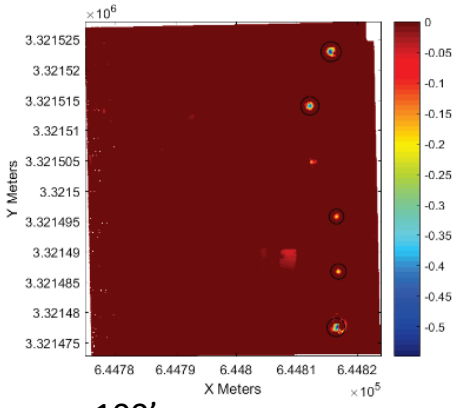
NA

Tile 4

Craters

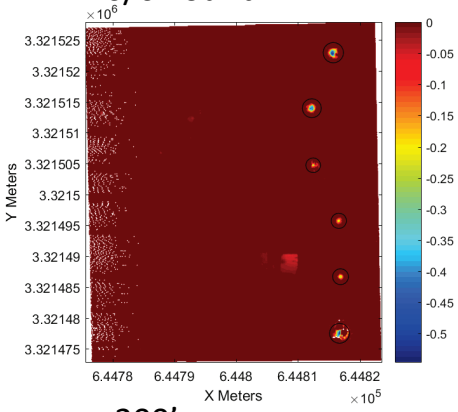
50’ 5 m/s

6/6 Found



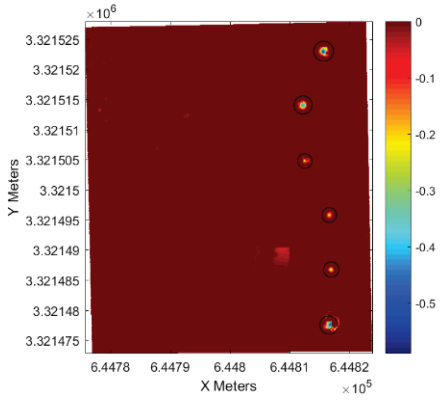
50’ 9 m/s

6/6 Found



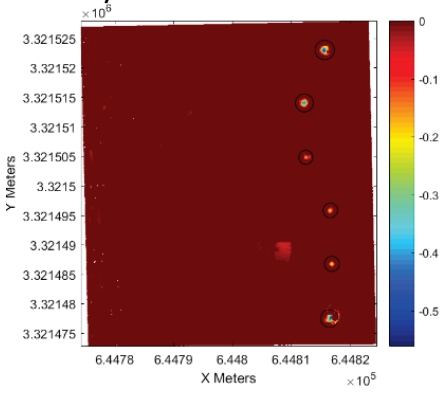
100’

6/6 Found



200’

6/6 Found

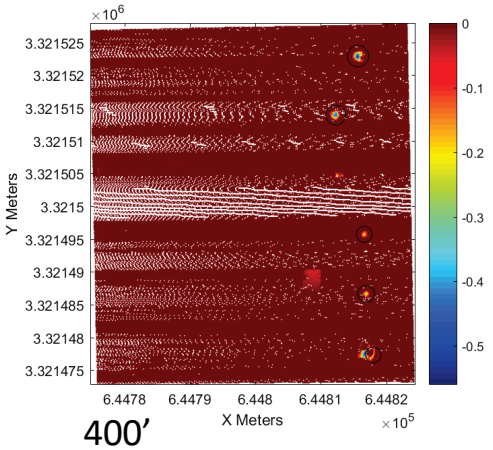


Tile 4

Craters

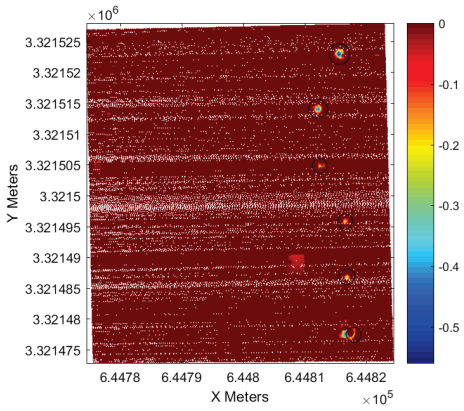
260’

5/6 Found



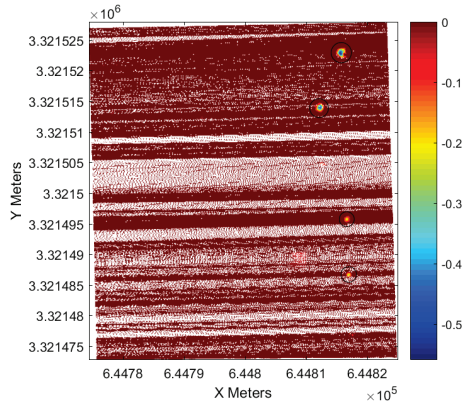
330’

