**US Army Corps
of Engineers®**
Engineer Research and
Development Center

Methodology for Remote Assessment of Pavement Distresses from Point Cloud Analysis

Ernest S. Berney IV, Naveen B. Ganesh, Andrew B. Ward,
J. Kent Newman, and John F. Rushing

April 2021

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.erdclibrary.on.worldcat.org/discovery.

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

Methodology for Remote Assessment of Pavement Distresses from Point Cloud Analysis

Ernest S. Berney IV, Andrew B. Ward, J. Kent Newman, and John F. Rushing

*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 #458278, "Point Cloud Analyses of Airfield Surfaces," and Project
#480380, "LiDAR and Photogrammetry for Terrain Analysis"

Abstract

The ability to remotely assess road and airfield pavement condition is critical to dynamic basing, contingency deployment, convoy entry and sustainment, and post-attack reconnaissance. Current Army processes to evaluate surface condition are time-consuming and require Soldier presence. Recent developments in the area of photogrammetry and light detection and ranging (LiDAR) enable rapid generation of three-dimensional point cloud models of the pavement surface. Point clouds were generated from data collected on a series of asphalt, concrete, and unsurfaced pavements using ground- and aerial-based sensors. ERDC-developed algorithms automatically discretize the pavement surface into cross- and grid-based sections to identify physical surface distresses such as depressions, ruts, and cracks. Depressions can be sized from the point-to-point distances bounding each depression, and surface roughness is determined based on the point heights along a given cross section. Noted distresses are exported to a distress map file containing only the distress points and their locations for later visualization and quality control along with classification and quantification. Further research and automation into point cloud analysis is ongoing with the goal of enabling Soldiers with limited training the capability to rapidly assess pavement surface condition from a remote platform.

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.....	v
Preface	ix
1 Introduction	1
1.1 Purpose	1
1.2 Scope.....	2
1.3 Deliverables	3
2 Methodology.....	4
2.1 Photogrammetry	6
2.1.1 Alignment	6
2.1.2 Georectification.....	8
2.1.3 Dense reconstruction	8
2.1.4 Product generation	9
2.2 Light Detection and Ranging (LiDAR)	14
2.2.1 Ground-based LiDAR.....	16
2.2.2 Elevated LiDAR.....	17
3 Data Collection Events.....	19
3.1 Hagler Army Airfield – July 2015	20
3.1.1 Static imaging	21
3.1.2 Photogrammetry point cloud rendering.....	22
3.1.3 LiDAR point cloud rendering	23
3.2 Selah Army Airfield	25
3.2.1 LiDAR reconstruction.....	26
3.2.2 Vehicle-based photogrammetry	27
3.3 Grayling Army Airfield	28
3.4 Pope Army Airfield	29
3.5 Camp Rudder	31
3.5.1 Telehandler LiDAR	32
4 MATLAB Algorithm Development.....	36
4.1 Mesh creation methods	36
4.2 MATLAB functions for damage detection	40
4.2.1 Crack detection method	40
4.2.2 Crack map	41
4.2.3 Crack widths.....	42
4.3 MATLAB functions for volumetric distress detection	43
4.3.1 Algorithm introduction	44
4.3.2 Initialization	44
4.3.3 Surface flooding.....	44

4.3.4	Surface draining.....	45
4.3.5	Isolating depressions.....	47
4.4	MATLAB implementation of crack detection.....	48
4.5	MATLAB implementation of depression detection.....	52
4.6	Indexes of roughness measurement.....	54
4.6.1	Boeing Bump Index.....	54
4.6.2	International roughness index	55
4.7	MATLAB implementation of roughness calculation.....	57
4.7.1	Construction of cross section.....	57
4.7.2	ProFAA analysis of cross section.....	61
5	Data Analysis.....	64
5.1	Detection of crack features.....	64
5.1.1	Hagler AAF.....	64
5.1.2	Selah AAF.....	67
5.1.3	Grayling AAF.....	69
5.1.4	Pope Army Airfield.....	71
5.1.5	Camp Rudder	73
5.2	Detection of volumetric distresses.....	75
5.2.1	Hagler AAF.....	76
5.2.2	Grayling AAF.....	77
5.2.3	Camp Rudder	80
5.3	Detection of surface roughness	88
5.3.1	Camp Rudder	88
5.3.2	ERDC Cumberland Road – LiDAR versus photogrammetry	93
6	Conclusions and Recommendations.....	100
6.1	Conclusions.....	100
6.2	Recommendations and future research	101
	References.....	103
	Unit Conversion Factors.....	106
	Report Documentation Page	

Figures and Tables

Figures

Figure 1. Google maps capture of a taxiway at Pope Army Airfield, NC.	4
Figure 2. Pixel-level dense reconstruction.....	9
Figure 3. Product generation screenshots.	9
Figure 4. DJI Phantom 3 UAS (left) and Turbo Ace Matrix-E UAS (right).	12
Figure 5. Two-man operational team for Matrix-E UAS.	13
Figure 6. Example of Matrix-E flight path displayed on Panasonic Android tablet.	13
Figure 7. Illustration of LiDAR pulse and return.	15
Figure 8. View of a crack from a LiDAR point scan.	15
Figure 9. View of a depression from a LiDAR point scan.....	16
Figure 10. Reigl VZ-2000 LiDAR scanner mounted on vehicle.	17
Figure 11. Reigl VZ-2000 LiDAR scanner mounted on cage on telehandler lift.	18
Figure 12. Airfield test locations for LiDAR and photogrammetry.....	19
Figure 13. Layout and condition of Hagler Army Airfield at Camp Shelby, MS.....	20
Figure 14. VZ-2000 LiDAR system mounted on vehicle for mobile data collection.	21
Figure 15. Vehicle-mounted Canon Mark 5-D camera on elevated boom for image capture.	22
Figure 16. View of Photoscan point cloud generation of HAAF Ramp B.	23
Figure 17. View of LiDAR point cloud intensity of HAAF parking apron.....	24
Figure 18. Comparison of colorized LiDAR (left) and Photogrammetry (right) point clouds.	24
Figure 19. Image of Selah Army Airfield Taxiway.	25
Figure 20. Final mounted LiDAR scanner on truck.	26
Figure 21. LiDAR data collection pattern at Selah Army Airfield.	26
Figure 22. LiDAR point cloud of Selah AAF Runway with severe block cracking.....	27
Figure 23. LiDAR point cloud of Selah Army Airfield at transition between old and new asphalt.	27
Figure 24. Canon Mark V camera mounted on vehicle boom.....	28
Figure 25. Operating surfaces at Grayling AAF (2013).....	29
Figure 26. LiDAR point cloud of asphalt section of Grayling Army Airfield.....	29
Figure 27. Survey areas at Pope AAF (2018).	30
Figure 28. LiDAR point cloud of asphalt distresses at Pope AFB.....	30
Figure 29. LiDAR point cloud of concrete distresses at Pope AFB.	31
Figure 30. Aerial view of pavement sections surveyed at Camp Rudder.	32
Figure 31. 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 (lower right).	33
Figure 32. Example of VZ-2000 mounted to an aluminum cage hoisted on a telehandler lift providing a sensor height of approximately 50 ft above ground level.....	34
Figure 33. LiDAR point cloud of transition between old and new concrete on main runway at Camp Rudder.....	35
Figure 34. Photograph taken from the telehandler tower while driving on main runway.	35
Figure 35. Delaunay triangulation and Voronoi Diagram for a random 3-D point cloud input.	38
Figure 36. Example of N x N grid applied to point cloud and grid averaging and grid standard deviation algorithms applied to cell.	39
Figure 37. Surface generated using grid standard deviation method (σ_z values contoured by color).	41
Figure 38. Locations of points associated with high z standard deviation values (i.e. cracks).....	42
Figure 39. Methodology of grid standard deviation and grid average return method with a crack.....	43
Figure 40. No-sink algorithm flooding example.....	45
Figure 41. Depression-filling example on damaged pavement surface (left, point cloud; middle, 500x500 mesh; right, depression-filled mesh.	47
Figure 42. Before and after pavement potholes using a depression-filling algorithm.	48
Figure 43. MATLAB-developed GUI for crack and rut detection.	49
Figure 44. Load file portion from the GUI.....	49
Figure 45. Mesh plot of imported post-surface from the GUI.....	50
Figure 46. Crack Detection portion of the GUI.....	50
Figure 47. Mesh plot of zDeviation contour from the GUI.	51
Figure 48. Crack extraction portion of the GUI.	51
Figure 49. Crack map plotted on the mesh in the GUI.	52
Figure 50. Rut, potholes, depression detection portion of the GUI.	52
Figure 51. Result from the depression filling algorithm done on the post surface.	53
Figure 52. Rut, potholes, depression extraction portion of the GUI.	53
Figure 53. Pothole region plotted on the mesh in the GUI.....	54
Figure 54. Boeing roughness criteria from 1995 to present.	55
Figure 55. International roughness index quarter-car model.	56
Figure 56. MATLAB-developed GUI for roughness calculation.	58
Figure 57. User selection of path and path width to determine X-Y-Z reference points.....	58
Figure 58. Load file portion from the GUI.....	59
Figure 59. Mesh plot of imported data from the GUI.....	59

Figure 60. Parameters for cross-section portion from the GUI.....	60
Figure 61. Setting up cross section and cross-section display from the GUI.	60
Figure 62. Plot of average z-elevation versus R distance along user selected path.	61
Figure 63. Export and ProFAA portion from the GUI.	62
Figure 64. Convert txt file to ProFAA Format.	62
Figure 65. ProFAA profile and roughness indexes application.	63
Figure 66. Hagler AAF concrete apron with severe cracking and spalling.....	65
Figure 67. Detection of standard deviation of z-elevation on concrete segment at Hagler AAF using MATLAB crack detection algorithm.....	66
Figure 68. Crack extraction performed on MATLAB crack detection algorithm.	66
Figure 69. LiDAR point cloud of concrete apron at Hagler AAF with overlain crack locations from MATLAB extraction (top) and isolated rut map (bottom).	67
Figure 70. Block and L&T cracking found on main runway at Selah AAF; extracted points overlain on colorized LiDAR.	68
Figure 71. Magnification of crack features overlain on colorized LiDAR at Selah AAF.	69
Figure 72. Colorized LiDAR segment of Grayling AAF.....	69
Figure 73. MATLAB crack extraction algorithm for Grayling AAF segment.....	70
Figure 74. Crack map of Grayling AAF segment (top) and overlaying LiDAR (bottom).....	70
Figure 75. Example of crack map detail and ability to avoid filled cracks.....	71
Figure 76. MATLAB intensity crack extraction at Pope AAF Apron.....	72
Figure 77. Colorized LiDAR (left) and extracted crack map (right) from Pope AAF Apron.	72
Figure 78. Pope AAF Apron crack map overlain on LiDAR and detail of extracted crack point alignment with LiDAR.....	73
Figure 79. Detection of cracks using MATLAB algorithm at Camp Rudder.....	74
Figure 80. Crack extraction using MATLAB algorithms at Camp Rudder.	74
Figure 81. LiDAR point cloud of main runway transition at Camp Rudder with overlain crack locations from MATLAB extraction (top) and isolated crack map (bottom).....	75
Figure 82. MATLAB volumetric distress extraction on Hagler AAF apron.....	76
Figure 83. Volumetric distress points overlain LiDAR point cloud of spalled crack at Hagler AAF apron.....	77
Figure 84. MATLAB volumetric distress GUI at Grayling AAF.....	78
Figure 85. Grayling AAF segment colorized LiDAR (top) and volumetric distress map (bottom).....	79
Figure 86. Volumetric distress map overlain on LiDAR point cloud for Grayling AAF.	80
Figure 87. Volumetric distress algorithm applied to sand-asphalt section.	81
Figure 88. Extraction of rut and pothole features from section of sand-asphalt (right) taken from LiDAR point cloud (left).	81

