TREC Final Reports

Transportation Research and Education Center (TREC)

3-2012

Real-Time Change and Damage Detection of Landslides and Other Earth Movements Threatening Public Infrastructure

Michael J. Olsen
Oregon State University

Shawn Butcher
Oregon State University

Evon P. Silvia
Oregon State University

Recommended Citation

Olsen, Michael, J., Shawn Butcher, and Evon P. Silvia. Real-Time Change and Damage Detection of Landslides and Other Earth Movements Threatening Public Infrastructure. OTREC-RR-11-23. Portland, OR: Transportation Research and Education Center (TREC), 2012. https://doi.org/10.15760/trec.47

RESEARCH

REAL-TIME CHANGE AND DAMAGE DETECTION OF LANDSLIDES AND OTHER EARTH MOVEMENTS THREATENING PUBLIC INFRASTRUCTURE

Final Report

SR 500-500
OTREC-RR-11-23

RESEARCH ODOT/OTREC

This publication is a result of joint funding by
Oregon Department of Transportation
(ODOT) and Oregon Transportation Research
and Education Consortium (OTREC)

REAL-TIME CHANGE AND DAMAGE DETECTION OF LANDSLIDES AND OTHER EARTH MOVEMENTS THREATENING PUBLIC INFRASTRUCTURE

Final Report

SR 500-500
OTREC-RR-11-23

by

Michael J. Olsen, Shawn Butcher, and Evon P. Silvia
Oregon State University

for

Oregon Department of Transportation
Research Section
200 Hawthorne Avenue SE, Suite B-240
Salem OR 97301-5192

and

Oregon Transportation Research
and Education Consortium (OTREC)
P.O. Box 751
Portland, OR 97207

March 2012

Technical Report Documentation Page

1. Report No. OTREC-RR-11-23	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle

Real-time change and damage detection of landslides and other earth movements threatening public infrastructure

5. Report Date

March 2012-

6. Performing Organization Code

7. Author(s)

Michael J. Olsen, Shawn Butcher, Evon Silvia

. Performing Organization Report No.

SR 500-500

9. Performing Organization Name and Address

Oregon State University
220 Owen Hall
Corvallis, OR 97331

10. Work Unit No. (TRAIS)

11. Contract or Grant No.

12. Sponsoring Agency Name and Address

Oregon Department of Transportation
Research Section and
200 Hawthorne Ave. SE, Suite B-240
Salem, OR 97301-5192

Oregon Transportation Research
and Education Consortium (OTREC)
P.O. Box 751
Portland, Oregon 97207

3. Type of Report and Period Covered

Final Report

14. Sponsoring Agency Code

15. Supplementary Notes

Additional funding provided by the Oregon Department of Transportation and Oregon State University

16. Abstract

Geologic hazards such as coastal erosion, landslides, seismic loading, etc. constantly threaten public highway construction and maintenance. Repeat surveys using terrestrial laser scanning (TLS, ground-based LiDAR) enable rapid 3D data acquisition to map, see, analyze, and understand the processes generating such problems. Previously, change detection and analysis between scan surveys was conducted during post-processing upon return to the office, instead of while collecting data in the field. Change detection in the field improves the effectiveness and efficiency of the field investigation. We have developed a new algorithm that quickly geo-references scans upon field acquisition and simultaneously performs change detection by comparing these newly acquired scans to baseline models.

This algorithm has been integrated with a simple, intuitive GUI that enables change detection to be completed quickly (<1min per scan) while data are being acquired in the field. This software has also been enhanced to ensure a productive workflow.

Implementation and testing of the algorithm is underway at several sites that have been problematic for state agencies. Two test sites showing active movement are along Highway 101 in Oregon. These are the Spencer Creek Bridge site and the Johnson Creek landslide. An additional trial site is the US20 Pioneer-Eddyville highway realignment project, where several active landslides and surficial slope failures of embankment fill slopes have significantly disrupted construction efforts. Substantial baseline information at these sites was collected, which will be useful for future Oregon Department of Transportation and Oregon Department of Geology and Mineral Industries studies.

Performing change detection in the field offers several significant advantages to current post-processing workflows. First, field change detection serves as an augmented reality system, enabling field crews and researchers to see immediate results, on site, so that they are able to make key observations while present at the site, instead of being reliant on their personal memories or notes. Second, and importantly, it can improve the overall efficiency of the survey. When this information is available to the operator during field data acquisition, areas of minimal change can be quickly surveyed at coarser resolutions and areas of substantial change can be scanned at higher resolutions. This also translates into reduced processing time and data maintenance, which are currently significant hurdles for analyzing 3D laser scan datasets. Finally, this method provides immediate validation and quality control of the RTK GPS and laser scan data being collected, leading to more confidence in the acquired data and allowing any issues to be resolved directly in the field.

17. Key Words

LiDAR, laser scanning, landslides, change detection

18. Distribution Statement

Copies available from NTIS, and online at
http://www.oregon.gov/ODOT/TD/TP_RES/

19. Security Classification (of this report) Unclassified	20. Security Classification (of this page) Unclassified	21. No. of Pages 108	22. Price

Reproduction of completed page authorized ♲ Printed on recycled paper

SI* (MODERN METRIC) CONVERSION FACTORS

APPROXIMATE CONVERSIONS TO SI UNITS					APPROXIMATE CONVERSIONS FROM SI UNITS				
Symbol	When You Know	Multiply By	To Find	Symbol	Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH					**LENGTH**		
in	inches	25.4	millimeters	mm	mm	millimeters	0.039	inches	in
ft	feet	0.305	meters	m	m	meters	3.28	feet	ft
yd	yards	0.914	meters	m	m	meters	1.09	yards	yd
mi	miles	1.61	kilometers	km	km	kilometers	0.621	miles	mi
		AREA					**AREA**		
in^2	square inches	645.2	millimeters squared	mm^2	mm^2	millimeters squared	0.0016	square inches	in^2
ft^2	square feet	0.093	meters squared	m^2	m^2	meters squared	10.764	square feet	ft^2
yd^2	square yards	0.836	meters squared	m^2	m^2	meters squared	1.196	square yards	yd^2
ac	acres	0.405	hectares	ha	ha	hectares	2.47	acres	ac
mi^2	square miles	2.59	kilometers squared	km^2	km^2	kilometers squared	0.386	square miles	mi^2
		VOLUME					**VOLUME**		
fl oz	fluid ounces	29.57	milliliters	ml	ml	milliliters	0.034	fluid ounces	fl oz
gal	gallons	3.785	liters	L	L	liters	0.264	gallons	gal
ft^3	cubic feet	0.028	meters cubed	m^3	m^3	meters cubed	35.315	cubic feet	ft^3
yd^3	cubic yards	0.765	meters cubed	m^3	m^3	meters cubed	1.308	cubic yards	yd^3
NOTE: Volumes greater than 1000 L shall be shown in m^3.									
		MASS					**MASS**		
oz	ounces	28.35	grams	g	g	grams	0.035	ounces	oz
lb	pounds	0.454	kilograms	kg	kg	kilograms	2.205	pounds	lb
T	short tons (2000 lb)	0.907	megagrams	Mg	Mg	megagrams	1.102	short tons (2000 lb)	T
		TEMPERATURE (exact)					**TEMPERATURE (exact)**		
°F	Fahrenheit	(F-32)/1.8	Celsius	°C	°C	Celsius	1.8C+32	Fahrenheit	°F

*SI is the symbol for the International System of Measurement

ACKNOWLEDGEMENTS

This project was funded by the Oregon Transportation Research and Education Consortium (OTREC) under project #2011-398. The Oregon Department of Transportation (ODOT) and Oregon State University (OSU) also provided funding for this effort. Graduate students Keith Williams, Abby Chin, and Tony Rikli assisted with field data collection and processing. Computer science undergraduate students Alfred Flammang, Rebecca Pankow, and Andrew Johnson wrote code for the project. Additional undergraduate students Cierra Eby, Kris Puderbaugh, and Amanda Olson assisted with data collection and processing. We thank Matthew Mabey (ODOT), Ron Singh (ODOT), Jonathan Allan (DOGAMI), Curran Mohney (ODOT), and Alan Blair (FHWA) for serving on the technical advisory committee and for their valuable insights. We appreciate George Priest and Jonathan Allan (DOGAMI) for providing us with a geologic tour of the Johnson Creek Landslide.

DISCLAIMER

This document is disseminated under the sponsorship of the Oregon Department of Transportation, the United States (U.S.) Department of Transportation, and the U.S. Department of Transportation University Transportation Centers Program in the interest of information exchange. The State of Oregon and the U.S. Government assume no liability of its contents or use thereof.