6/6 Found



400’

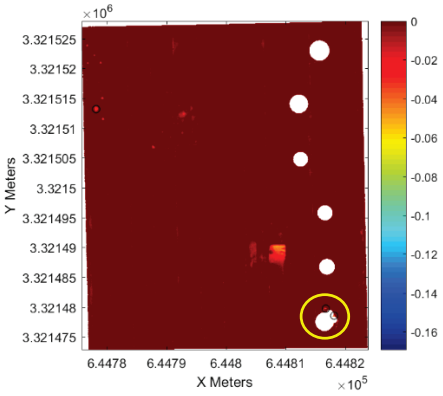
4/6 Found



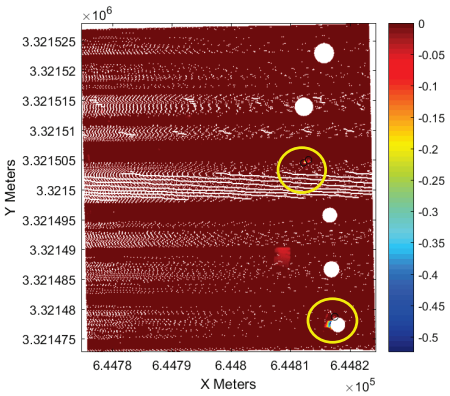
Tile 4

Camoflets

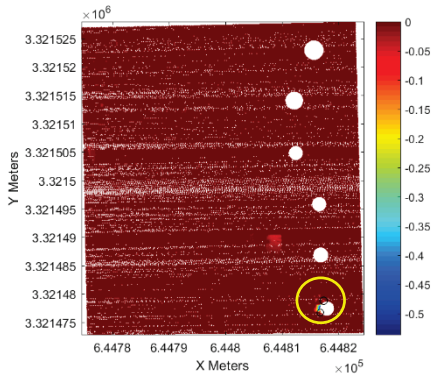
200’
1 FP



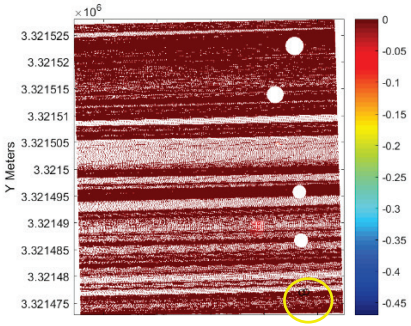
260’
3 FP



330’
2 FP

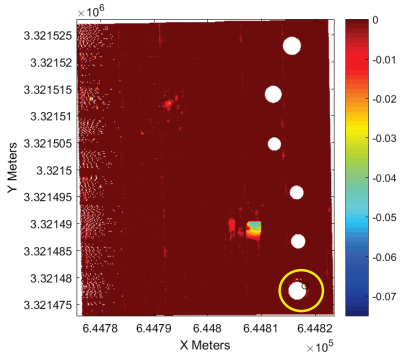


400’
2 FP

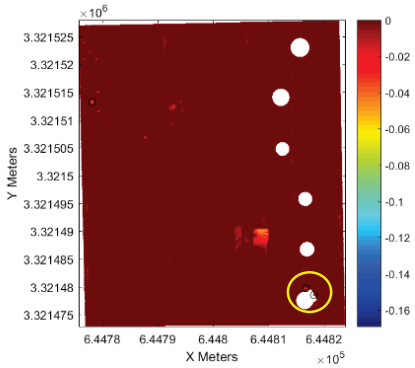


Tile 4

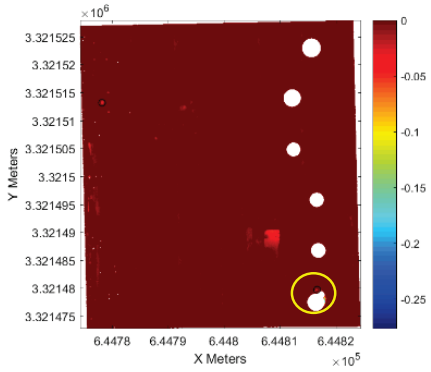
Spall Fields
50’ 9m/s
1 FP



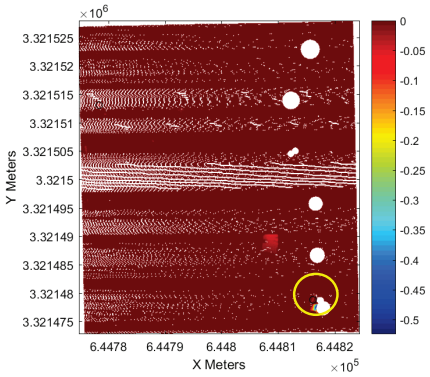
100’
2 FP



200’
1 FP (left is on grass)