Figure 89. Magnified image of extracted points overlaying LiDAR for potholes and ruts.	82
Figure 90. MATLAB extraction of rut features on main runway at Camp Rudder.	83
Figure 91. LiDAR point cloud of main runway transition at Camp Rudder with overlay rut locations from MATLAB extraction (top) and isolated rut map (bottom).	84
Figure 92. LiDAR point cloud cross section from unsurfaced road at Camp Rudder, plan view (top) and cross-section view (bottom).	85
Figure 93. Extracted volumetric distresses from active road surface at Camp Rudder.	86
Figure 94. LiDAR point cloud of active road surface at Camp Rudder with overlay rut locations from MATLAB extraction (top) and isolated rut map (bottom).	87
Figure 95. Close up of surface depressions in LiDAR outlined with rut-extracted points at active road surface from Camp Rudder.	88
Figure 96. MATLAB analysis of sand asphalt pavement roughness.	89
Figure 97. ProFAA analysis of sand asphalt pavement section.	90
Figure 98. MATLAB analysis of new concrete pavement roughness.	91
Figure 99. ProFAA analysis of new concrete pavement section.	91
Figure 100. MATLAB analysis of unsurfaced road roughness.	92
Figure 101. ProFAA analysis of unsurfaced road section.	92
Figure 102. ERDC Cumberland Road test strip.	93
Figure 103. LiDAR and photogrammetric height map of ERDC Cumberland Road.	94
Figure 104. Colorized LiDAR height map of pothole.	95
Figure 105. Comparison of sharp LiDAR point cloud resolution (left) versus smoothed photogrammetry point cloud (right).	95
Figure 106. IRI from LiDAR and photogrammetry for various cross sections (top) and comparison of values (bottom).	97
Figure 107. IRI versus distance for LiDAR (left) and photogrammetry (right) over several cross sections near water-filled depression.	98
Figure 108. IRI from LiDAR and photogrammetry for various cross sections (top) and comparison of values (bottom) with LiDAR missing values retained.	99

Tables

Table 1. Location of data collection events.	19
Table 2. Ground LiDAR point densities at various speeds for HAAF.	21
Table 3. No-sink surface draining scan directions.	46

Preface

This study was conducted for the Office of the Assistant Secretary of the Army for Acquisition and Logistics under Project Numbers 458278, “Point Cloud Analyses of Airfield Surfaces,” and 480380, “LiDAR and Photogrammetry for Terrain Analysis.” The technical monitors were Dr. J. Kent Newman and Dr. John F. Rushing.

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 C. 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 ERDC-GSL was Mr. Charles W. Ertle II, and the Director was Mr. Bartley P. Durst. The Deputy Director of ERDC-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.

1 Introduction

1.1 Purpose

Force projection within the Army requires the ability to transport Soldiers and materiel to remote locations throughout the world. Often this deployment requires access, whether through force or contingency, to pavement surfaces. Knowledge of the condition of these paved surfaces determines the type, number, and payload of vehicle that can operate, which has serious implications on mission planning. However, the Army does not have an efficient means to remotely assess the condition of paved surfaces, relying solely on coarse evaluations of slope, curvature, length, and width. The performance of a condition survey requires placing Soldiers physically on the pavement to assess the ground condition. This places Soldiers in harm's way and potentially reveals strategic entry points to the opposition.

The pavement surface condition is assessed by measuring the number, type, and severity of the physical distresses present. These distresses in turn describe the amount of pavement degradation that has occurred from either traffic or environmental factors. For example, the presence of severe longitudinal surface roughness can damage aircraft during take-off and landing. Likewise, severe depressions and rutting can hinder convoy speeds. Currently, the general condition of a paved surface can be determined by measuring the pavement condition index (PCI), a procedure developed by the U.S. Army Corps of Engineers (Shahin and Starr 1978). The PCI is determined through a procedure of identifying the type, extent, and severity of surface defects as defined in the PAVER (Shahin et al. 1977) Distress Identification Manuals for asphalt and concrete pavement surfaces. These distresses range from cracks and faulting to spalling and potholes to other defects, such as weathering and jet blast erosion. Determining the PCI for pavement surfaces is a time-consuming process that requires multiple, trained personnel working over hours and days visually assessing the pavement surface capturing only a statistical subset of the pavement.

Surface roughness is determined by measuring the change in surface elevation over a given length of paved surface and varying this measurement over several scales. Various indices have been developed to

characterize these changes in elevation including the International Roughness Index (IRI) for vehicles and aircraft and the Boeing Bump Index (BBI) specifically for aircraft. These indices determine the vibrational effects on a vehicle based on axle spacing. A roughness determination of the pavement surface is time-consuming as well, requiring specialized vehicle-mounted equipment such as inertial profilers (Spangler and Kelly 1966) that can detect small changes in surface elevation while driving along the pavement surface.

To provide the PCI and IRI values for a pavement surface without involving the Soldier, an alternative technology is sought that will enable data collection from aerial reconnaissance. Data processing can occur in a secure location away from the areas of interest. This report documents the use of point clouds derived from photogrammetry and Light Detection and Ranging (LiDAR) sensors deployed from mobile and unmanned aerial systems (UASs) as a means to collect these data and the software systems that can process these data and identify the key distress values required to assess pavement surfaces.

1.2 Scope

To solve the problem of collecting remote data that can be used to evaluate pavement surface condition, this research explores the use of imagery derived from electro-optical (EO) cameras or LiDAR sensors that can be deployed from an UAS. This imagery is processed into three-dimensional point cloud models of the pavement surface allowing visualization of cracks, ruts, and depressions. Processing of photographs into three-dimensional point clouds requires the use of the Structure from Motion (SfM) algorithms that merge photographic images into a point cloud based on feature mapping between adjacent photographs. LiDAR sensors directly generate a real-world dimensional point cloud and includes intensity of light return as an additional detection data field. These point clouds provide an accurate, comprehensive, and expedient procedure to evaluate the pavement surface condition allowing the use of software that can identify the physical presence of specific surface distresses.

Photographs were taken from a Sony A6000 digital camera mounted normal to the pavement surface on a Turbo Ace Matrix-E quad-rotor UAS flying at elevations greater than 50 ft above the surface. The captured digital images were converted into three-dimensional point clouds using Agisoft Photoscan software. LiDAR was collected using a Reigl VZ-2000

mobile sensor that was mounted either to the back of a vehicle for early data collection and later suspended approximately 50 ft above the ground surface on a telescoping boom to simulate UAS deployment.

ERDC-developed algorithms in MATLAB read the point cloud files and automatically perform a series of cross-sectional and grid-based analyses across the airfield surface to determine physical deviations from a smooth runway surface such as depressions, ruts, and cracks. Crack widths can be determined from the location of adjoining points, and surface roughness can be determined based on vertical changes in elevation along a given cross section. Distresses identified with the MATLAB detection algorithms are exported to a distress map file containing only the distressed points and their locations. This allows for easy visualization to the user and processing for classification. Further research and automation into point cloud analysis is ongoing with the goal of enabling Soldiers to rapidly assess pavement surface condition from a remote platform.

1.3 Deliverables

This report documents ERDC's progress made to date in providing a comprehensive solution to remote assessment of pavements using point cloud clouds. The documentation of technical solutions developed includes:

- The process used to obtain and analyze EO images and LiDAR for point cloud reconstruction at a scale significant to identify surface roughness and crack type, including data collection activities to provide the basis for the software development.
- The software layout and operation for using point clouds and identifying crack and volumetric distress features and vertical elevation profiles.
- The output of the crack and distress maps and the work done to provide a visualization of these files.
- The software integration of the MATLAB tools with ProFAA software allowing calculation of IRI and BBI.

2 Methodology

Currently, any assessment of a military pavement can be conducted either by walking, driving, or flying via the following techniques.

1. Boots-on-the-ground approach whereby one or typically multiple Soldiers walk a pavement surface, document physical distresses, and physically measure the length, width, and extent of these distresses. Roughness must be assessed with specialized equipment that is not readily available.
2. Vehicle-based assessment using tools such as the Automated Route Reconnaissance Kit (ARRK) (Lynch et al. 2005) or a rudimentary windshield inspection noting road restrictions such as narrow bridges, steep slopes, or sharp curves as noted in Army Field Manual ATP 3-34 (U.S. Army 2016).
3. Aerial imagery analysis typically in the form of satellite images (Figure 1). Imagery at this altitude can provide only a cursory look of the surface condition, and little in the way of measured quantities can be ascertained. Additionally, the images may not be up-to-date.

Figure 1. Google maps capture of a taxiway at Pope Army Airfield, NC.

Approach number one is time-consuming, reveals the military's interest in a specific location, and potentially puts Soldiers in danger. The issues with approach number two are:

1. Requires Soldiers to be present all along the road network of interest, exposing both the Soldier and the mission to compromise.
2. Provides little in assessing the state of the roadway for future traffic.
3. Requires a complex set of manually documented short-hand notation that is not readily understood and rarely used by the Soldier.

The issues with approach number three are interpretation of imagery can lead to false indicators of pavement condition:

1. Rutting and depressions are difficult to identify in a 2-D image.
2. It is difficult to determine treated cracks from untreated cracks on a concrete surface as both appear black in the image.
3. It is difficult to determine crack location on asphalt surfaces where colorization causes difficulty in separating shadows from pavement.
4. Images are susceptible to poor contrast due to time of day and are inaccessible during nighttime reconnaissance.

To overcome these limitations, a new approach to imaging the pavement surface was undertaken using two modern technologies.

1. Photogrammetry: Photographs taken from a simple camera mounted to an unmanned aerial system provide low-altitude, high-pixel density images that, when taken in sequence, generate a three-dimensional point cloud capable of producing scale and depth to the surface.
2. LiDAR: An active LiDAR sensor mounted to an elevated platform to simulate an unmanned aerial system provided high-density three-dimensional point clouds with automated geo-positioning, providing both scale and depth to the surface.

The intent of the research was to establish a procedure whereby the pavement surface is represented in a scaled, three-dimensional point cloud so that engineering calculations can be performed. By digitizing the airfield surface at a high resolution (points separated by 1 in. or less), software can automate the assessment of the surface condition based on parameters a person may use during a physical inspection without having to place Soldiers in the area. The following sections describe the process of each imaging process that was evaluated during the research.

2.1 Photogrammetry

Photogrammetry is the science of calculating the distance between points on a three-dimensional (3-D) surface from measurements made on two-dimensional (2-D) photographs. This allows for the reconstruction of a 3-D scene using only 2-D imagery. Problems associated with 3-D reconstruction are difficult and long-studied. To perform a pixel-level reconstruction of a 3-D scene requires either ample time or immense computational resources. Three-dimensional reconstruction is made even more difficult by the diverse makeup of the computational requirements of the various steps of the 3-D reconstruction process. For example, mesh generation has strict memory and CPU requirements, where pixel-level dense reconstruction algorithms, like multi-view stereo, have heavy graphics processing unit (GPU) requirements. Recent advances in the parallel processing of 2-D imagery for the purpose of 3-D reconstruction allows for the timely reconstruction of a 3-D scene.