The contents of this report reflect the view of the authors who are solely responsible for the facts and accuracy of the material presented. The contents do not necessarily reflect the official views of the Oregon Department of Transportation, U.S. Department of Transportation, or the U.S. Department of Transportation University Transportation Centers Program.

The State of Oregon and the U.S. Government do not endorse products of manufacturers. Trademarks or manufacturers' names appear herein only because they are considered essential to the object of this document.

This report does not constitute a standard, specification, or regulation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

EXECUTIVE SUMMARY

Geologic hazards such as coastal erosion, landslides, and seismic loading constantly threaten public highway construction and maintenance. Repeat surveys using terrestrial laser scanning (TLS, ground-based Light Detection and Ranging (LiDAR)) enable rapid, 3D data acquisition to map, see, analyze, and understand the processes generating such problems. Previously, change detection and analysis between scan surveys was conducted during post-processing upon return to the office, instead of while collecting data in the field. Change detection in the field improves the effectiveness and efficiency of the field investigation. We have developed a new algorithm that quickly geo-references scans upon field acquisition and simultaneously performs change detection by comparing these newly acquired scans to baseline models.

This algorithm has been integrated with a simple, intuitive graphical user interface (GUI) that enables change detection to be completed quickly (<1min per scan) while data are being acquired in the field. This software has also been enhanced to ensure a productive workflow.

Implementation and testing of the algorithm is underway at several sites that have been problematic for state agencies. Two test sites showing active movement are along Highway 101 in Oregon. These are the Spencer Creek Bridge site and the Johnson Creek landslide. An additional trial site is the US20 Pioneer-Eddyville highway realignment project, where several active landslides and surficial slope failures of embankment fill slopes have significantly disrupted construction efforts. Substantial baseline information at these sites was collected, which will be useful for future studies by the Oregon Department of Transportation (ODOT) and the Oregon Department of Geology and Mineral Industries (DOGAMI).

Performing change detection in the field offers several significant advantages to current post-processing workflows. First, field change detection serves as an augmented reality system, enabling field crews and researchers to see immediate results, on site, so that they are able to make key observations while present at the site, instead of being reliant on their personal memories or notes. Second, and importantly, it can improve the overall efficiency of the survey. When this information is available to the operator during field data acquisition, areas of minimal change can be quickly surveyed at coarser resolutions and areas of substantial change can be scanned at higher resolutions. This also translates into reduced processing time and data maintenance, which are currently significant hurdles for analyzing 3D laser scan datasets. Finally, this method provides immediate validation and quality control of the real-time kinematic RTK GPS and laser scan data being collected, leading to more confidence in the acquired data and allowing any issues to be resolved directly in the field.

1.0 INTRODUCTION

1.1 PROJECT OVERVIEW

This research focuses on improving and developing new uses for 3D laser scanning in highway construction, maintenance, and monitoring. This is important in helping DOTs detect problematic areas and hazards before they occur, thereby improving public safety and efficiency of resource use. For example, steep rock cuts and active landslides can be very problematic for many of Oregon's highway corridors, particularly in the coast range. Closure or restricted traffic flow through highways can have significant economic penalties, particularly for communities that have limited routes to link them with other communities. Further, many of these corridors serve as lifeline routes to the coast of Oregon from the Willamette Valley. Hence, it is crucial that these critical routes remain open to evacuate people from the coast and transport supplies in emergency and disaster situations. Many landslides show some form of movement prior to catastrophic failure. Discoveries by repeat monitoring can provide agencies with critical information on the spatial and temporal variability of the movement, enabling the DOTs to determine the best solution before a hazard occurs instead of being forced to act quickly under emergency conditions.

Three-dimensional laser scanning is a relatively new, versatile technology, and, as such, many applications of Terrestrial Laser Scanning (TLS) have not been fully developed. This powerful technology has been shown to be an efficient method to obtain a substantial amount of geospatial data rapidly. However, because of hardware and software costs, a steep learning curve, technical skills, and other factors, TLS has not been universally adopted in the transportation sector. ODOT and some other government agencies currently use 3D scanning on some projects. However, such use is limited by the resources and time required for data processing and interpretation. ODOT has expressed a need for easier implementation and expanded use of this technology.

Further, this project studies the stability of sea cliffs along portions of Highway 101 and the impact of erosion on public infrastructure and coastal development. Recent media articles (e.g., Tobias 2010) discuss the destructive impacts of the ocean on several beach towns that provide revenue to the local and state economy through tourism expenditures. These coastal communities are constantly threatened by storms, sea level rise, and tsunami hazards. The real-time change analysis tool can be used to evaluate the effectiveness of remediation and preventive measures in situ. Determining sea cliff recession rates will guide the creation of appropriate development setback laws and help prioritize remediation efforts for the protection of roadways and other public infrastructure.

Another current emphasis at ODOT is to transition from traditional 2D paper-based design to 3D digital design (*Singh 2008*). Laser scanning technology is beneficial to facilitate the engineering community's move toward new, advanced 3D analysis and design techniques. Full 3D highway design models can provide substantial time and cost savings and be a spatial and temporal database for the life-cycle of a road. Increased use of laser scanning technology can assist ODOT in implementing 3D laser scanning for construction automation and machine control purposes by creating a system for analysis of real-time information from the scanner.

1.2 OBJECTIVES OF STUDY

This project focuses on both technological and scientific advances to improve the efficiency and effectiveness of highway monitoring. The following important objectives were accomplished:

- **Obtain critical time-series 3D data.** We have collected a substantial amount of time-series data covering sections of troubled public infrastructure. This data (and future data collected at these sites) will be valuable for geotechnical modeling and analysis of landslides and other types of mass wasting to shed light on possible mitigation techniques. Chapter 3 discusses the data collected and observations from the data.

- **Develop a new real-time change detection algorithm**. A newly developed real time change detection algorithm was programmed. The algorithm was designed to be small, fast and work with a variety of laser scan datasets. This open source program will be provided to ODOT Geometronics and others for use in monitoring hazardous sections of highways to detect any movement or to indicate that no movement has taken place since the previous visit. The algorithm is discussed in Chapter 2. Chapter 4 discusses the testing of the algorithm.

- **Increase understanding of geologic processes.** High-resolution laser scan data now allows us to see and understand processes at resolutions which escape traditional observation and measurement. This improves understanding of erosion and other processes leading to slope instability. Chapter 3 discusses analyses performed using the data at the test locations.

- **Improve efficiency and quality of 3D surveying.** The development of this real-time system will make field investigations more efficient by allowing users to determine when and where higher resolutions are needed. Also, the change analysis itself acts as a quality control for field data collection because any data misalignments will show up as changes when compared to the baseline model. Chapter 2 discusses optimizations implemented in the algorithm to improve efficiency.

- **Provide critical information to ODOT to improve allocation of resources.** This study allows for verification and quantification of movements that are occurring at the studied locations. Such information will allow DOT's to prioritize monitoring and rehabilitation efforts. Chapter 5 discusses an implementation plan for use of the algorithm and data in ODOT's operations.

- **Expose additional units in ODOT and other state agencies to 3D technologies.** Because of the interdisciplinary nature of this project, it will be useful to visually show divisions such as geotechnical, construction, operations, etc. the additional information available through the use of 3D laser scanning. Chapter 5 discusses strategies for assisting ODOT with pursuing for this objective.

1.3 LITERATURE REVIEW

1.3.1 Terrestrial Laser Scanning (TLS)

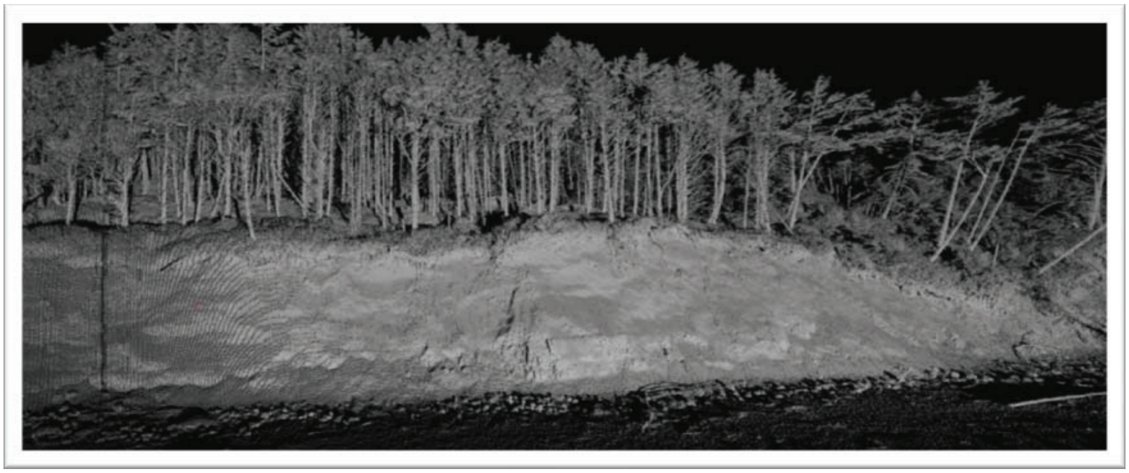


Figure 1.1: 3D laser scan survey point cloud of the Johnson Creek landslide (June 2011).