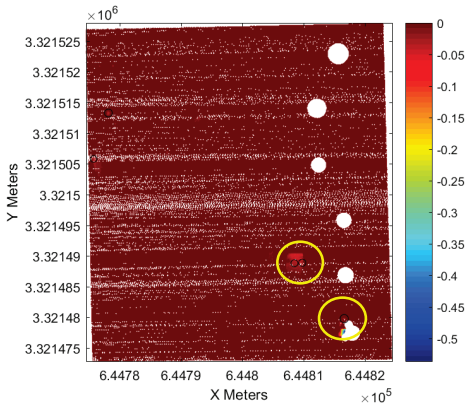
260’
1 FP



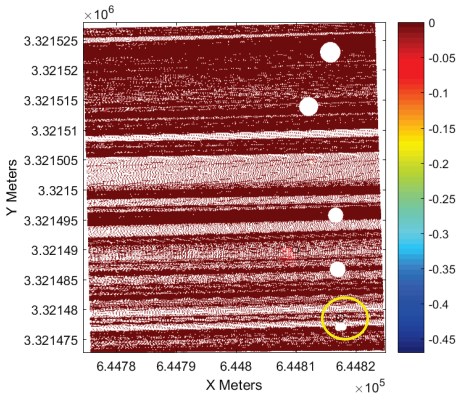
Tile 4

Spall Fields

330’
2 FP



400’
1 FP

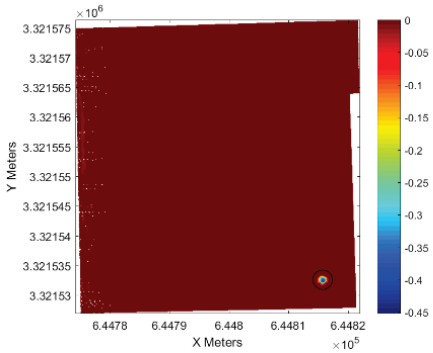


Tile 5

Craters

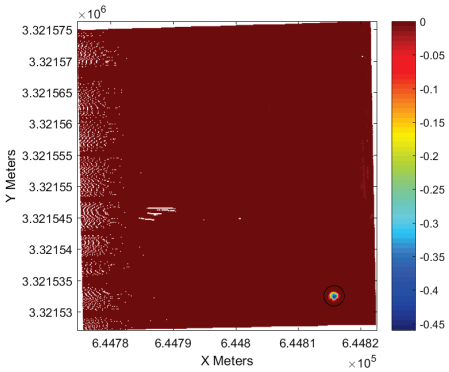
50’ 5 m/s

1/1 Found



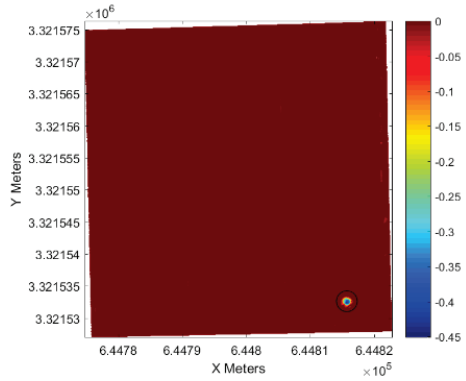
50’ 9 m/s

1/1 Found



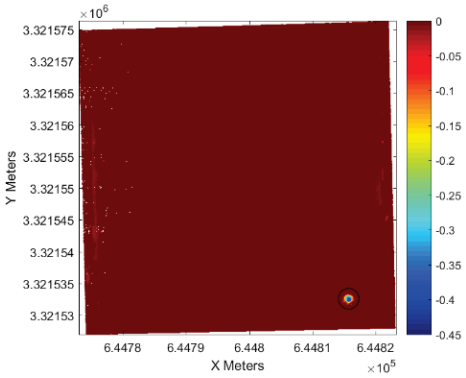
100’

1/1 Found



200’