The algorithms supporting the process of 3-D reconstruction are complex and time-consuming such that, the use of commercial-off-the-shelf (COTS) photogrammetry software is often preferred to developing in-house solutions. Examples of popular COTS 3-D reconstruction software are Agisoft Photoscan, Pix4D Mapper, and Bentley ContextCapture. Steps necessary to complete the 3-D reconstruction process as they relate to the present work are detailed below. The process can be separated into five categories, i.e., data collection, Structure from Motion (SfM), georectification with ground control, dense reconstruction, and product generation (Verhoeven et al. 2013).

2.1.1 Alignment

In the alignment, or SfM, step, 2-D photos are aligned with each other based on detected unique features present in multiple photos. The principal steps in the alignment phase have been discussed in various works (Ullman 1979, Westoby et al. 2012, Verhoeven et al. 2013) dating as far back as the 1950s where the process was known as the “kinetic depth effect” (Wallach and O’Connell 1953).

2.1.1.1 Feature detection

The first part of alignment is feature detection. In this step, interest points (IPs) are detected that are highly repeatable, i.e., invariant, in the 2-D

photos, to rotation and distortion. An IP is a point in the image that is distinct and has a distinct neighborhood of points. Multiple IPs are needed in each 2-D image for sufficient alignment.

2.1.1.2 Feature description

The next step comprises the computation of feature descriptors relating the IPs and their local area. Descriptor information typically relies on the intensity profile of the local area. Popular feature detection techniques include descriptor matching. Many time-optimized feature detection/description techniques are available.

- Speeded Up Robust Features (SURF) (Bay et al. 2008)
- Scale-invariant Feature Transform (SIFT) (Lowe 1999)
- Binary Robust Invariant Scalable Keypoints (BRISK) (Leutenegger et al. 2011)
- Features from Accelerated Segment Test (FAST) (Rosten and Drummond 2005)

2.1.1.3 Descriptor matching

Following the detection and description of unique IPs, individual IP descriptors are matched in multiple images. This is performed by first defining the Euclidean distance between descriptors in a pair of images. Shorter Euclidean distance between two considered descriptors indicates likely matching. Robust outlier analysis is then performed to cull false positives. The most popular such outlier removal approach is Random Sampling Consensus (RANSAC) (Fischler and Bolles 1980). In this approach, a least-squares straight line is fit to a randomly sampled set of matched IPs until a pre-determined goodness-of-fit threshold is met.

2.1.1.4 Triangulation

Once inlier IPs are selected, their positions are determined through observations of the individual IPs in two or more images. This procedure allows for the creation of a local coordinate system scaled in metric or imperial units.

2.1.1.5 Bundle adjustment

The minimization of calculated re-projection errors with respect to intrinsic and extrinsic camera parameters is performed here (Triggs et

al. 1999). This step minimizes actual error in the 3-D scene reconstruction in all images simultaneously, as opposed to the pair-wise matching performed in previous steps.

2.1.2 Georectification

Ground control points (GCPs) are used to translate the local SfM coordinate system into a global coordinate system like World Geodetic System 1984 (WGS84), Universal Transverse Mercator (UTM), etc. Furthermore, scaling and geolocation accuracy will depend heavily on ground-control accuracy and placement. Pattern recognition features are available in popular 3-D reconstruction software (Pix4D and Photoscan) to use automatically detectable GCPs in scene reconstruction.

2.1.3 Dense reconstruction

A 3-D scene is reconstructed at the pixel level (Figure 2) as opposed to the feature level in the previous steps. Typical pixel-level dense reconstruction is performed using a dense multi-view stereo (MVS) approach. In general, the MVS approach consists of generating candidate pixel pairs from multiple images and determining the probability of matching them using photo-consistency measures. Variability within COTS software in the 3-D reconstruction process is likely due to proprietary implementations of this step. Most photogrammetric software will use a type of MVS for pixel-level dense reconstruction. The MVS algorithm is usually complemented by a camera projection model like the pinhole camera model. The pinhole camera model incorporates the extrinsic and intrinsic camera properties into a projection matrix. The projection matrix includes properties like focal length, principal point, and skew (Furukawa and Hernandez 2015).

Figure 2. Pixel-level dense reconstruction.

2.1.4 Product generation

Once pixel-level 3-D reconstruction is accomplished, data products like textured meshes (Figure 3a), orthomosaics (Figure 3b), and digital elevation models (DEMs, Figure 3c) can be generated.

Figure 3. Product generation screenshots.

a) Textured mesh

b) Georeferenced orthomosaic

c) Georeferenced DEM

2.1.5 Electro-optical imager

This research made use of EO cameras mounted on a UAS to provide images that were oriented normal to the pavement surface. Normal images were found to be optimal to best identify defects in the pavement by enabling penetration of the developed three-dimensional point cloud into small surface defects that might otherwise be missed from images taken at oblique angles. Two UAS platforms were used in this research. The first was a DJI Phantom 3, and the second was a Turbo Ace Matrix-E with a Sony A6000 camera (Figure 4). The Phantom provided a simple user interface for quick data collection and a high-definition (4K) video camera to allow for rapid, high-quality imaging of the runway surface. The Matrix-

E provided a customizable payload platform that enabled mounting of a 24-mp camera for higher quality static ground images and the capability to add additional sensors, thermal, etc., for future data collection events.

Figure 4. DJI Phantom 3 UAS (left) and Turbo Ace Matrix-E UAS (right).

The DJI was piloted by a single individual using an iPhone as the user interface. The Matrix-E operations involved a two-man process with a pilot and an individual monitoring the flight path and battery performance with a Panasonic FZ-B2 Android tablet (Figure 5 and Figure 6).

Figure 5. Two-man operational team for Matrix-E UAS.

Figure 6. Example of Matrix-E flight path displayed on Panasonic Android tablet.

Two-dimensional EO data were collected with a Sony A6000 camera using a 35-mm fixed focus lens. The A6000 collected EO data at a resolution of approximately 24 megapixels for an image resolution of $6,000 \times 4,000$ pixels. Photos were collected to allow for the overlap of images in the direction of travel (front-lap) and in the direction transverse to travel (side-lap). Image overlap, ω , is a function of image capture displacement, d , flight altitude, h , and field of view (FOV), α (see Equation below).

$$\omega = 1 - \frac{d}{2h} \cot\left(\frac{\alpha}{2}\right)$$

Side-lap is calculated by using the horizontal FOV, and front-lap is calculated using the vertical FOV. Using the Sony A6000 equipped with a 35-mm lens, the horizontal FOV is 54.4° , and the vertical FOV is 37.8° . Typically, the overlap equation is solved for the image capture displacement and used to determine an intervalometer time, such that, if the speed of the UAS, v , is known, and the distance is calculated using the above equation, then the time between image captures, t , is calculable (see Equation below).

$$t = \frac{2h}{v} (1 - \omega) \tan\left(\frac{\alpha}{2}\right)$$

2.2 Light Detection and Ranging (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 7). The LiDAR system used in this work has an attached global positioning sensor (GPS) 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 7. Illustration of LiDAR pulse and return.

The use of LiDAR enables detection of small variations in surface texture allowing distinction of cracks and depressions that are indicators of pavement degradation. Figure 8 and Figure 9 illustrate the detail at which a LiDAR scan of pavement can provide in finding both cracks and depressions. The benefit of the point cloud is that, being digitized, it enables engineering calculations to be made, including width, depth, and length.

Figure 8. View of a crack from a LiDAR point scan.

Figure 9. View of a depression from a LiDAR point scan.

2.2.1 Ground-based LiDAR

The terrestrial LiDAR system employed in this research, a Riegl VZ-2000 mobile LiDAR unit, was used as the primary data collection sensor. It was mounted in two configurations for this research; initially, in a horizontal configuration on the rear of a pickup truck to enable rapid movement along the pavement (Figure 10). 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 pavement 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 at times to fully penetrate the depth of cracks noted on the pavement. Further, the low height (< 10 feet) at which the sensor was deployed required multiple passes when assessing an 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 slowing the analysis process and requiring more subject matter expert involvement.

Figure 10. Reigl VZ-2000 LiDAR scanner mounted on vehicle.

2.2.2 Elevated LiDAR

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 scan the pavement surface in a single swath (Figure 11). The telehandler-mounted LiDAR worked especially well, providing a wide area of coverage and data nearly normal to the pavement surface, which allowed penetration into cracks. 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 11. Reigl VZ-2000 LiDAR scanner mounted on cage on telehandler lift.

3 Data Collection Events

Data collection during development of the point cloud capability occurred in several locations experimenting with both asphalt and concrete surfaces primarily focused on airfields. Data collection occurred at the following locations shown in Table 1 and illustrated in Figure 12.

Table 1. Location of data collection events.

Name of Airfield	Geographic Location	Surface	Sensor Deployed
Hagler Army Airfield	Camp Shelby, MS	Concrete	Mobile LiDAR, Photogrammetry
Selah Army Airfield	Yakima Training Center, WA	Asphalt	Mobile LiDAR
Grayling Army Airfield	Camp Grayling, MI	Concrete, Asphalt	Mobile LiDAR
Pope Army Airfield	Pope Air Force Base, NC	Concrete, Asphalt	Mobile LiDAR
Camp Rudder	Eglin Air Force Base, FL	Concrete, Asphalt, Soil	Mobile LiDAR, Photogrammetry

Figure 12. Airfield test locations for LiDAR and photogrammetry.

Airfields were targeted specifically because they represented small, defined surfaces that were easy to survey with the mobile and aerial equipment. Each selected site coincided with a manual pavement assessment (PCI) being conducted in conjunction with the point cloud collection. This was

3.1 Hagler Army Airfield – July 2015

Figure 13. Layout and condition of Hagler Army Airfield at Camp Shelby, MS.

measurement program operated at 1010 kHz with a line speed of 239.85 lines per second and a degree increment of 0.0570 deg. The LiDAR data were output in LAS format in UTM Zone 16N NAD 83 coordinate system with resulting units in meters. These collection parameters were adhered to for all LiDAR data collection events.

Traveling at near constant speeds, when possible, with a target distance of no more than 4 m from the sensor and collecting at a rate of 1010 kHz, resulted in the following point densities (Table 2).

Table 2. Ground LiDAR point densities at various speeds for HAAF.

Speed (km/h)	Line Distance (m)	Point Distance (m)	Point Density pts/m ²
8	0.0093	0.0099	10864.00
16	0.0185	0.0099	5432.13
32	0.0371	0.0099	2716.07

Figure 14. VZ-2000 LiDAR system mounted on vehicle for mobile data collection.

3.1.1 Static imaging

Ground based static imaging was achieved through the use of a vehicle-mounted platform that could be controlled from within the cab (Figure 15). In this configuration, a Canon 5D Mark III (36 megapixel) camera was mounted to an extendable boom and pointed normal to the pavement surface at a height of 15 ft above the surface. This camera was driven slowly across the airfield to collect images using an automated exposure setting. An overlap of between 60% longitudinally and 40%

laterally was attempted as this is the minimum value in practice to achieve proper 3-D point cloud reconstruction.

Figure 15. Vehicle-mounted Canon Mark 5-D camera on elevated boom for image capture.