Terrestrial laser scanners (TLS) use Light Detection and Ranging (LiDAR) technology to produce highly detailed 3D models (Figure 1.1) of the terrain surface. However, their performance is highly dependent on the intended application and size of target to scan. These applications vary from forest management to monitoring of tunnels and other critical infrastructure (*Vosselman and Maas 2010*). Current time-of-flight scanners acquire data at rates of tens to hundreds of thousands of points per second with accuracies of 3 to 20 millimeters. Many scanners have a maximum effective range of 100 to 300 meters, whereas some long-range scanners are capable of measuring terrain surfaces at distances up to 2-3 kilometers. In addition to terrain mapping, most scanners concurrently photograph the scene, assigning RGB color values to each scan point. Also, an intensity value (i.e., the strength of the signal degradation) is measured, providing additional information about the type of reflecting material (e.g., geology, vegetation cover etc.). The intensity value is also influenced by scanning geometry (e.g., obliquity of laser pulse and range to target). Because LiDAR is a line-of-sight technology, multiple scans must be merged together to form a complete 3D model. More recently, mobile laser scanning systems have become available that are capable of both stop-and-go and continuous scanning (*Barber 2008, Glennie 2009*). These systems enable efficient coverage of large geographic areas.

The resulting data of terrestrial systems is similar to that acquired by airborne systems. However, the key difference between these two approaches is that the TLS scanner is mounted to a tripod on the ground, providing an improved view of vertical features, whereas airborne laser scanners (ALS) are mounted in a plane and look down, providing a better perspective on horizontal features. Young et al. (*2010*) provide a comparison of airborne and terrestrial laser scanning for sea cliff erosion studies. For site investigations, generally, TLS will be more economical.

Three-dimensional laser scanning has become increasingly effective for geotechnical and geologic analysis. Kayen et al. (*2010*) provide an overview discussion of a wide variety of geotechnical applications of 3D laser scanning including post-earthquake deformations, landslide analysis, liquefaction settlements, and trench volume calculations. For example, Kayen et al. (*2006*) used TLS to model both ground and structural deformations following the 2004 Niigata Earthquake, showing its usefulness in quickly ascertaining post hazard damage. Previous research by DOGAMI and ODOT (*Priest et al. 2008*) implemented scanning to study seaward movements of the Johnson Creek landslide, which is also subject to significant toe erosion.

TLS has recently proven to be an efficient tool for sea cliff erosion studies. For example, Collins and Sitar (*2004*) and Young and Ashford (*2007*) used TLS to study bluff retreat in weakly cemented sands. Lim et al. (*2005*) and Rosser et al. (*200*5) analyzed recession of rocky sea cliffs using TLS. Olsen et al. (*2008*) discussed a monitoring program established along the San Diego County coastline, where repeat scans were performed as part of a rapid response effort to quickly acquire scan data following a sea cliff collapse.

TLS has also been used to undertake detailed geological assessments of several landslides, enabling improved understanding of the processes and mechanisms contributing to landslide movement (e.g. *Jaboyedoff 2010*). For example, Collins et al. (*2007*) used TLS to study failure mechanisms of the coastal Northridge Bluff landslide in California demonstrating significant movement of the slide in 2003 and 2007. Jaboyedoff et al. (*2009*) used laser scanning to study retrogressive landslides in sensitive clays and rotational landslides occurring along river banks. More recently, Dunning et al. (*2010*) combined TLS surveying of landslides with numerical modeling of the landslide. Collins and Sitar (*2011*) analyzed the stability of steep slopes composed of cemented sand using detailed profiles derived from terrestrial laser scanning. Considerable work has also been undertaken in recent years to document the patterns of landslides and mechanisms for failure. For example, Burns and Madin (*2010*) demonstrated a methodology to use airborne LiDAR to map landslides in northwest Oregon, ultimately creating landslide hazard maps that could be used by local government for planning purposes. Similarly, Schulz (*2005*) discussed landslide susceptibility estimation from airborne LiDAR data.

TLS has been used for several transportation related slope assessments. Kemeny and Turner (*2008*) evaluated the use of laser scanning for highway rock slope stability analyses and found that ground-based LiDAR offered several advantages compared to traditional techniques including safety, accuracy, access, and analysis speed. Kemeny et al. (*2008*) used LiDAR to evaluate several rockfall sites near highways in Utah and Colorado. Turner et al. (*2006*) discuss processing procedures to use TLS to evaluate the stability of rocky slopes.

1.3.2 Change Detection Using LiDAR

Change detection from LiDAR data sets is a topic of great interest in the research and user communities. LiDAR provides data at an exceptional resolution, which coupled with rapid and growing acquisition capabilities, allows for unprecedented analysis among research and industry user groups. Industry experts predict that advancing capabilities of LiDAR systems will enable the development of real-time change detection and the integration of multiple sensors (e.g. hyperspectral) in the next five years (*Samarasekera 2011*).

1.3.2.1 Airborne LiDAR

Large-scale LiDAR change detection has been proposed as a method for conducting building inventory assessments in dense urban areas (*Vu et al. 2004*). By collecting baseline models of a region of interest into a database, damage assessment can be quickly implemented following a natural disaster. This, in turn, could provide valuable information in the critical time window for search and rescue efforts.

The methodology proposed by Vu et al. (*2004*) uses grid averaging with nearest neighbor interpolation to preserve sharp building edges to create these baseline models rather than point clouds or triangulated models. Change is detected via the creation of a histogram difference grid and comparing new values to the baseline model. The mean generally is indicated as the bias between two datasets. Hence, new values greater than the mean plus a multiple of the standard deviation indicate construction regions, while values less than the mean minus a multiple of the standard deviation denote regions of demolition or destruction. The process is semi-automated with manual interaction to resolve conflicts and quality control.

To improve the accuracy of airborne LiDAR change detection, Trinder and Salah (*2011*) evaluated several methods of analysis, including:

- **Principle Components Analysis (PCA) -** graph pixels along an axis of correlation and along a perpendicular axis of differences.

- **Support Vector Machine (SVM) -** implements a post-classification system to form data into object classes (e.g., buildings and roads) from statistical analysis.

- **Image Differencing -** subtracts the recent scan from the baseline model.

- **Fusion Method -** uses a Simple Majority Vote (SMV), determined from the combined votes of the three previous change detection methods. The fusion method compares the results from all three change analysis algorithms to look for agreement or dissimilarity between methods. If all three methods return different results, then the method of highest accuracy gets the most consideration. The accuracy of this method is dependent on the size of the region analyzed, optimum for regions ≥50 m^2.

To fulfill the need for up-to-date spatial data for battlefield environments, Butkiewicz et al. (*2008*) created a real-time change detection framework to analyze airborne LiDAR data from multiple airborne systems. The framework is distinct from the data collection process (e.g., it does consider geo-referencing the data); instead it collects and merges multiple input sources. This necessitates that the input data be previously geo-referenced into a common coordinate system for distribution to a scalable number of change detection units. Points are compared on an individual basis to the baseline model. A quality control factor is used to account for source precision and base-model density. Points that fall outside the quality-control enhanced model are marked as change. An additional procedure tracks the change status of sub-regions within the model for priority to determine if active change is occurring.

1.3.2.2 Terrestrial Laser Scanning

Change detection using terrestrial laser scanning centers on monitoring activities such as landslides and rock faces. However, note that these techniques have been developed as post-processing analyses and are separate from the data collection procedure.

One method of analyzing landslide data generates displacement fields for the face of the slide (*Teza et al. 2006*), requiring generation of a baseline model and additional scan data acquired from a single scan position. Using a piecewise alignment method (PAM), the new data are subdivided into side-length regions, typically several meters. These points are then individually aligned to the base model using an Iterative Closest Point (ICP) algorithm (*Besl and McKay 1992*). Displacement vectors are computed using these points for each sub-area. This procedure requires the use of several different commercial programs, although the scripts to run it on these packages are provided freely by the authors.