1/1 Found

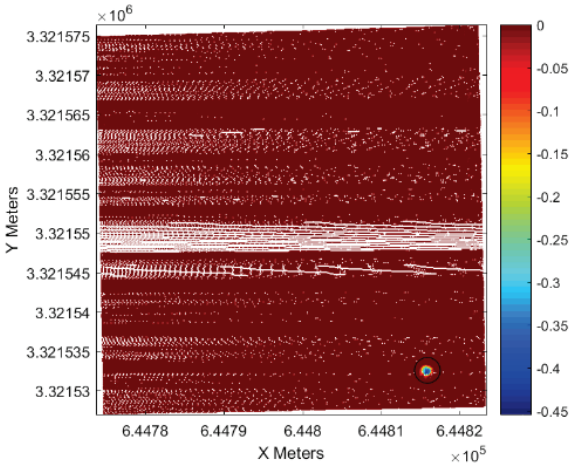


Tile 5

Craters
260’
1/1 Found

330’
0/1 Found

400’
0/1 Found

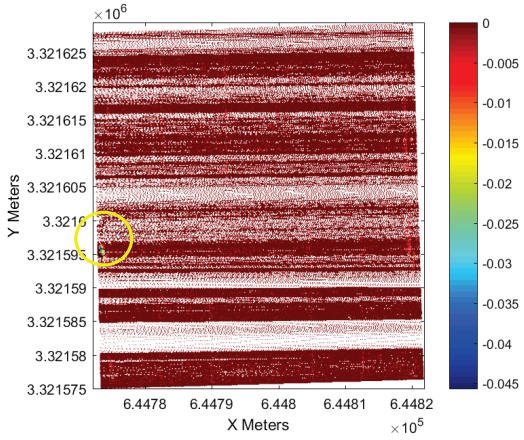


NA

NA

Tile 6

Spall Fields
400’
1 FP



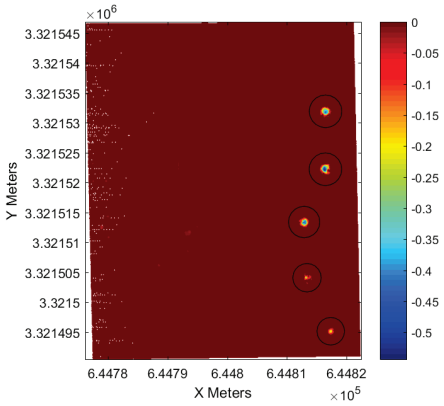
Mar 2019: MATLAB screen captures for Mar 2019 night collection