3.1.2 Photogrammetry point cloud rendering

The initial data gathering for the taxiway portion of HAAF resulted in 3,683 photographs taken over a period of 30 min resulting in a LASer (LAS) point cloud file of 21 GB in size and a processing time of over 2 days using Agisoft Photoscan rendering software. The number of photos that identified each pixel in the point cloud (the effective overlap) was 4.7. As a first attempt, this demonstrated that additional research is necessary to reduce the frequency of photographs in order to minimize the data set and enable faster processing. An effective overlap of 4.7 indicates that far too many photos were taken to obtain the necessary detail on the runway surface, and this issue will be the focus of continued efforts on data collection. A processed optical point cloud of the HAAF C-130 concrete parking apron (Ramp B) is shown in Figure 16.

Figure 16. View of Photoscan point cloud generation of HAAF Ramp B.

3.1.3 LiDAR point cloud rendering

A colored LiDAR scan of the runway was completed in approximately 30 min, but this does not account for setup time and satellite acquisition of a GPS base station to provide RTK level positioning of the LiDAR pixels. A combined LAS file of the entire runway including the taxiway encompassed a size of 9.8 GB and was processed in approximately 4 hr using the Riegl RiScan software. A processed color LiDAR point cloud of Ramp B is shown in Figure 17.

Figure 17. View of LiDAR point cloud intensity of HAAF parking apron.

A comparison of a point cloud generated from LiDAR colorized from photographs to a point cloud generated from the ground-based photograph shown in Figure 18. The comparison illustrates very consistent image/distress patterns between techniques and highlights the ability to use either system to combine physical pixel space to identify distresses with visual observation for quality control and referencing. Point cloud densities are far greater for the photographs in this illustration given the high-resolution camera used to capture the imagery.

Figure 18. Comparison of colorized LiDAR (left) and Photogrammetry (right) point clouds.

3.2 Selah Army Airfield

Selah Army Airfield (SAA) represented an asphalt pavement surface that was in very poor condition with substantial block and alligator cracking along with excessive vegetative growth within the cracks (Figure 19). Building on the lessons learned from the HAAF data collection, data were collected using the Riegl VZ-2000 LiDAR sensor mounted to a Ford F-250 camper shell and tilted outward from the vehicle approximately 20 deg from the vertical (Figure 10 and Figure 20). This enabled the LiDAR scanner to capture approximately half the runway width in a single pass to minimize LiDAR overlap and any stitching required to create a complete LAS file.

Figure 19. Image of Selah Army Airfield Taxiway.

Figure 20. Final mounted LiDAR scanner on truck.

3.2.1 LiDAR reconstruction

LiDAR data were collected at a high frequency (1000 kHz) to allow for a rapid vehicle speed (25-30 mph), yet a high point density to determine crack location and width. Data were collected in two passes as shown in Figure 21 to ensure that full airfield coverage was obtained. The processed LiDAR point clouds show the considerable vegetation occurring within the crack features (Figure 22) and the severity of the surface distresses in the closed runway versus a newly constructed extension of the old runway (Figure 23). LiDAR LAS files were generated in segments corresponding to passes and airfield features such as runway, taxiway, and parallel scans. These LAS files were manually stitched together in post-processing to create a uniform point cloud across the full width of the runway for visualization. The optimum method for detecting small surface distresses is when only a single LAS file is analyzed at a time so that no artificial distresses are created at the manually stitched boundary between LAS files.

Figure 21. LiDAR data collection pattern at Selah Army Airfield.

Figure 22. LiDAR point cloud of Selah AAF Runway with severe block cracking.

Figure 23. LiDAR point cloud of Selah Army Airfield at transition between old and new asphalt.

3.2.2 Vehicle-based photogrammetry

The SAA data collect was the last attempt to use photogrammetry from a vehicle-based platform. Photogrammetry was attempted using a ground-based camera system similar to that deployed at HAAF (Figure 24). A

modification applied to this collect was a software/hardware system that attempted to allow for photographs to be taken at a measured distance rather than time to optimize the number of photographs to minimize post-processing times. However, this attempt failed to reliably capture photos at measured intervals resulting in a point cloud with insufficient overlap for point cloud reconstruction. Therefore, data analysis presented in this report originated only from the LiDAR point cloud.

Figure 24. Canon Mark V camera mounted on vehicle boom.

3.3 Grayling Army Airfield

Grayling Army Airfield represented an opportunity to collect a substantial amount of mobile LiDAR data on a heavily distressed asphalt surface but without the significant vegetation observed at Selah AAF (Figure 25). The LiDAR setup was identical to that used at Selah, and data collection occurred in the same fashion whereby two passes enabled full coverage of the airfield surface (Figure 21). An example of the processed LiDAR at Grayling is shown in Figure 26.

Figure 25. Operating surfaces at Grayling AAF (2013).

Figure 26. LiDAR point cloud of asphalt section of Grayling Army Airfield.

3.4 Pope Army Airfield

Pope field represented a larger, more active runway and taxiway system while also providing a mixture of old and new concrete and worn asphalt, as well as grooving on both the concrete and asphalt pavements, to determine if the LiDAR would be sensitive to these types

of pavement features (Figure 27). The areas scanned included the main runway, sections of asphalt (Figure 28), along with grooved concrete and grooved asphalt and Taxiway Bravo. The Shark and Claw aprons included a large amount of highly distressed concrete (Figure 29).

Figure 27. Survey areas at Pope AAF (2018).

Figure 28. LiDAR point cloud of asphalt distresses at Pope AFB.

Figure 29. LiDAR point cloud of concrete distresses at Pope AFB.

3.5 Camp Rudder

The collection event at Camp Rudder provided a unique opportunity to collect roughness data on a variety of surface textures. The airfield surfaces of Camp Rudder consisted of a series of pavement types ranging from old to new concrete, to a very old sand-asphalt surface, and a recently constructed unsurfaced access road (Figure 30). The facility is used as an expeditionary training center for the Army, and only a portion of the main runway had been reconstructed, while the remaining taxiways and landing strips have remained in place since World War II.

Figure 30. Aerial view of pavement sections surveyed at Camp Rudder.

The Camp Rudder data collect also provided the first aerial LiDAR evaluation of a distressed pavement providing a normal perspective to the ground maximizing the potential resolution.

3.5.1 Telehandler LiDAR

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 40 and 50 ft above the ground. This simulated a LiDAR data perspective that could be achieved by a UAS-based LiDAR system because such a system was not ready for deployment at the time of this research. The telehandler was driven across the test site to capture the surface conditions at speeds of between 2 and 3 km/hr to maintain stability. The elevated position of the sensor enabled a truly normal perspective of the surface distresses. The frame system used for placement of the VZ-2000 is described in Figure 31, and an example of deployment on a runway surface is shown in Figure 32.

Figure 31. 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 (lower right).

Figure 32. Example of VZ-2000 mounted to an aluminum cage hoisted on a telehandler lift providing a sensor height of approximately 50 ft above ground level.

An example of LiDAR collected aurally using the telehandler is shown in Figure 33 where a transition occurs between old and new pavement along the main runway. A photograph taken from the top of the tower by a Nikon D800 36MP camera mounted to the VZ-2000 LiDAR sensor illustrates the perspective used to generate the point cloud (Figure 34). Chapter 5 documents further image sets of the various surfaces.

Figure 33. LiDAR point cloud of transition between old and new concrete on main runway at Camp Rudder.

Figure 34. Photograph taken from the telehandler tower while driving on main runway.

4 MATLAB Algorithm Development

This research sought to enable the rapid detection of pavement distresses through an analysis of unorganized three-dimensional point cloud data collected by photogrammetry and LiDAR sensors. The MATLAB programming language and Integrated Development Environment (IDE) was chosen to implement the methods described in this section. 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, and as such 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 the 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 35). 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 differentiable. This provides a much smoother surface than that produced by the nearest-neighbor method.

Figure 35. 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. 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 large volume of data collected on an airfield or roadway network. While the grid-averaging technique is beneficial for the detection of ruts and depressions on the pavement surface, this method tends to smooth the surface and mask the presence of subtle distresses such as cracks. Therefore, a grid standard deviation approach is applied in the detection of crack features.

When considering small distresses such as cracks, the key identifier in detection is the rapid change in surface elevation over a relatively level

plane where a crack penetrates the pavement. To best detect these locations, the average $\bar{z}$ elevation would negate the depth influence of a crack in a grid cell, so the standard deviation of elevation found within each cell, σ_z , is returned to the user. The standard deviation of z elevation within a cell highlights locations where large changes in z occur, locations most likely to have a crack, versus small changes in z likely where the surface is smooth and undamaged.

To enable detection of fine surface features, a user-defined grid size ($N \times N$) is constructed from the 3-D scattered data, and the grid points are input into the distress detection algorithms (Figure 36). Due to the small size of an individual crack, a grid size must be selected that yields sufficient detail to register points surrounding a crack, with a cell typically on the order of 5 cm or less across depending on the density of the point cloud.

Figure 36. Example of $N \times N$ grid applied to point cloud and grid averaging and grid standard deviation algorithms applied to cell.

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*, *txtscan*, and *bin2mat*. 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 function *txtscan* provides fast import of PLY and TXT files containing photogrammetric generated point clouds. 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.

4.2.1 Crack detection method

Cracks are detected by establishing a minimum standard deviation in which cracks are highlighted along the pavement surface. The grid standard deviation (GSD) approach in *bin2mat* creates the surface defined by the standard deviations of the z values found within each $N \times N$ grid cell (Figure 37). The user can then manually adjust the minimum standard deviation threshold to highlight the presence of desired cracks, which would establish the boundary between the red-colored pixels and the orange-colored pixels in Figure 37. This process can further be automated by selecting a default minimum threshold on which the *bin2mat* algorithm computes the standard deviation surface.

Figure 37. Surface generated using grid standard deviation method (σ_z values contoured by color).

4.2.2 Crack map

Once the user has manipulated the GSD surface to highlight all cracks of interest based on defining a minimum standard deviation, the software then creates a new text file. The text file contains a null value for each cell that has a standard deviation value less than the minimum specified and the grid-averaged z elevation in all other cells. The average z elevation is coupled with the x,y position of the centroid of the cell to define the crack located cell in a three-dimensional space. These points can then be displayed in point cloud viewing software such as CloudCompare to visualize only the surface cracking (Figure 38). From this figure, the patterns of cracks and slab joints in a concrete surface are readily identified.

Figure 38. Locations of points associated with high z standard deviation values (i.e. cracks).

4.2.3 Crack widths

Crack widths can be calculated based on the centroid to centroid x-y position from the data returned from neighboring cells. This can produce error in detecting crack widths dependent on the exact position of the crack within a cell as the cell-to-cell center can potentially be wider than the initial crack. Figure 39 illustrates how this process functions when a crack is present. The point cloud is distributed across the crack feature, but only a few points exist within each cell. The z values of each point within a cell are averaged, a standard deviation calculated, and that value placed at the centroid of each cell. In the color contour shown, smooth areas represented by a blue color would have z elevations that are all similar. Crack locations would exist in the orange and red locations where large z differentials occur. The cell-to-cell distance between the red and orange cells highlighted would represent an approximate crack width for severity measurements. An improvement to this feature would be to place a weighted x-y position based on the depth of z to draw the returned location closer to the center of the crack.

Figure 39. Methodology of grid standard deviation and grid average return method with a crack.

4.3 MATLAB functions for volumetric distress detection

The problem of measuring volumetric pavement surface distresses such as ruts and potholes is an exercise in quantifying surface depressions. As such, a method for efficiently detecting and classifying these distresses is desired. The GSD described earlier tends to miss larger distresses such as potholes or ruts as relative changes in elevation for these distresses occur as slopes rather than crevices, returning smaller standard deviation values. Therefore, a second approach to detecting depressions was considered.