Additional steps necessary to improve fine-scale change detection include evaluation of geo-referencing methods and vegetation filtering. Alba et al. (*2009*) performed laser scanning to evaluate change on rock faces. Geo-referencing was accomplished through several methods including direct geo-referencing by mounting scanners on a static control pillar, aligning scans by using multiple ground control points to mount prism targets on, and aligning point clouds using the ICP algorithm. Both octree (division of the data into a set of hierarchical cubes of progressively decreasing size) and Iterative filtering were used to eliminate vegetation covering the cliffs. Use of the octree filter results in vegetation removal but requires breaking the point cloud into large cubes, which decreases the resolution of the data set. The Iterative filter maintains the point cloud resolution, but requires manual inspection of the cloud to ensure data clipping has not occurred. The purpose of these procedures is to generate the highest accuracy representation of the ground surface in both the baseline and new scan models so that a deformation map can be generated and analyzed.

Alba and Scaioni (*2010*) expanded this method to replace the RGB image data obtained from an attached digital camera with a near infrared (NIR) camera. An automated algorithm was created to filter new scan data based upon NIR data spectrum analysis. The data are subsequently filtered into multiple 2.5D regions, and the point cloud sampled to generate a square-grid lattice digital elevation model (DEM). The change detection algorithm focuses on searching for volumetric changes along the DEM surface, primarily cavities in the rock face from detached boulders. The result is a matrix containing discovered holes in the DEM, which is refined and analyzed by deformation analysis. This method is capable of very small change detection on the order of cubic centimeters, but was only tested on real data comprising only a small section of approximately 32 square meters.

Girardeau-Montaut et al. (*2005*) note the critical nature of accurately aligning the scan, and they emphasize that ICP methods require scans be very similar. They also discuss that change detection between epochs needs to be accomplished quickly and efficiently under time constraints. The basis for their change detection algorithm relies on the use of octree-based comparisons. Three separate methods are evaluated: average distance between points, best fitting plane orientation, and the Hausdorff distance. The first two methods are reported as fast, but with low accuracy. The last method is the most precise, as it still relies on octree filtering, but allows the nearest neighbor to lie in a cube not homologous to the original scans cube. All three methods make use of direct point-to-point comparisons to calculate change between scans. Suggested future developments specifically acknowledge development of automation in the process intended to be usable by operators less knowledgeable about point cloud manipulation.

1.3.2.3 Other Systems

Change detection is also being applied to update GIS databases for applications such as land-use demarcation. Pixel data taken from airborne cameras are classified into objects comprising groups of pixels (forest, settlement, etc.) based upon a supervised maximum likelihood method. These classified objects are then compared against the GIS database to determine if change has occurred. LiDAR data can be incorporated into the classification model to improve object separation based upon attributes such as height and slope. However, this process is intended for large-scale, land-use projects and is unsuitable for small-scale change detection. The nature of the pixel grouping algorithm averages out small-scale differences within the polygon boundary of the object class (*Walter 2004*).

1.3.3 Need for a New Terrestrial Algorithm

Previous work has developed important tools and algorithms for change detection; however these tools have been limited in their application to real-time change using terrestrial laser scanning. Particularly since TLS scans are geo-referenced by a scan position setup rather than dynamically for each individual point as is done with airborne systems. All of these methods share the requirement for generating a baseline data model via data post-processing steps against which to compare newer data. Most of these methods require that new scan data acquired in the field be taken back to the lab for additional post-processing followed by the application of the respective change detection methodology. Hence, there is no opportunity to take advantage of the new data until after an operator has left the site. Several improvements are required to overcome these hurdles:

1. **Increased accuracy of geo-referencing.** The methodology needs to geo-reference terrestrial laser scan data on the fly. Mobile platforms (such as airborne LiDAR) obtain orientation information through the flight path and an Inertial Motion Unit (IMU). However, terrestrial scanners remain fixed during a scan. Past change detection workflows using TLS either 1) do not geo-reference the data and perform all scans during a short time period from

one setup, or 2) perform the change analysis during post-processing, after the geo-referencing has been completed through a separate workflow.

2. **Increased speed.** For the results to be usable in the field, the algorithm must be able to sort through a substantial amount of data quickly to calculate change.

3. **Improved interface to scanning system.** Access to the scan data through custom interfaces is critical to saving field time and training.

4. **Simplified processing.** The workflow should be simple and easy to follow since field crews have substantial constraints. All processing would ideally be accomplished through use of a single software program.

5. **Public program.** This results in increased use and cost savings for DOTs and other organizations.

2.0 METHODOLOGY

The workflow for the developed **Li**DAR **Ch**ange **En**gine (*Lichen*) is presented in Figure 2.1 and comprises two distinct phases. Phase one consists of creating the baseline model, which follows a typical LiDAR post-processing workflow and would be completed earlier. However, in the case of on-site monitoring, the baseline surface can be created in the field from the first scan acquired on the site. The second phase is specific to *Lichen*, comprising the real-time analysis tools. In essence, this process creates a bridge between baseline and repeat scans thus enabling data usability in the field, which has not previously been attainable.

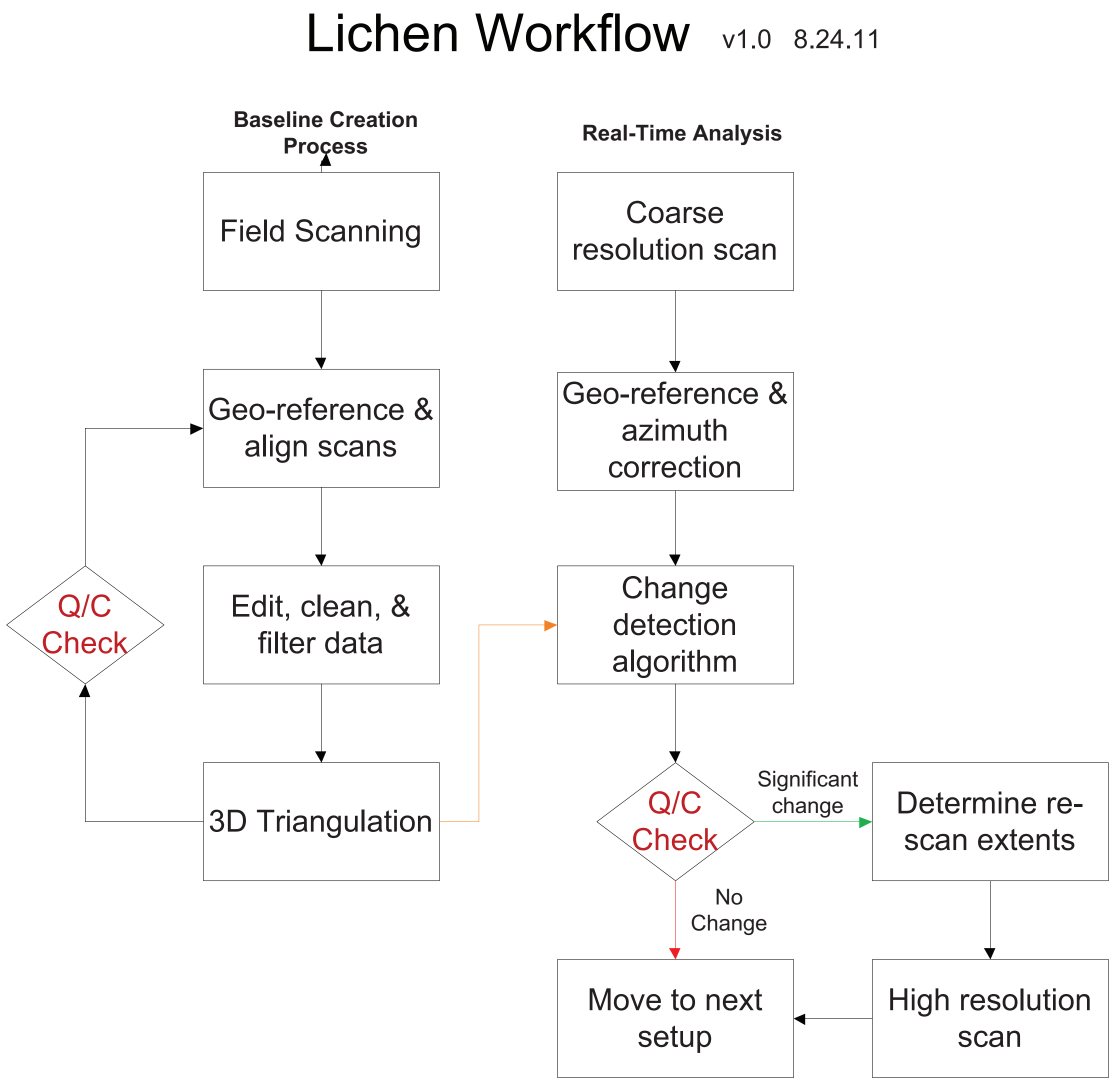

Lichen Workflow v1.0 8.24.11
Baseline Creation Process
Real-Time Analysis
Field Scanning
Geo-reference & align scans
Edit, clean, & filter data
3D Triangulation
Q/C Check
Coarse resolution scan
Geo-reference & azimuth correction
Change detection algorithm
Q/C Check
Significant change
Determine re-scan extents
No Change
Move to next setup
High resolution scan

Figure 2.1: Implemented methodology used for real-time analysis with *Lichen*.