Tile 2

50 ft-5 m/s

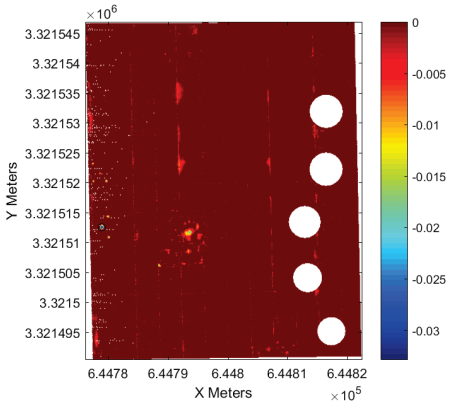
Craters

5/5 Found



Spalls

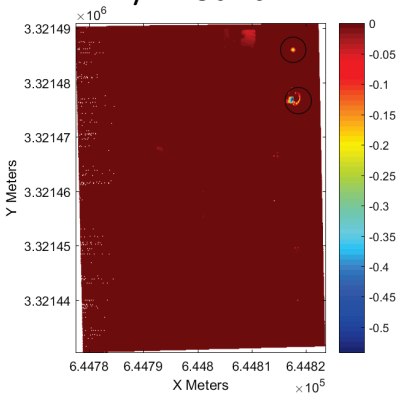
1 FP



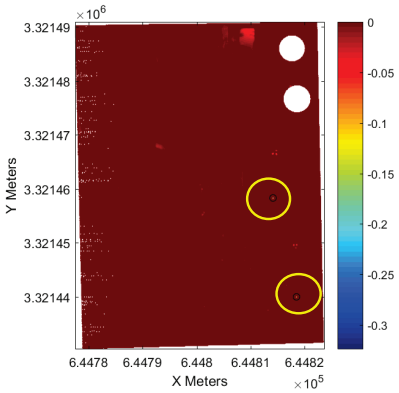
Tile 3

50 ft-5 m/s

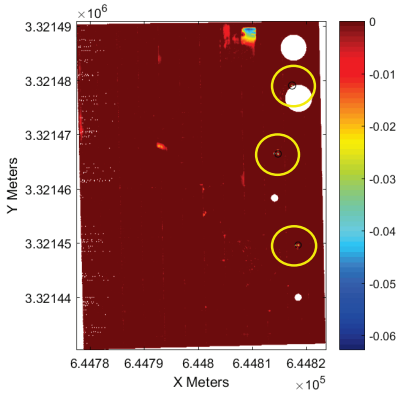
Craters
2/2 Found



Camoflets
2/2 Found



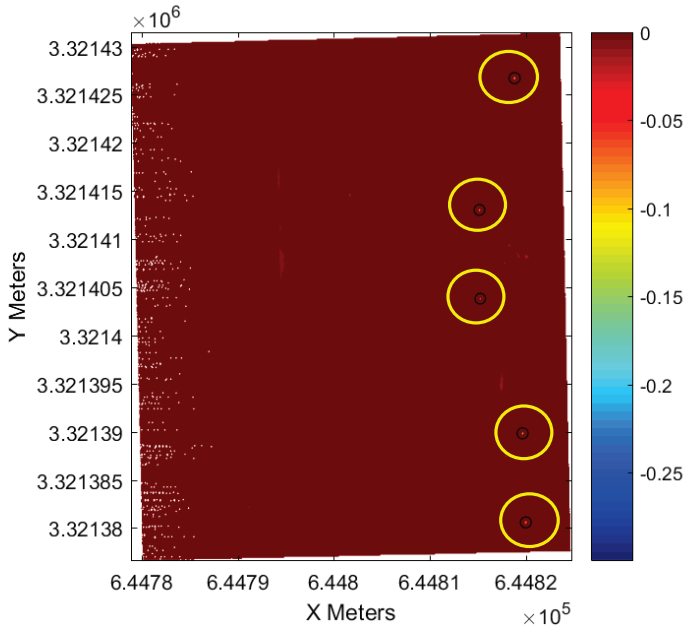
Spalls
2/2 Found, 1 FP (crater not fully removed)



Tile 4

50 ft-5 m/s

Camoflets
5/5 Found



Unit Conversion Factors

Multiply	By	To Obtain
cubic feet	0.02831685	cubic meters
cubic inches	1.6387064 E-05	cubic meters
degrees (angle)	0.01745329	radians
feet	0.3048	meters
inches	0.0254	meters
miles (U.S. statute)	1,609.347	meters
miles per hour	0.44704	meters per second
pounds (force)	4.448222	newtons
pounds (mass)	0.45359237	kilograms
square feet	0.09290304	square meters
square inches	6.4516 E-04	square meters
yards	0.9144	meters

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 instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this 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 Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE *(DD-MM-YYYY)*
March 2021

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*

4. TITLE AND SUBTITLE

First Generation Automated Assessment of Airfield Damage using LiDAR Point Clouds

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Ernest S. Berney IV, Andrew B. Ward, and Naveen B. Ganesh

5d. PROJECT NUMBER
471376 and 480403

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Geotechnical and Structures Laboratory
U.S. Army Engineer Research and Development Center
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

8. PERFORMING ORGANIZATION REPORT NUMBER

ERDC TR-21-4

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Corps of Engineers
Washington, DC 20314-1000

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

Rapid Airfield Damage Assessment System (RADAS)

Conducted for United States Air Force Civil Engineering Center under "Airfield Damage Detection Using LiDAR Point Clouds" and "Automatic Pavement Damage Assessment using LiDAR"

14. ABSTRACT

This research developed an automated software technique for identifying type, size, and location of man-made airfield damage including craters, spalls, and camouflets from a digitized three-dimensional point cloud of the airfield surface. Point clouds were initially generated from Light Detection and Ranging (LiDAR) sensors mounted on elevated lifts to simulate aerial data collection and, later, an actual unmanned aerial system. LiDAR data provided a high-resolution, globally positioned, and dimensionally scaled point cloud exported in a LAS file format that was automatically retrieved and processed using volumetric detection algorithms developed in the MATLAB software environment. Developed MATLAB algorithms used a three-stage filling technique to identify the boundaries of craters first, then spalls, then camouflets, and scaled their sizes based on the greatest pointwise extents. All pavement damages and their locations were saved as shapefiles and uploaded into the GeoExPT processing environment for visualization and quality control. This technique requires no user input between data collection and GeoExPT visualization, allowing for a completely automated software analysis with all filters and data processing hidden from the user.

15. SUBJECT TERMS

Unmanned Aerial System (UAS)
Runways (Aeronautics)—Maintenance and repair
Drones
MATLAB
Light Detection and Ranging (LiDAR)
Airfields
Damage
Military bases
Air bases
Craters
Detection
Automated
Optical radar
Remote sensing

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(include area code)*
Unclassified	Unclassified	Unclassified	SAR	177	