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 like potholes on nearby surfaces.

4.3.1 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 algorithm is efficient enough for the present work. For the remainder of this section, the original Planchon and Darboux (2001) algorithm will be referred to as “no-sink.”

4.3.2 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 40) representing a pothole and a rut. It is imperative then that the surface has no holes in it. This presents a problem as the output of the LiDAR sensor or photogrammetry 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.

4.3.3 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.

Figure 40. No-sink algorithm flooding example.

4.3.4 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 3).

Table 3. 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 3 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 3.

There is a directional dependence on the individual scans referenced in Table 3. For instance, if a rut has more 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 41).

Figure 41. Depression-filling example on damaged pavement surface (left, point cloud; middle, 500x500 mesh; right, depression-filled mesh).

It can be seen in Figure 41 that despite pavement damage with uneven upheaval, the no-sink algorithm filled the pothole with water such that the water is allowed to “drain” out following subsequent scans (see Table 3.) Smaller depressions on the surface are also filled with water representing possible rutting or spalling.

4.3.5 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 42). It is noteworthy that in Figure 42, the surface surrounding the three depressions is 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.

The depression filling algorithm (DFA) defines the location of any grid-averaged points lying beneath the no sink pre surface. The user then generates a txtfile similar to the crack map, which returns a null value for any points outside the distress and the z-averaged elevation and x,y centroid of each cell contained within the distress. This can be displayed in a point cloud viewing software such as CloudCompare.

Figure 42. Before and after pavement potholes using a depression-filling algorithm.

4.4 MATLAB implementation of crack detection

This section describes the operation of the MATLAB crack detection software by the user. In its current configuration, the MATLAB implementation requires manual manipulation of the data by the user, but this process can be automated if the user inputs are preset at the beginning of the process.

The process begins with loading a point cloud of interest into the MATLAB GUI shown in Figure 43. This is done by first selecting a coordinate system scale (Universal Transverse Mercator (UTM) meters or feet) in the “Units” dropdown box (see Figure 43) so that the point cloud will be appropriately scaled. Following this, the user must select a grid size resolution for the pre- and post-surface meshes. This is done by inputting a value in the “Spacing” text box (see Figure 44). This resolution value represents the physical size of each grid cell within the mesh, N . Following the above two inputs, the “Load File” button can be pressed to import either LiDAR data or photogrammetry data (see Figure 44). Imported data formats are restricted to the LAS format, TXT format, and PLY format. Following import, the point data are displayed as a mesh on the right side of the software (Figure 45). This plot is updated throughout this process as the user progresses through the analysis.

Figure 43. MATLAB-developed GUI for crack and rut detection.

Figure 44. Load file portion from the GUI.

Figure 45. Mesh plot of imported post-surface from the GUI.

Once the 3-D point data is imported into the software, the user will first select the checkbox for “Switch to Contour (zDeviation)” under the Crack Detection Panel (Figure 46) and click the button “Plot zDev” to create the standard deviation values of the z elevation in each grid cell. Following these steps, a contour plot for zDeviation is shown on the right side of the GUI (Figure 47). Note that this is an initial plot displaying some, but likely not all of the scanned cracks. At this point, the user will need to adjust the minimum zDeviation by changing the value of ‘max’ until the red highlighted cracks represent the complete crack map (Figure 46).

Figure 46. Crack Detection portion of the GUI.

Figure 47. Mesh plot of zDeviation contour from the GUI.

The user will move on to the Crack Extraction Method to create the crack map. The user will provide two values for the “greater than” textbox and “less than” textbox (Figure 48). These two values determine what points from mesh computed in the previous step are extracted. The goal is to extract only those points whose color contour within the mesh denotes the presence of cracks. The plot on the right side will be updated with cracks highlighted in black on top of the average z-elevation height map after the user clicks the button “Extract Cracks” (Figure 49). The software will create a text file that contains only the black points, which is termed the crack map, by clicking the button “Create Txt for Cracks” (Figure 48).

Figure 48. Crack extraction portion of the GUI.

Figure 49. Crack map plotted on the mesh in the GUI.

4.5 MATLAB implementation of depression detection

Once the cracks within the point cloud have been identified and isolated, the user can proceed to detecting ruts, potholes, and depressions. For the sake of demonstration, a different data set taken from a section of roadway on the ERDC campus is shown to highlight the results of pothole detection.

After following the steps to load a point cloud from a LAS, TXT, or PLY file as shown in Figure 44 or continuing from the crack extraction function, the user selects 'Create Surface by Filling' (Figure 50). This function will use the depression filling algorithm on the loaded data and produce a filled contoured surface on the right side of the GUI (Figure 51). The red highlighted area represents where the z elevation is zero and is considered as the Pre Surface. The user then needs to click the button "Plot Difference Surface" to calculate the elevation difference between the Pre (zero elevation) and the Post surface (anything lying below the zero-elevation contour).

Figure 50. Rut, potholes, depression detection portion of the GUI.

Figure 51. Result from the depression filling algorithm done on the post surface.

The user now needs to extract the points that contain ruts, potholes, or depression. The user provides a value for the “greater than” textbox and the “less than” textbox (Figure 52); these two values determine what points from the mesh computed in the previous step are extracted. The goal is to extract only those points that are not red -- those below a zero elevation. The plot on the right side will be updated with ruts or potholes highlighted in dark red, after the user clicks the button “Extract Ruts” (Figure 53). The user will lastly create a text file that contains only these points by clicking button “Create Txt for Ruts” (Figure 52), similar to extraction of only the points associated with cracks. Post-processing of these points can allow for distress classification and area or volume measurements to assess severity.

Figure 52. Rut, potholes, depression extraction portion of the GUI.

Rut, Pot Holes, Depression Extraction

greater than	<input type="text" value="-0.035"/>	<input type="button" value="Extract Ruts"/>	<input type="button" value="Create Txt for Ruts"/>
less than	<input type="text" value="-0.01"/>		

Figure 53. Pothole region plotted on the mesh in the GUI.

4.6 Indexes of roughness measurement

Pavement roughness is quantified as the change in pavement surface from a theoretically uniform plane. Deviations in pavement surface roughness can negatively affect the ride quality of aircraft during landing and takeoff operations or influence vehicle speed. For highways, roughness is typically determined by use of the IRI. For aircraft, the standard for pavement roughness measurement is the BBI or the IRI.

4.6.1 Boeing Bump Index

The original Boeing Bump criteria was developed by Boeing in an effort to better quantify runway surface roughness standards (DeBord 1995). This criteria was further developed by the Federal Aviation Administration (FAA) into the BBI (Federal Aviation Administration 2009). The BBI is not a measure of roughness wavelength but rather a measure of single-event bumps.

The original Boeing criteria, from 1975 to 1994, were developed from a combination of instrumented Boeing 737 operations and numerical models of Boeing 737 aircraft operating on a testbed (DeBord 1995). Operational

data collected further refined the Boeing criteria in 1995 to only three categories, i.e., acceptable, excessive, and unacceptable (Figure 54).

Figure 54. Boeing roughness criteria from 1995 to present.

The FAA's development of the BBI was an effort to simplify the Boeing Bump criteria's use. Following FAA AC 150/5380-9, the BBI is determined by computing the ratio of measured bump height and the limit of acceptable bump height from Figure 54 for all sample points in the surface profile. The largest value for each sample point in the profile is the BBI for that point. A BBI of one or less falls into the acceptable area, and a BBI greater than one is either in the excessive or unacceptable zone.

4.6.2 International roughness index

The IRI is an index describing the roughness of a single wheel-track surface using a quarter-car model. The quarter-car model consists of a system of springs and masses approximating a single tire rolling across a surface (Figure 55).

Figure 55. International roughness index quarter-car model.

Where:

m_s = sprung mass (vehicle mass supported by a single tire),

m_u = un-sprung mass (mass of wheel and axle assembly supporting a single tire),

k_s = spring constant controlling the springing action of the vehicle suspension,

c_s = damping constant controlling the dampening action of the vehicle suspension,

k_t = spring constant controlling the springing action of the vehicle's single tire,

z_s = displacement of the sprung mass,

z_u = displacement of the un-sprung mass,

z_p = displacement of the tire from the surface profile.

Given the simplified model in Figure 55, the equations of motion for such a system can be easily determined by Newtonian (Sayers 1988), Lagrangian, or Hamiltonian means and follow in Equations 1 and 2 below.

$$m_s \ddot{z}_s + k_s(z_s - z_u) + c_s(\dot{z}_s - \dot{z}_u) = 0 \quad (1)$$

$$m_u \ddot{z}_u - k_s(z_s - z_u) - c_s(\dot{z}_s - \dot{z}_u) = k_t(z_p - z_u) \quad (2)$$

The two second-order ordinary differential equations (ODEs) above, when decomposed into four first-order ODEs, can be time integrated numerically to yield the motion of the suspension over the entire profile. The accumulated suspension motion, $\sum |z_s - z_u|$, is divided by the total length of the profile and reported as the IRI (Sayers 1988). The IRI is reported in either metric units of meters per kilometer (m/km) or imperial units of inches per mile ($in./mile$).

4.7 MATLAB implementation of roughness calculation

Determination of surface roughness is an exercise in establishing the relative change in z elevation between points along a path traced within the point cloud. Four community-developed MATLAB functions of note are required for the present work, i.e., *lasdata*, *textscan*, *bin2mat*, and *inpoly*. The *lasdata* and *textscan* functions have been described previously in the importation of point cloud data into MATLAB. The *bin2mat* function is essential in creating the mesh from which a rapid analysis of the surface geometry can be performed as described in Section 4.1. For surface roughness, the *inpoly* function is accessed to perform extraction of data from a user-defined cross section or path through the mesh.

4.7.1 Construction of cross section

The MATLAB GUI developed for the roughness calculation (Figure 56) creates a mesh from the input data based on a user-defined distance scale and displays the height map to the user. The user then interactively selects points along the surface to define a path to measure roughness. Along this path, the user will input a width of cross section used to determine a z -elevation averaging window of the data in the cross section generated by the GUI (Figure 57). The average z -elevation along the cross section is coupled to the X-Y centroid point, and the length between subsequent X-Y data points is converted to a distance R . The R-Z point positions along the selected path are output to a text file used by a third-party program, ProFAA, to compute the roughness based on IRI or BBI. ProFAA converts the data in the text file to its own format, including its preferred point spacing, R , which is used internally to calculate the Boeing Bump Index.

Figure 56. MATLAB-developed GUI for roughness calculation.

Figure 57. User selection of path and path width to determine X-Y-Z reference points.

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 (Figure 58). Then the “Load File” button can be pressed to import either LiDAR data or photogrammetry data. Imported data formats are restricted to the LAS, TXT, or PLY format. Following import, the point data are displayed as a mesh on the left side of the software (Figure 59).

Figure 58. Load file portion from the GUI.

Figure 59. Mesh plot of imported data from the GUI.

The next step is to create the box surrounding points on the runway that will be averaged together to represent an average z-elevation for roughness calculation as denoted by the red area in Figure 57. The user enters the cross-sectional width from the data plot in the text box “Width of Bound” (Figure 60). Here, a value of 0.5 ft is provided. Then the user inputs the longitudinal length by entering a value in the “Spacing.” Here, the value of 0.05 ft is provided. These two values create a box that is 6 in. wide by 0.6-in. long to represent each z-elevation along the length of the longitudinal cross section.