2.1 BASELINE SCAN CREATION

Creating the baseline scan is a four-step process encompassing traditional LiDAR post-processing. The first step in the process is collecting scan data from sites of interest, usually from multiple scan position setups. The scan data are then geo-referenced into a common coordinate system (e.g., UTM, State Plane, OCRS, etc.) and aligned for best fit. The data may then be processed to remove noise or unwanted features, as well as filtered for efficiency or rapid cleaning. What remains is a pruned point cloud containing only the region of interest. This is then fed into one or more triangulation algorithms to generate a 3D triangulated surface.

2.1.1 Field Scanning

Oregon State University (OSU) field researchers have traveled to Beverly Beach, the Johnson Creek Landslide, and the US 20: Pioneer Mountain to Eddyville realignment project (see Chapter 3) throughout the duration of this project to collect high resolution LiDAR data. This data serves both as the baseline data for testing the real-time change software, as example data for future users, and as key temporal data for future analysis by DOGAMI, ODOT, OSU, and other state agencies. Scan data was collected using a Riegl VZ400 terrestrial laser scanner with a calibrated Nikon D700 digital SLR camera to record RGB values (the instrument specifications are listed in Table 2.1). The nominal measurement accuracy of this scanner is 5 mm. Note that real-world performance often varies depending on surface material and scanning geometry. The scanner was operated in high-speed mode (up to 125,000 points per second) and its internal inclination sensor was utilized to extract leveling information (roll and pitch) for each scan position. The scanner was either mounted upon a tripod (see Figure 2.2) or a metal plate as part of a mobile cart system. Scanner positional data was collected using a Trimble R8 GNSS receiver through RTK GPS (using the ORGN network) or Fast Static GPS (processed through OPUS). Scans were generally spaced roughly 50 meters apart and collected for long (upwards 1 km) segments. Chapter 3 provides more details on the results of the data acquisition.

Table 2.1: Technical specifications for the Riegl VZ-400 scanner (from www.riegl.com)

Parameter	Long Range Mode	High Speed Mode
Laser Pulse Repetition Rate	100 kHz	300 kHz
Effective Measurement Rate	42,000 meas./second	125,000 meas./second
Max Measurement Range (natural targets, ρ ≥ 90%, highly reflective)	600 m	300 m
(natural targets ρ ≥ 20%, less reflective)	280 m	160 m
Minimum Range	1.5 m	1.5 m
Accuracy (1σ)	5 mm	5 mm
Precision (1σ)	3 mm	3 mm
Angular Measurement resolution	Better 0.0005^{o} (1.8 arcsec)	Better 0.0005^{o} (1.8 arcsec)
Beam divergence	0.3 mrad	0.3 mrad

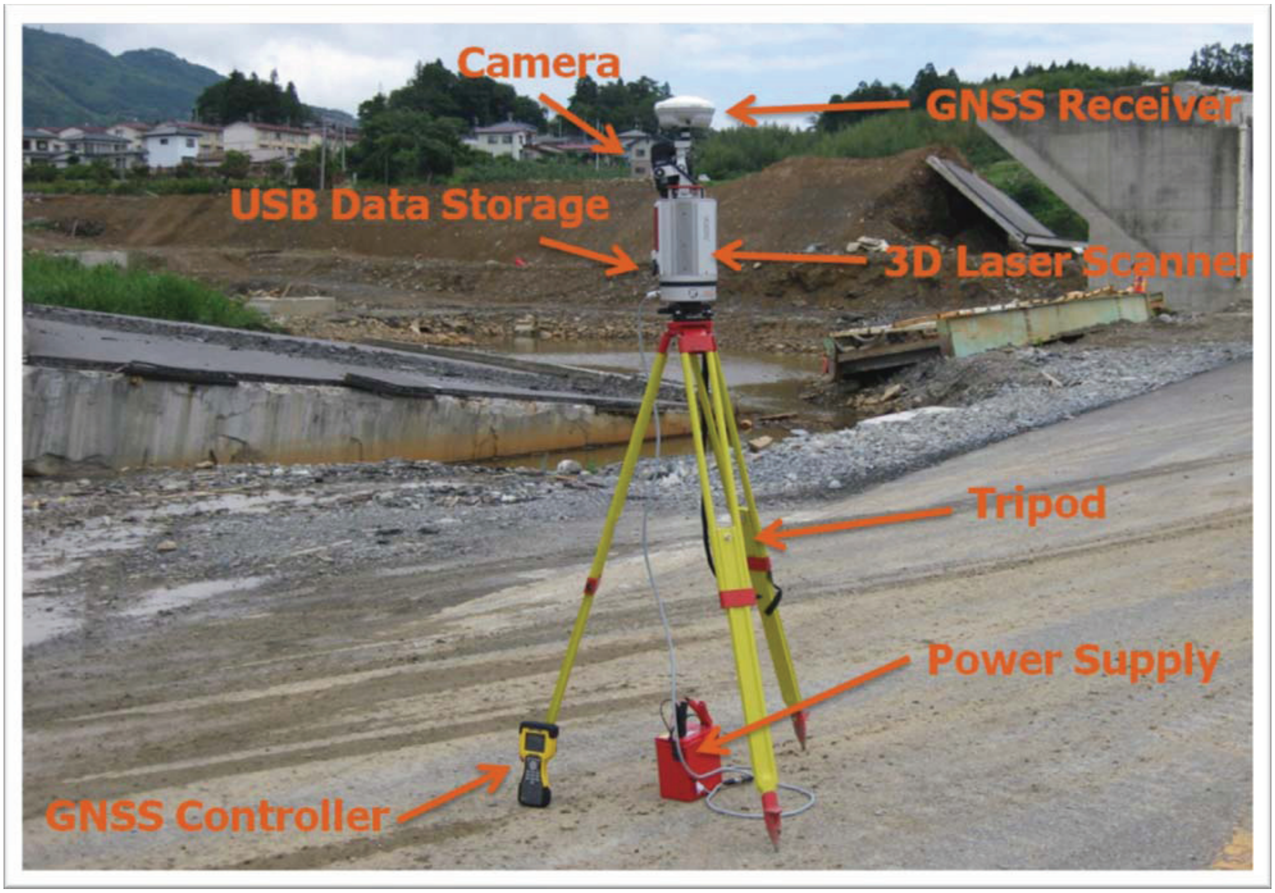


Figure 2.2: Tripod-mounted TLS system with labeled components.

2.1.2 Geo-referencing

Geo-referencing is a crucial step in aligning multiple scan positions and in relating data from different epochs. Since each scan is normally recorded in its own scanner coordinate system, one scan is not relatable to another unless it is adjusted into the same coordinate system. This is accomplished through a least squares transformation (e.g. *Ghilani 2010*), applying both rotations and translations along the *X, Y,* and *Z* axes. Additional parameters such as scale, shearing, etc. can be included in some coordinate transformations, but are not generally applied to laser scan data. Some scanners apply radial scaling through temperature, pressure, and relative humidity corrections. Both the real-time change algorithm and the baseline scan processing were geo-referenced using the methodology of Olsen et al. (*2009, 2011*) to ensure accurate point cloud alignments. The methodology was developed for efficient surveying in dynamic environments where survey control can be difficult to establish. This section will discuss scan transformations using this workflow. Note that some of the discussion is relevant to any coordinate system transformation. For detailed assessment of the validity of the geo-referencing, see Olsen et al. (*2009, 2011*). This discussion is meant primarily for background purposes.

When the scanner remains fixed (e.g., some site field tests in Chapter 3) for continuous monitoring during a short time period, no adjustments are necessary and all scans can remain in the scanner coordinate system for change analysis.

2.1.2.1 Rotation Matrix

Given a right-handed scanner coordinate system (Figure 2.3) with the *X* axis pointing along the scanner's initial line of sight, the *Y* axis at 90° counter-clockwise from the *X* axis in the horizontal plane, and the *Z* axis pointing vertically upwards, rotation of the scan setup into a project coordinate system (*X'*, *Y'*, *Z'*) is usually described in terms of roll (α), pitch (β), and yaw (γ) as rotations around the *X*, *Y*, and *Z* axes, respectively.

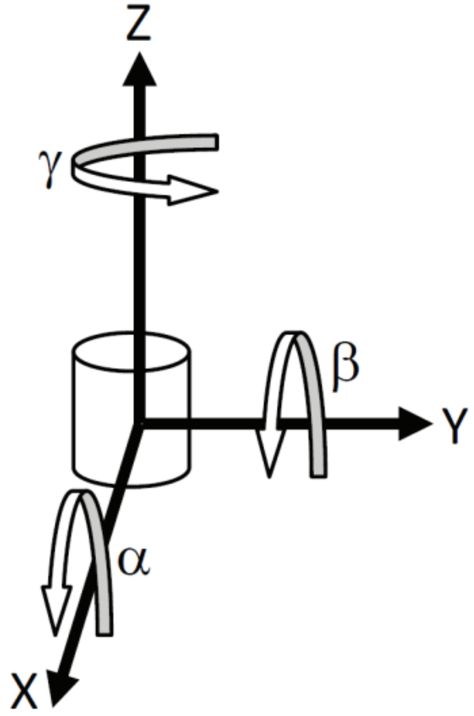


Figure 2.3: Laser scanner transformation parameters.

Given a laser scan data point, ***P₀***, with coordinates (*X*, *Y*, *Z*), alignment adjustments can be applied by rotating the point around the *X*, *Y*, and *Z* axes centered at the scan origin:

$$P' = R_z R_y R_x P_0 \tag{1}$$

$$P' = R(\alpha,\beta,\gamma)P_0 \tag{2}$$

$$R(\alpha,\beta,\gamma) = \begin{bmatrix} \cos\beta\cos\gamma & -\cos\alpha\sin\gamma + \sin\alpha\sin\beta\cos\gamma & \sin\alpha\sin\gamma + \cos\alpha\sin\beta\cos\gamma \\ \cos\beta\sin\gamma & \cos\alpha\cos\gamma + \sin\alpha\sin\beta\sin\gamma & -\sin\alpha\cos\gamma + \cos\alpha\sin\beta\sin\gamma \\ -\sin\beta & \sin\alpha\cos\beta & \cos\alpha\cos\beta \end{bmatrix} \tag{3}$$

Where $\boldsymbol{R_x}$, $\boldsymbol{R_y}$, and $\boldsymbol{R_z}$ are the rotation matrices around each individual axis, $\boldsymbol{R}$ is the combined rotation matrix around the scan origin, and ***P'*** is a vector of the point coordinates in the new coordinate system. Note that a translation along *X'*, *Y'*, and *Z'* needs to be performed after rotation for complete registration. When applying only level corrections, $\boldsymbol{R_z}$ would be an identity matrix and γ would be zero, simplifying $\boldsymbol{R}$.

2.1.2.2 Inclination Sensors[1]

Many modern terrestrial laser scanners (TLSs) are equipped with inclination/tilt sensors—also referred to as level compensators—that can correct out-of-level imperfections in an instrument setup. The geo-referencing and real-time change algorithm is highly dependent on accurate inclination readings to geo-reference data in real-time. The scanner's internal inclination sensors measure roll and pitch of the instrument, while yaw represents the scanner's bearing or azimuth relative to an external project coordinate system. With traditional LiDAR post-processing, the roll and pitch are extracted from within manufacturer software. When running *Lichen*, these values are extracted through the custom scanner interface (e.g. the DriveVZ400 program).

Rigorous testing revealed that caution must be exercised in scan planning to ensure quality inclination sensor data. Specifically, lab tests indicate that scanner rotation speed influences the reliability of inclination sensor readings recorded during rotation. These inertial effects can be cancelled, to a limited extent, by measuring during both clockwise and counter-clockwise rotations of the scanner (preferably not during a scan to minimize disturbance). Using scan durations within a "stability threshold" can also minimize the inertial effects and variability in inclination sensor readings for scanners continuously measuring inclination data during rotation. Finally, inclination sensor readings from a full 360° rotation are recommended to reduce systematic bias. See Silvia and Olsen (Accepted) for more details. These findings were implemented into the data acquisition portions of this project to ensure quality results.

2.1.2.3 Azimuth (Rotation about Z-axis)

The azimuth values for scan positions are initially estimated using a digital or manual compass and then translated into a scanner yaw angle using the following equation:

$$\gamma = AZ - 90 \ \ (\text{for AZ} \leq 270^{\mathrm{o}})$$

$$\gamma = AZ - 450 \ \ (\text{for AZ} > 270^{\mathrm{o}}) \tag{4}$$

Where γ = the rotation about the Z-axis, and AZ = the azimuth measured using the compass (corrected for declination).

[1] Full details of the calibration procedure, analysis, and a case study of application can be found in:
Silvia, E.P., and Olsen, M.J., (Accepted with revisions). "To level or not to level: Laser scanner inclination sensor accuracy evaluation," *ASCE Journal of Surveying Engineering*.

The rotation value is then improved using the least squares solution presented in Olsen et al. (*2011*), which calculates the optimal rotation about the Z-axis of a scan to best fit a second scan. The solution was revised to eliminate the small angle approximation used in the original derivation. This re-derivation reduces the iterations required to determine the rotation angle of the first scan compared to matching points in the second scan, particularly at larger angular offsets. The following equation describes the rotation about the Z-axis of scan B to fit matching points in scan A, with residuals:

$$\begin{Bmatrix} v_{x1} & v_{y1} \\ v_{x2} & v_{y2} \\ \vdots & \\ v_{xn} & v_{yn} \end{Bmatrix} = \begin{Bmatrix} x_{B1}+X_{B0} & y_{B1}+Y_{B0} \\ x_{B2}+X_{B0} & y_{B2}+Y_{B0} \\ \vdots & \\ x_{Bn}+X_{B0} & y_{Bn}+Y_{B0} \end{Bmatrix} \begin{bmatrix} \cos\alpha_B & -\sin\alpha_B \\ \sin\alpha_B & \cos\alpha_B \end{bmatrix} + \begin{Bmatrix} X_{B0} & X_{B0} \\ X_{B0} & X_{B0} \\ \vdots & \\ X_{B0} & X_{B0} \end{Bmatrix} - \begin{Bmatrix} x_{A1} & y_{A1} \\ x_{A2} & y_{A2} \\ \vdots & \\ x_{An} & y_{An} \end{Bmatrix} \qquad (5)$$

Where

v_{xi}, v_{yi} = residuals in the x and y directions of the i^{th} point,

x_{Ai}, y_{Ai} = the x and y coordinates of the i^{th} point in *Scan A*,

x_{Bi}, y_{Bi} = the x and y coordinates of the i^{th} point in *Scan B*,

x_{B0}, y_{B0} = the x and y coordinates of origin of *Scan B*, and

α_B = the necessary rotation of S*can B* to align it with *Scan A*.

To apply the least squares method, we need to calculate the sum of the squares of the errors:

$$V_{SS}^2 = v_{x1}^2 + v_{y1}^2 + v_{x2}^2 + v_{y2}^2 + \cdots + v_{xn}^2 + v_{yn}^2 \qquad (6)$$

To minimize, we can differentiate the above equation and set it equal to zero:

$$\frac{\partial V_{SS}^2}{\partial \alpha_B} = \frac{\partial v_{x1}^2}{\partial \alpha_B} + \frac{\partial v_{y1}^2}{\partial \alpha_B} + \frac{\partial v_{x2}^2}{\partial \alpha_B} + \frac{\partial v_{y2}^2}{\partial \alpha_B} + \cdots \frac{\partial v_{xn}^2}{\partial \alpha_B} + \frac{\partial v_{yn}^2}{\partial \alpha_B} = 0 \qquad (7)$$

which has a least squares solution:

$$\alpha_B = \tan^{-1}\left(\frac{\overline{\Delta y}}{\overline{\Delta x}}\right) \qquad (8)$$

where:

$$\overline{\Delta y} = \sum_{i=1}^{n} x_{Ai} y_{Bi} - \sum_{i=1}^{n} y_{Ai} x_{Bi} + y_{B0}\left(\sum_{i=1}^{n} x_{Bi} - \sum_{i=1}^{n} x_{Ai}\right) + x_{B0}\left(\sum_{i=1}^{n} y_{Ai} - \sum_{i=1}^{n} y_{Bi}\right) \tag{9}$$

and

$$\overline{\Delta x} = \sum_{i=1}^{n} x_{Ai} x_{Bi} + \sum_{i=1}^{n} y_{Ai} y_{Bi} - y_{B0}\left(\sum_{i=1}^{n} y_{Ai} + \sum_{i=1}^{n} y_{Bi}\right) - x_{B0}\left(\sum_{i=1}^{n} x_{Ai} + \sum_{i=1}^{n} x_{Bi}\right) + n\left(y_{B0}^{2} + x_{B0}^{2}\right) \tag{10}$$

2.1.2.4 Translation

Upon completion of rotations, the scan is then translated into the final coordinate system. The translation values can be obtained through either a GPS/GNSS receiver, or by setting the instrument up over a control point, whose coordinates were established previously via long-duration static GPS or Total Station surveys. While one could use a local, arbitrary coordinate system, we recommend that a standard coordinate system be used such that the data can be integrated with other datasets or re-established should the control monuments be disturbed.