The user then has to define the longitudinal path along the pavement section displayed. The user selects the button “Create a CS Line” (Figure 60), and a pop-up window will ask the user to provide a name for the cross section. Using the mouse and moving a cross-hair located within the displayed data set, the user clicks a series of points outlining a unique path along the length of the pavement section (green stars in Figure 61) and right clicks to stop providing the points.

Figure 60. Parameters for cross-section portion from the GUI.

Width of Bound: 0.5

Spacing: 0.05

Create a CS Line

cs1

Figure 61. Setting up cross section and cross-section display from the GUI.

To display a plot of z-elevation versus longitudinal distance separated R units along the pavement section, the user selects the named cross section from the list box (Figure 60). The cross section will then be displayed in the plot area on the right side of software (Figure 62). MATLAB now makes a unit conversion to feet even if the user started the program in meters. The plot is always displayed in units of feet since ProFAA can only ingest data in units of feet.

Figure 62. Plot of average z-elevation versus R distance along user selected path.

4.7.2 ProFAA analysis of cross section

To enable analysis by the ProFAA software, the user must save the cross section by clicking the button “Export to txt” (Figure 63) and a dialog window will pop up asking the user to provide the name for the TXT file and press “Open.” There will be two types of files: one with just z-elevation data, and other with both z-elevation and R-distance. The file with only z-elevation data will be the input to the ProFAA converter. The user then needs to press the button “Open Convert Profile Format” (Figure 63). This will open ProFAA’s Convert Profile Format application, and here the user will convert the text file selected previously into ProFAA format (Figure 64). The user also needs to provide the “Spacing,” which was 0.05 ft as noted earlier. Lastly, the user will click the “Read File” button and select the text file with z-elevation only. A plot will be displayed, and this will confirm the completion of conversion.

Figure 63. Export and ProFAA portion from the GUI.

Export to txt

Min Z 122.7481 Max Z 123.2847

Open Convert Profile Format Open ProFAA

Boeing Bump Number

Figure 64. Convert txt file to ProFAA Format.

Convert Profile Format

Profile Format
Input File Text Output File ProFAA

Run Batch Input Sample Spacing, feet 0.05 Output Sample Spacing, feet 0.05 Elevation Scale Factor Out = In * (Scale Factor) 1000 ft High Pass ☐

Select Directory Read File Exit Finished

VERY IMPORTANT: Check the input sample spacing units, the output sample spacing units, and the elevation scale factor before using the output file.

DO NOT ASSUME that standard file formats have been read correctly. ALWAYS CHECK.

Input File Name D:\MATLAB\Boeing Bump Extract\Black_center_1.txt

Output File Name D:\MATLAB\Boeing Bump Extract\Black_center_1 - 1.pro

Plot Plot Channel No. 1 Total Channels 1

.0973 .txt- Profile 1 0.000 5.279 -0.2101 Hundreds of Feet

.0973 .Pro- Profile 1 0.000 5.279 -0.2101 Hundreds of Feet

To calculate IRI or Boeing Bump, the user selects the button “Open ProFAA” (Figure 63), and this will open the Profile and Roughness Indexes application. In this new window, the user selects “Read File” (Figure 65) and selects the new file created with extension .pro,

generated during the Convert Profile Format application. A plot of the profile will be shown on the right side of the application. To compute IRI or Boeing Bump, the user then clicks the appropriate Radio button under the Indexes column. Here, Boeing Bump is selected, which will plot the profile again along with BBI. These values can then be used to compare roughness and ride condition between surfaces as will be demonstrated in the subsequent chapter.

Figure 65. ProFAA profile and roughness indexes application.

5 Data Analysis

The intent of the MATLAB software is to provide a methodology in which cracks and volumetric distresses can be obtained from a three-dimensional point cloud. This chapter provides examples of the execution of the software on specific surface types from the varying data collections to illustrate its functionality. Future research will entail application of the software on a large scale across the entirety of the pavement surface and automation of the process. Techniques must still be developed to automatically classify and quantify distress types so that comparisons can be made to manual condition surveys and allow for calculation of pavement condition index.

5.1 Detection of crack features

Detection of crack features requires the Crack Detection and Crack Extraction tabs in the MATLAB GUI. The following section illustrates the extraction results of the software identifying crack locations occurring on various surfaces from the LiDAR scanned surfaces. The images consist of an overall LiDAR point cloud, either colorized or presented in intensity scale, followed by an overlay of extracted points from the MATLAB algorithms. These extracted points coincide very closely with the actual distresses on the surface owing to inclusion of global positioning of each extracted point from the coordinates included in the original LiDAR point cloud. The illustrations are generated in the CloudCompare software, which enables both the original LAS file to be displayed and the extracted point cloud file from the MATLAB GUI at the same time. Point pixel sizes in the extracted crack file are magnified to allow easier visualization in the report.

5.1.1 Hagler AAF

The concrete apron at Hagler AAF was the first pavement surface studied in the development of the crack detection algorithms discussed earlier. The concrete was severely distressed and was an ideal surface in which to detect cracking. Figure 66 shows a colorized LiDAR point cloud segment of the apron area and the large cracks displayed within each slab. A section of LiDAR data was chosen for demonstrating the ERDC algorithms as it represented the highest quality data set available.

Figure 66. Hagler AAF concrete apron with severe cracking and spalling.

The most recent generation of the ERDC MATLAB algorithms was applied to demonstrate the crack detection process. Figure 67 shows the application of the software to the central section of concrete depicted in Figure 66. The red lines in the image demonstrate the optimized visualization of the standard deviation of z-elevation. Large cracks are represented by thicker red boundaries, and slab joints can be seen as cracks occurring at regularly spaced intervals. Figure 68 illustrates the crack extraction process by which the black lines displayed will be written out to a crack map text file. The extracted crack map can be visualized in the lower half of Figure 69. The upper image shows the crack map overlain on top of the colorized LiDAR point cloud. This ability to overlay the detected crack points provides the user a measure of quality control on whether or not all requisite distresses were captured along with the geospatial precision of the distress locations. The xyz coordinates from the original LAS, TXT, or PLY files are retained throughout the analysis process and exported with the crack map showing close agreement to crack positions within the original point cloud. This process can be repeated for all sample units imaged at Hagler AAF to provide a complete crack map for later analysis to manual surveys conducted at the same time.

Figure 67. Detection of standard deviation of z-elevation on concrete segment at Hagler AAF using MATLAB crack detection algorithm.

Figure 68. Crack extraction performed on MATLAB crack detection algorithm.

Figure 69. LiDAR point cloud of concrete apron at Hagler AAF with overlain crack locations from MATLAB extraction (top) and isolated rut map (bottom).

5.1.2 Selah AAF

Selah Army Airfield was considered a failed runway surface owing to extensive longitudinal and transverse (L&T) and block cracking. An extraction of the crack points with the MATLAB algorithm reveals a patterned system of cracks along the entire length of the runway indicative of block cracking (Figure 70). The presence of the cracks are further enhanced by the substantial amount of vegetative growth found in and around each crack feature. The accuracy of the extraction and the associated geopositioning is illustrated in Figure 71.

Figure 70. Block and L&T cracking found on main runway at Selah AAF; extracted points overlain on colorized LiDAR.

Figure 71. Magnification of crack features overlain on colorized LiDAR at Selah AAF.

5.1.3 Grayling AAF

Grayling AAF contained a multitude of filled cracks along with the growth of new cracks occurring from those previously repaired as shown in Figure 72. Given the complex photo image of this segment of Grayling, it would appear to have a considerable amount of severe cracking. However, applying the MATLAB crack extraction algorithm (Figure 73) reveals a crack map (Figure 74) that is able to distinguish between smooth, filled cracks, and penetrative new cracks. This a capability that is very difficult to perform when using only static photo interpretations of a pavement surface. A detailed image of the crack differentiation is shown in Figure 75.

Figure 72. Colorized LiDAR segment of Grayling AAF.

Figure 73. MATLAB crack extraction algorithm for Grayling AAF segment.

Figure 74. Crack map of Grayling AAF segment (top) and overlaying LiDAR (bottom).

Figure 75. Example of crack map detail and ability to avoid filled cracks.

5.1.4 Pope Army Airfield

Pope Army Airfield consisted of a large amount of concrete surface. The apron in particular was in very poor condition with extensive cracking throughout the slabs and was slated for major rehabilitation. Crack extraction was performed on a section of the concrete apron using the intensity values obtained from the LiDAR data. The extraction technique to highlight the crack boundaries using LiDAR intensity (Figure 76) works similarly to the standard deviation of z , with the user selecting "Plot Intensity Deviation" instead of "Plot $zDev$ " in Figure 46. This procedure works well on concrete surfaces when joints are not filled with joint sealant as exhibited by the detailed crack map generated in Figure 77. Further, with good geo-positioning the crack map can align precisely with the LiDAR cloud with very few false positives (Figure 78). This figure also illustrates the detailed nature of the extraction algorithm to capture the breadth of cracks observed in a very deteriorated pavement surface.

Figure 76. MATLAB intensity crack extraction at Pope AAF Apron.

Figure 77. Colorized LiDAR (left) and extracted crack map (right) from Pope AAF Apron.

Figure 78. Pope AAF Apron crack map overlain on LiDAR and detail of extracted crack point alignment with LiDAR.

5.1.5 Camp Rudder

An example of crack extraction at Camp Rudder is demonstrated at the well-defined transition between old aged pavement and new concrete pavement shown previously in Figure 33. Figure 79 shows the ability of the standard deviation z-elevation algorithm to differentiate between the aged, cracked old surface, and the new clean concrete surface with well-defined slab boundaries. These clean crack delineations are readily observed and, when extracted (Figure 80), match well to the LiDAR point cloud shown in Figure 81. The sand asphalt and soil surfaces will be used to illustrate volumetric distress extraction in the following sections.

Figure 79. Detection of cracks using MATLAB algorithm at Camp Rudder.

Figure 80. Crack extraction using MATLAB algorithms at Camp Rudder.

Figure 81. LiDAR point cloud of main runway transition at Camp Rudder with overlain crack locations from MATLAB extraction (top) and isolated crack map (bottom).

5.2 Detection of volumetric distresses

Detection of volumetric distresses requires the Rut, Potholes, and Depressions tab in the MATALB GUI. The following section illustrates the extraction results of the software identifying ruts, depressions, and spalling occurring on various surfaces from the LiDAR scanned surfaces. The images and extraction results are presented in the same format as described earlier in the Crack Detection section.

5.2.1 Hagler AAF

The concrete apron at Hagler AAF consisted of a number of large spalls along slab joints that could be extracted as depressions. The dark red patches scattered throughout Figure 82 illustrate the depression extraction on a segment of apron taken from Figure 66. Figure 83 illustrates the volumetric distress points identified within severe spalling occurring along a crack. This data sample is taken from the upper right most crack occurring in the apron segment shown in Figure 82. While each spall location along the crack is identified, in some cases only one or two LiDAR points are available to indicate their location. This lack of point density around these spalls will make area calculations difficult when quantifying distress severity. This is a limitation of the overall digital analysis of point clouds and can be improved by the adoption of higher density LiDAR or image sensors.

Figure 82. MATLAB volumetric distress extraction on Hagler AAF apron.

Figure 83. Volumetric distress points overlain LiDAR point cloud of spalled crack at Hagler AAF apron.