Coastal studies utilized RTK GPS with corrections streamed in real-time using the ORGN GPS network (http://www.theorgn.org/), which allows for cm-level accuracy with short occupations (5 seconds +). For quality control purposes, we recorded multiple GPS-derived coordinates. In areas where the ORGN network is not available, we implemented Rapid Static GPS, which was post-processed through the Online Positioning User Service (OPUS, http://www.ngs.noaa.gov/OPUS/). For the continuous monitoring test, the scanner was setup over a simulated control point. In these instances, translation is not necessary to perform real-time analysis. However, translation from the control points coordinates would be necessary to move scans into an external coordinate system.

Note that GPS units store coordinates in WGS84 natively. Hence, scans can easily be updated to any coordinate system by re-projecting the native GPS coordinates into the desired coordinate system and re-performing the azimuth adjustment.

2.1.2.5 GPS Antenna Height Corrections for RTK GPS Coordinates

The methodology of Olsen et al. (*2011*) and most software packages assume that the correction applied to the GPS coordinates for the height of the GPS antenna above the scanner origin is purely vertical, and hence, is only applied in the Z-direction. However, un-level setups create offsets in the X and Y directions, as well. An updated correction was developed such that the scanner origin could be represented with improved coordinates. Although generally small (typically < 1 cm), this correction can lead to improved results for change detection. The correction is applied through the following equation:

$$\begin{Bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{Bmatrix} = R \begin{Bmatrix} 0 \\ 0 \\ -(H+h) \end{Bmatrix} \tag{11}$$

Where: $R = R_z R_y R_x$, as defined in Equation 1,

$\Delta X, \Delta Y, \Delta Z$ are the corrections applied to the X,Y,Z GPS coordinates,

H = the distance between the bottom of the GPS antenna and the scanner origin, and

h = the internal phase offset in the GPS antenna.

Note that if the scanner and GPS antenna were perfectly level, the offset would remain entirely in the Z-direction. Also note that typically the phase offset (h) is automatically applied to the GPS coordinates, even if the user enters an antenna height of 0 meters. Hence, either the h term can be neglected in the equation or a reverse correction needs to be applied. Generally, h is very small and can be neglected for setups that are only a few degrees out of level, without significant effects on the results.

2.1.3 Distance Cut-off Threshold Analysis for Corresponding Points

One of the key challenges in geo-referencing point cloud data is that points are not acquired at the "exact" same location for multiple scans. Besl and McKay (*1992*) introduced the Iterative Closest Point (ICP) algorithm, which selects point pairs in the first scan to compare to the closest points in the second scan for the least squares formulation. Bernardini and Rushmeier (*2002*) provide a detailed discussion of several methods developed to improve the ICP method and its application to scanning objects.

Because the point pairs are not really the same point in reality, bias will exist in the matching points. Zhang (*1994*) introduced parameters such as a distance threshold (generally determined through statistical analysis) to reduce inconsistent point-pair matching. The premise of this methodology is that points that are above this distance threshold should not be paired together. While the statistical procedure works well for aligning scans performed in the same survey, it may over-predict the distance threshold when aligning a new scan to a baseline survey when substantial change has occurred. Additionally, the resolution of the scan can play a role in changing and lowering the distance threshold when statistical analyses are performed.

Hence, a study was performed to analyze a more realistic distance threshold at which scan point pairs would be consistent and when it would be clear that change has occurred.

Twelve scans performed at Beverly Beach State Park were aligned and the distances between corresponding points were analyzed. Figure 2.4 shows the average number of point pairs that meet the distance cut-off threshold criteria versus distance threshold for all of the scan pairs. Figure 2.5 plots the calculated average root mean square (RMS) against the distance threshold criteria. From these plots, one can see a pivot point emerge at 0.05 meters. The number of point pairs rapidly drops off when a distance threshold below 0.05 meters is used. The RMS values

(Figure 2.5), also show this as a point of maximum curvature. Hence, when a distance cut-off threshold below 0.05 meters is used, the RMS appears artificially low because valid point pairs have been removed. However, above this threshold, the RMS value is artificially high because invalid point pairs are used in the RMS calculation. To be conservative in the RMS estimates, a distance threshold of 0.1 meters was used for this project. An additional consideration is that during the initial phase of the alignment, for scans that are several degrees out of alignment, a high threshold will enable faster convergence because it will result in larger rotation values (see Equation 8).

Figure 2.6 and Figure 2.7 plot each scan individually. Note that the scans were performed at varying resolutions. The first scans (e.g. 1,2,3, etc.) were performed at a lower resolution. As such, they have a higher RMS and lower point pair count. However, as the distance threshold is decreased, the influence of resolution is less substantial and the values from all of the scans tend to converge (lower standard deviation).

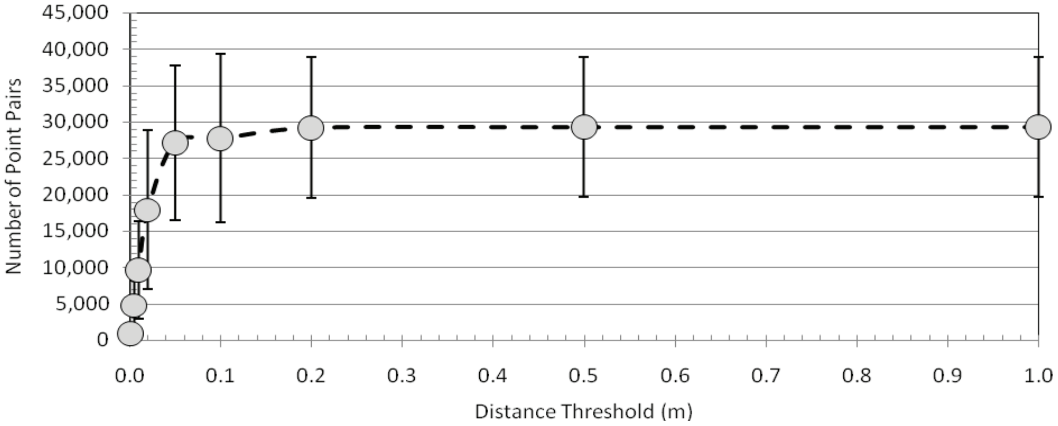


Figure 2.4: Average number of point pairs versus distance threshold.