5.2.2 Grayling AAF

The aged asphalt at Grayling AAF provided an opportunity to demonstrate rut detection with the MATLAB GUI. From a segment of the runway surface containing a number of filled cracks a large rut depression is detected (Figure 84). Isolated rutting occurred near areas of severe cracking in designations by the rut map displayed in Figure 85. A close examination of the rut extraction shows good alignment with the perimeter of the depressed area lying along the runway as shown in Figure 86. This type of pavement distress can only be detected with a three-dimensional representation of the surface that cannot be achieved with a traditional aerial photo reconnaissance.

Figure 84. MATLAB volumetric distress GUI at Grayling AAF.

Figure 85. Grayling AAF segment colorized LiDAR (top) and volumetric distress map (bottom).

Figure 86, Volumetric distress map overlain on LiDAR point cloud for Grayling AAF.

5.2.3 Camp Rudder

5.2.3.1 Sand asphalt section

The sand-asphalt section at Camp Rudder represented a heavily rutted and undulating surface due to severe weathering and wear. When the volumetric distress extraction tool is applied (Figure 87), the extent of the surface distress can be readily observed (Figure 88). Based solely on this rut extraction, the Soldier could note that this surface would result in a severe roughness profile limiting both aircraft and vehicle traffic. The extracted distress map is shown overlain onto the original LiDAR point cloud for validation of coordinates (Figure 89). Later, these volumetric distresses can be quantified based on size and percentage of area covered to provide a pavement condition index value.

Figure 87. Volumetric distress algorithm applied to sand-asphalt section.

Figure 88. Extraction of rut and pothole features from section of sand-asphalt (right) taken from LiDAR point cloud (left).

Figure 89. Magnified image of extracted points overlaying LiDAR for potholes and ruts.

5.2.3.2 Main runway

The main runway transition at Camp Rudder provided a comparison between smooth and rough runway surfaces. Recalling Figure 34, a portion of the aged asphalt section with severe block and alligator cracking was removed and replaced with a concrete-surfaced runway containing regular slab intervals. In Figure 90, the height map at the transition can be clearly seen with severe rutting due to alligator cracking on the bottom and no surface distress on the new concrete at the top. Figure 91 illustrates in detail the distress map filling of surface depressions at the transition boundary between asphalt and concrete surfaces.

Figure 90. MATLAB extraction of rut features on main runway at Camp Rudder.

Figure 91. LiDAR point cloud of main runway transition at Camp Rudder with overlain rut locations from MATLAB extraction (top) and isolated rut map (bottom).

5.2.3.3 Soil roadway

Camp Rudder also provided a unique opportunity to capture LiDAR on a newly constructed unsurfaced road that was in use as a vehicle transport between the closed runway group and a new runway under construction. Figure 92 is a cross section taken from the unsurfaced roadway. The highlighted active road surface segment was used for volumetric distress detection Figure 93. The extracted distress map is shown in Figure 94, and detail of the extracted points filling in small rutted areas across the

soil surface in Figure 95. An unsurfaced runway is susceptible to small depressions due to wind, water, and blast erosion, and these can be captured with the volumetric distress detection algorithm.

Figure 92. LiDAR point cloud cross section from unsurfaced road at Camp Rudder, plan view (top) and cross-section view (bottom).

Figure 93. Extracted volumetric distresses from active road surface at Camp Rudder.

Figure 94. LiDAR point cloud of active road surface at Camp Rudder with overlain rut locations from MATLAB extraction (top) and isolated rut map (bottom).

Figure 95. Close up of surface depressions in LiDAR outlined with rut-extracted points at active road surface from Camp Rudder.

5.3 Detection of surface roughness

Detection of surface roughness requires use of the separate Roughness MATLAB GUI and roughness indices using the ProFAA software package. The following section illustrates the extraction results of the software identifying changes in z-elevation occurring along longitudinal cross sections on the various surfaces. Both LiDAR and photogrammetry data are used in this section along with a comparison of advantages and disadvantages of each technology. Examples of roughness conditions are presented as changes in IRI or BBI from the ERDC-developed roughness extractions.

5.3.1 Camp Rudder

5.3.1.1 Sand asphalt roughness

The data collected at Camp Rudder identified large differences in vertical roughness over the various surface types as noted earlier in the amount of vertical distresses detected. The roughest pavement evaluated was the sand asphalt surface. The sand asphalt was a very aged (70+ yr) pavement without a strong bearing layer allowing for significant surface deformation due to settlement and trafficking. Beginning with Figure 96, a roughness profile was taken longitudinally through the runway resulting in severe changes in vertical elevation all along the surface. The inclination of the red profile line indicates a

change in runway slope over the length sampled as captured by the GPS system. When the ProFAA function is selected in the MATLAB GUI, the Boeing Bump profiles are displayed (Figure 97). This pavement resulted in a Boeing Bump value of 0.0465, signifying a number of small deviations in elevation rather than very deep depressions or those that extend over a long distance. This follows the logic behind a little-used, contingency runway strip constructed of a very flexible surface-base combination that deforms and weathers uniformly over time.

Figure 96. MATLAB analysis of sand asphalt pavement roughness.

Figure 97. ProFAA analysis of sand asphalt pavement section.

5.3.1.2 New concrete surface

Demonstrating the sensitivity of the roughness tool, the LiDAR collected over the well-maintained new concrete surface displayed a much smoother response. Figure 98 shows a longitudinal profile taken through a section of new concrete pavement, and the difference with Figure 96 can be clearly seen by a smooth, sloped line with only an occasional shallow depression. ProFAA displays a similarly smooth roughness profile and a Boeing Bump value of 0.0241, which is half that of the sand asphalt pavement (Figure 99). This suggests that LiDAR scans of the pavement surface have an ability to detect subtle changes in surface elevation that can be used to support roughness algorithms. The fact that Boeing Bump requires a data point only every 12 in. means much of the data are not being used to its optimum intent.

Figure 98. MATLAB analysis of new concrete pavement roughness.

Figure 99. ProFAA analysis of new concrete pavement section.

5.3.1.3 Unsurfaced roadway

Lastly, a roughness profile was drawn through a section of the unsurfaced road that was under construction at Camp Rudder at the time of the data collection. This shows a mixture of roughness patterns with some unevenness over the length of the selected path (Figure 100). While an unsurfaced runway is not typical of the ProFAA analysis, the

software produces a Boeing Bump index of 0.0513, which is the highest value of any of the surfaces sampled at Camp Rudder (Figure 101). This is expected given the uneven nature of compacted base material and grading on the section sampled.

Figure 100. MATLAB analysis of unsurfaced road roughness.

Figure 101. ProFAA analysis of unsurfaced road section.

These analyses demonstrate the ability of LiDAR from a 40-ft aerial platform to capture enough data to resolve subtle differences in

roughness between varying pavement surfaces. A more detailed analysis of the types of texture noted during the Camp Rudder exercise is documented in Ward (2019) using the same LiDAR data sets.

5.3.2 ERDC Cumberland Road – LiDAR versus photogrammetry

5.3.2.1 Data collection

A second roughness analysis was performed at the ERDC campus on Cumberland Road (Figure 102), which compared the ability of aerial photogrammetry to produce a point cloud of sufficient resolution to provide roughness measurements when compared to a similar LiDAR point cloud.

Figure 102. ERDC Cumberland Road test strip.

LiDAR test data on this segment of roadway was obtained using the mobile VZ-2000 mounted on the back of a truck as shown in Figure 10. Aerial photographs were obtained from the Matrix-E UAS (Figure 4)

and the Sony Alpha camera with a 35mm lens taking 24Mp still images from a flying altitude of 75 ft. Photogrammetric reconstruction took place using Agisoft Photoscan. The LiDAR and photogrammetric height maps used for roughness analysis are shown in Figure 103. There are two highlighted areas of interest including a pothole location for later magnification and a large depression that was partially filled with water at the time of the data collection.

Figure 103. LiDAR and photogrammetric height map of ERDC Cumberland Road.

LiDAR mesh

Photogrammetry mesh

The surface texture of the road can be easily distinguished in an example of colorized LiDAR shown in Figure 104. From the pothole (noted in Figure 103), the details of the surface granularity can be seen along with the abruptness of elevation change along the boundaries of the pothole. These point cloud data sets were segmented into a series of longitudinal cross sections within the MATLAB roughness program. The R-Z positions along each cross section are then loaded into the ProFAA software to develop an IRI value of each cross section.

Figure 104. Colorized LiDAR height map of pothole.

5.3.2.2 Data resolution

There are two primary differences between a LiDAR and a photogrammetric point cloud. The first is a smoothing effect that takes place at the pixel or point scale in the photogrammetric reconstruction. Figure 105 illustrates how this smoothing would influence the tire-pavement interaction. On the left, the LiDAR point cloud has no smoothing, so given a sharp drop in elevation due to a pothole or crack, the model would simulate a rougher bounce on a tire and, therefore, a higher roughness index. On the right, photogrammetric reconstruction requires averaging between pixel locations creating a smoother transition across the distress resulting in a lower roughness index. This effect can be seen in Figure 103 where there are sharper boundary changes in contour lines (colors) between the LiDAR and the photogrammetry indicating a smoother surface for the latter. Also along the center of the test section denoted by a long band of yellow contour, the LiDAR shows a number of depressions scattered throughout this region; however, the photogrammetry shows only a continuous yellow contour feature along the entire length as the smaller depressions have been averaged out.

Figure 105. Comparison of sharp LiDAR point cloud resolution (left) versus smoothed photogrammetry point cloud (right).

The second effect is the lack of a smooth texture along an otherwise smooth surface. Photogrammetry has an inherent problem in estimating elevations of flat surfaces. Due to the processing, flat or -- in this case -- smooth surfaces tend to exhibit a wavy texture when reconstructed, causing an increase in short-length roughness calculations. This can be seen in the very jagged contour lines across Figure 103 when compared to the crisp LiDAR contours.

In the roughness software, the IRI value can vary across these transitions where rapid z-elevation changes occur along the pavement. A series of longitudinal cross sections were taken across the test area and the IRI calculated as shown in Figure 106. From this plot, there is good agreement between the two point clouds; however, in most cross sections, the LiDAR data returns a higher IRI value than the photogrammetry owing to the smoothing effect discussed previously. The coefficient of determination between the two techniques is 86%, which signifies a reasonable roughness approximation for detailing vehicle behavior using either point cloud technique.

Figure 106. IRI from LiDAR and photogrammetry for various cross sections (top) and comparison of values (bottom).

5.3.2.3 LiDAR and water

While photogrammetry has its drawbacks in an increase in surface variability and a decrease in roughness due to smoothing, it does provide a detection feature in water-filled areas that LiDAR misses. LiDAR beam reflections off of water surfaces return a null value as the beam is scattered

away from the detector. This is seen in the LiDAR mesh in Figure 103 where a white area exists with no points, while in the photogrammetry mesh, a depression can be seen. Cross-sectional IRI values taken near this depression illustrate that when no data are present, no IRI can be calculated and ProFAA would not assume a depression (Figure 107).

Figure 107. IRI versus distance for LiDAR (left) and photogrammetry (right) over several cross sections near water-filled depression.

IRI vs Distance, LiDAR, CS# 7, 8, 9

IRI vs Distance, Photo, CS# 7, 8, 9

When this longitudinal cross-sectional data are plotted versus IRI (Figure 108), the value for the LiDAR cross sections is much lower due to the lack of data in the water-filled depression. Compared to Figure 106 when the influence of the water was removed from the analysis, the coefficient of determination drops to 77% when the null values from the water-filled depression are considered. These large changes in IRI value could provide a false sense of ground smoothness during a remote assessment and, therefore, care should be taken to establish moisture conditions prior to data collection.

Figure 108. IRI from LiDAR and photogrammetry for various cross sections (top) and comparison of values (bottom) with LiDAR missing values retained.

6 Conclusions and Recommendations

6.1 Conclusions

The ability to remotely assess road and airfield pavement condition is critical to dynamic basing, contingency deployment, convoy entry and sustainment, and post-attack reconnaissance. Current Army processes to evaluate surface condition are time-consuming and require Soldier presence. Recent developments in the area of photogrammetry and light detection and ranging (LiDAR) enable rapid generation of three-dimensional point cloud models of the pavement surface.

Rendering the pavement surface in three dimensions is superior for computation of volumetric distresses over two-dimensional, photo-based image analysis. These distresses can later be quantified and classified automatically to create pavement condition index values that can guide tactical decisions on vehicle maneuver. As such, georeferenced, three-dimensional point clouds were collected on a series of asphalt, concrete, and unsurfaced pavements using a mobile Riegl VZ-2000 LiDAR sensor and a UAS Matrix-E drone with a Sony A6000 24MP camera attached. This imagery was processed using ERDC-developed algorithms to identify physical surface distresses such as depressions, ruts, cracks, and overall roughness.

ERDC-developed algorithms in MATLAB can automatically discretize the pavement surface into cross- and grid-based sections for analysis of distresses according to the following.

- **Cracks** are identified based on the standard deviation of vertical elevation within small area cells across the pavement surface.
- **Ruts and depressions** are determined based on a depression filling algorithm to capture vertical elevations lying below the pavement surface.
- **Roughness** is determined based on discretization of vertical elevations along both a width of interest and a longitudinal cross section of pavement.

The ERDC software enables detected distresses and their georeferenced position to be exported to a text file containing only the distress points and their locations. This text file or distress map can be overlain on the existing point cloud for visualization and quality control. Cracks and depressions can be sized from the point-to-point distances bounding each depression to later determine severity. Surface roughness can be quantified using ProFAA software linked to the MATLAB GUI to calculate the standard International Roughness Index and Boeing Bump Index that can later relate to vehicle performance and speed.

The examples presented in this report document the quality that which the ERDC distress detection algorithms have on extracting appropriate cracks, depressions, and roughness from three-dimensional point clouds. This further demonstrates the importance of rendering the pavement in three dimensions so engineering analysis can occur. While the focus was primarily on the LiDAR point clouds, some photogrammetric point cloud data were analyzed to illustrate this data type can function as a suitable method for collecting three-dimensional data but with the drawback of smoothing and uneven surface textures.

6.2 Recommendations and future research

This report documented several examples of data analysis on varying pavement surfaces. It was noted that manual surveys on these pavement systems occurred simultaneously with the point cloud collection so that a comparison of distress detection could later be made. Future analysis should focus on extracting data from specific manual sample units for quantitative comparison between the digitized distresses identified. This will establish the accuracy of the extraction and allow refinement of the detection algorithms in instances where large discrepancies occur due to imaging quality, point density, or surface anomalies.

While photogrammetry was examined in this report, a more detailed study of its performance is required alongside companion LiDAR data collection to establish a quantitative comparison. This will guide future optical and LiDAR sensor investment and detail limitations in varying environmental conditions.

The next phase of research entails automating the ERDC algorithms with the goal of enabling Soldiers with limited training the capability to rapidly assess pavement surface condition from a remote platform. Automation should occur according to the following guidelines.

- **A default set of input values** should be established for each pavement type so that surface distresses can be detected without user interaction.
- **Classification of distress types** according to standard guidelines should be developed using some form of automated pattern recognition and convoluted neural network on the extracted distress maps.
- **Calculation of distress severity** using automated tools that calculate the width or extent of volume-based distresses based on point-to-point spacing.
- **Develop automated sample units** so that rapid comparisons between legacy manual pavement surveys can be correlated to automatic digital distress surveys.

By developing these automated features, the classification, quantity, and severity of each distress type within a given sample unit can then allow calculation of pavement condition automatically. An advanced system to calculate pavement condition can then be correlated to roughness and vice versa to establish airfield and vehicle criteria for the given surface. This will provide a superior product to current expeditionary Army pavement assessment procedures.

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.
- Bay, H., A. Ess, T. Tuytelaars, and L. Van Gool. 2008. Speeded up robust features (SURF). *Computer vision and image understanding (CVIU)* 110(3):346-359.
- DeBord, K. J. 1995. *Runway roughness measurement, quantification, and application – the Boeing method*. Document #D6-81746. Seattle, WA: Boeing Commercial Airplane Group Airport Technology Organization.
- Delaunay, B. 1934. Bulletin de l'Académie des Sciences de l'URSS. *Classe des Sciences Mathématiques et Naturelles* 6:793-800.
- Federal Aviation Administration (FAA). 2009. *Guidelines and procedures for measuring airport pavement roughness*. Advisory circular AC 150/5380-9. Washington, DC: U.S. Department of Transportation, FAA.
- Fischler, M. A., and R. C. Bolles. 1980. Random sampling consensus: a paradigm model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24:381-395.
- Furukawa, Y., and C. Hernandez. 2015. *Multi-view stereo: a tutorial*. Foundations and Trends in Computer Graphics and Vision 9(1-2):1-148.
- Huang, C., and J. M. Bradford. 1990. Portable laser scanner for measuring soil surface roughness. *Soil Science Society of America Journal* 54:1402-1406.
- Leutenegger, S., M. Chli, and R. Siegwart. 2011. BRISK: Binary robust invariant scalable keypoints. In *Proceedings of the IEEE International Conference, ICCV, 6-13 November, Barcelona, Spain*.
- Lowe, D. G. 1999. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision, 20-25 September, Corfu, Greece*, 2:1150-1157.
- Lynch, L. N., J. M. Jackson, S. K. Fairley, J. L. Williamson, J. C. Ray, T. R. Stanton, T. C. Falls, B. T. Webb, J. L. Crockett, and J. F. Powell. 2005. *Installation and operation of the Automated Route Reconnaissance Kit (ARRK)*. Vicksburg, MS: U.S. Army Engineer Research and Development Center.
- 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.
- Rosten, E., and T. Drummond. 2005. Fusing points and lines for high performance tracking. In *Proceedings of the IEEE International Conference on Computer Vision, 17-21 October, Beijing*, 2:1508-1511.

- Sayers, M. W. 1988. *Dynamic terrain inputs to predict structural integrity of ground vehicles*. Wright Patterson Air Force Base, Report NO. UMTRI 88-16. Ann Arbor, MI: University of Michigan Transportation Research Institute.
- Shahin, M. Y., Darter, M. I., and Kohn, S. D. 1977 *Development of a pavement maintenance management system*. Vol. I-V. Tyndall AFB, FL: U.S. Air Force Engineering Services Center (AFESC).
- Shahin, M. Y., and D. K. Starr. 1978. *Development of a pavement condition rating procedure for roads, streets, and parking lots*. Interim Report M-232. Champaign, IL: U.S. Army Construction Engineering Research Laboratory.
- Spangler, E. B., and W. J. Kelly. 1966. *GMC road profilometer – A method for measuring road profile*. Highway Research Record 121. Washington, DC: National Cooperative Highway Research Program.
- Triggs, B., P. McLauchlan, R. Hartley, and A. Fitzgibbon. 1999. Bundle adjustment – a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms, 21-22 September, Corfu, Greece*, 298-372.
- Ullman, S. 1979. The interpretation of structure from motion. *Proceedings of the Royal Society of London B* 203:405-426.
- U.S. Army. 2016. *Engineer reconnaissance*. Army Field Manual ATP 3-34.81. Washington, DC: U. S. Army.
- Verhoeven, G., C. Sevara, W. Karel, C. Ressler, M. Doneus, and C. Briese. 2013. New techniques for orthorectification of archaeological aerial frame imagery. Corsi C., Slapšak B., Vermeulen F. (eds), In *Good practice in archaeological diagnostics*, ed. C. Corsi, B. Slapšak, F. Vermeulen, Natural Science in Archaeology. Cham, Switzerland: Springer.
- 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.
- Wallach, H., and D. N. O'Connell. 1953. *The kinetic depth effect*. Journal of Experimental Psychology 45(4):205-217.
- Ward, A. B., J. K. Newman, and G. B. Herring IV. 2019. *Assessment of LiDAR- and photogrammetry-based airfield roughness profiling techniques*. ERDC/GSL TR-19-11. Vicksburg, MS: U.S. Army Engineer Research and Development Center.
- Westoby, M. J., J. Brasington, N. F. Glasser, M. J. Hambrey, and J. M. Reynolds. 2012. 'Structure-from-motion' photogrammetry: a low cost, effective tool for geoscience applications. *Geomorphology* 179:300-314.

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, 2509-2512. Melbourne, Australia.

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
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) April 2021			2. REPORT TYPE Final		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE Methodology for Remote Assessment of Pavement Distresses from Point Cloud Analysis					5a. CONTRACT NUMBER	
					5b. GRANT NUMBER	
					5c. PROGRAM ELEMENT NUMBER 487724	
6. AUTHOR(S) Ernest S. Berney IV, Naveen B. Ganesh, Andrew B. Ward, J. Kent Newman, and John F. Rushing					5d. PROJECT NUMBER 458278 and 480380	
					5e. TASK NUMBER 471941	
					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 Coastal and Hydraulics Laboratory U.S. Army Engineer Research and Development Center 3909 Halls Ferry Road Vicksburg, MS 39180					8. PERFORMING ORGANIZATION REPORT NUMBER ERDC TR-21-7	
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 Conducted for Office of the Assistant Secretary of the Army for Acquisition and Logistics, Project "LiDAR and Photogrammetry for Terrain Analysis"						
14. ABSTRACT The ability to remotely assess road and airfield pavement condition is critical to dynamic basing, contingency deployment, convoy entry and sustainment, and post-attack reconnaissance. Current Army processes to evaluate surface condition are time-consuming and require Soldier presence. Recent developments in the area of photogrammetry and light detection and ranging (LiDAR) enable rapid generation of three-dimensional point cloud models of the pavement surface. Point clouds were generated from data collected on a series of asphalt, concrete, and unsurfaced pavements using ground- and aerial-based sensors. ERDC-developed algorithms automatically discretize the pavement surface into cross- and grid-based sections to identify physical surface distresses such as depressions, ruts, and cracks. Depressions can be sized from the point-to-point distances bounding each depression, and surface roughness is determined based on the point heights along a given cross section. Noted distresses are exported to a distress map file containing only the distress points and their locations for later visualization and quality control along with classification and quantification. Further research and automation into point cloud analysis is ongoing with the goal of enabling Soldiers with limited training the capability to rapidly assess pavement surface condition from a remote platform.						
15. SUBJECT TERMS Remote assessment Pavement Runways (Aeronautics)—Maintenance and repair			Cracking Rutting Point clouds Military bases Air bases		LiDAR Photogrammetry Airfields Optical radar Remote sensing	
16. SECURITY CLASSIFICATION OF:				17. LIMITATION OF ABSTRACT SAR	18. NUMBER OF PAGES 118	19a. NAME OF RESPONSIBLE PERSON
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified	19b. TELEPHONE NUMBER (include area code)			