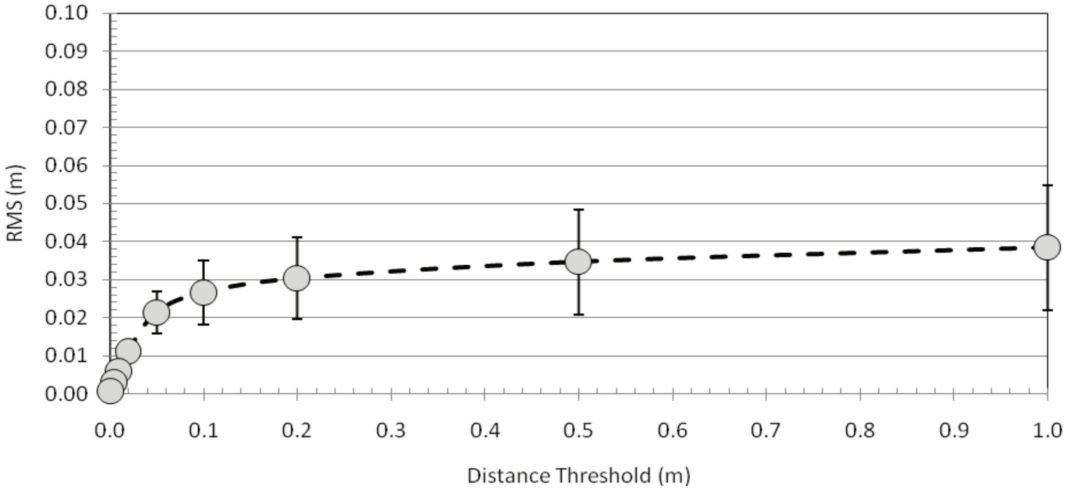


Figure 2.5: Average RMS of point pairs versus distance threshold used

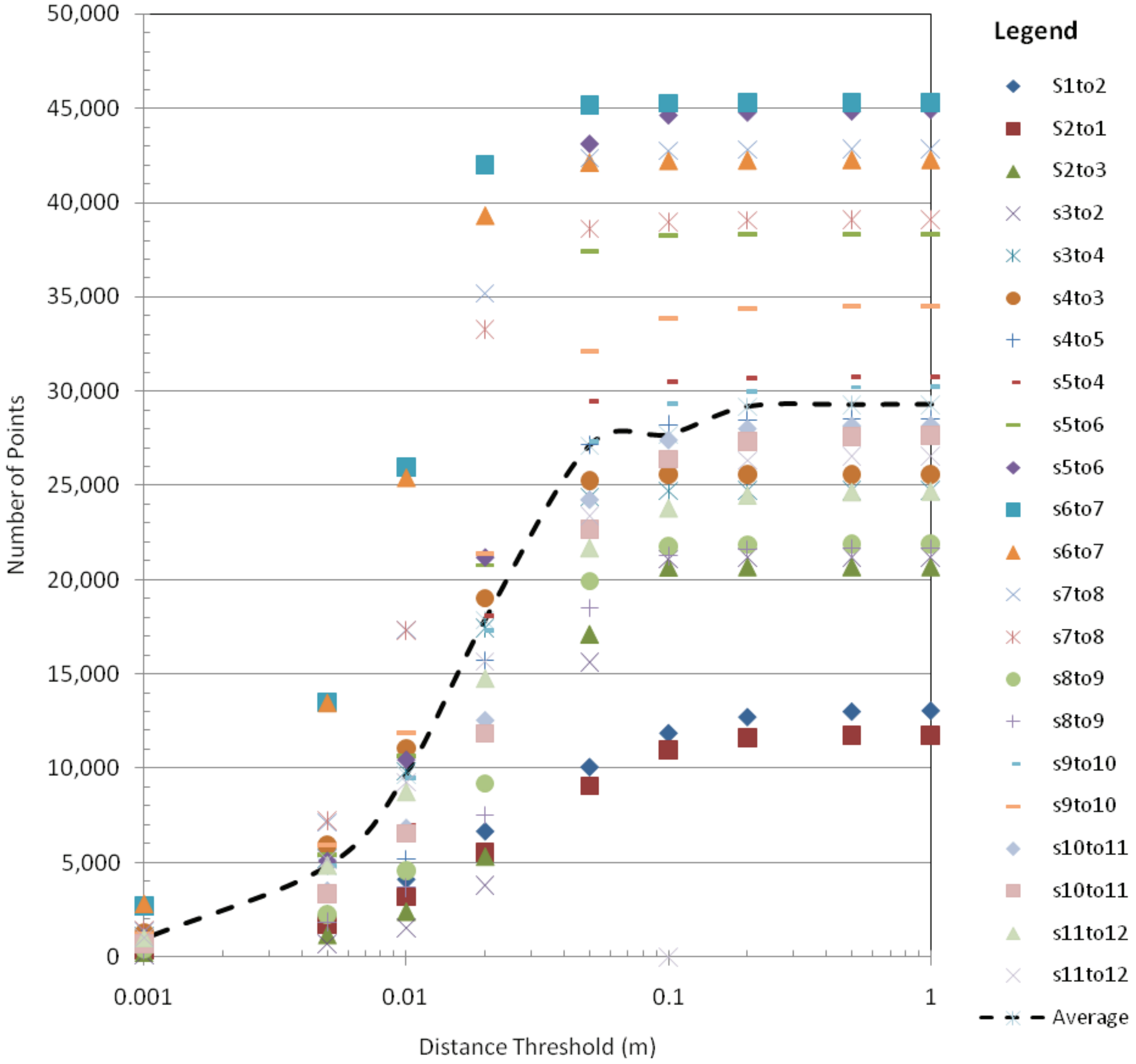


Figure 2.6: Number of point pairs versus distance threshold for each pair of scans (e.g., S1to2 = scan position 1 compared to scan position 2). Note the log scale used to plot the distance threshold.

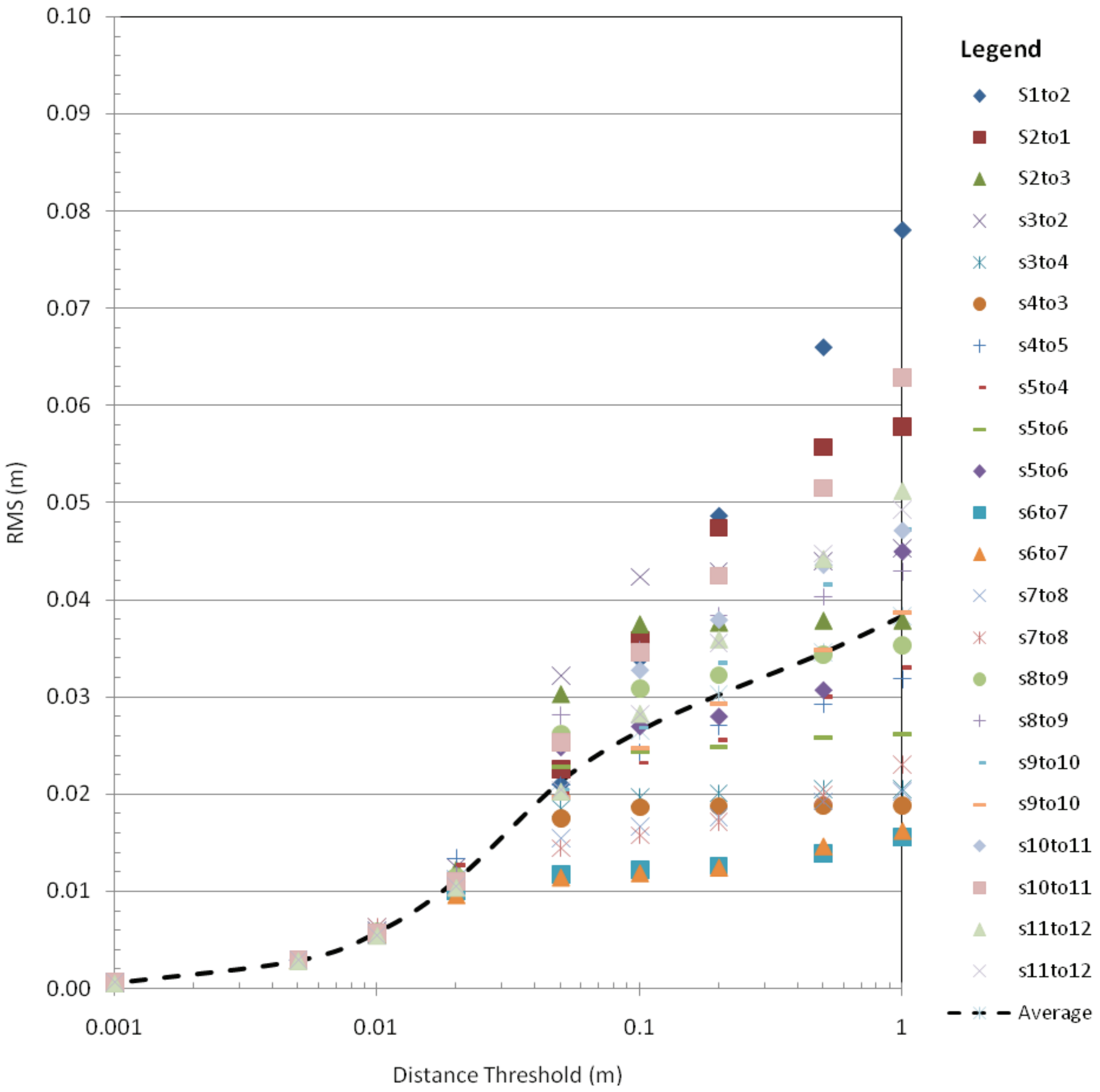


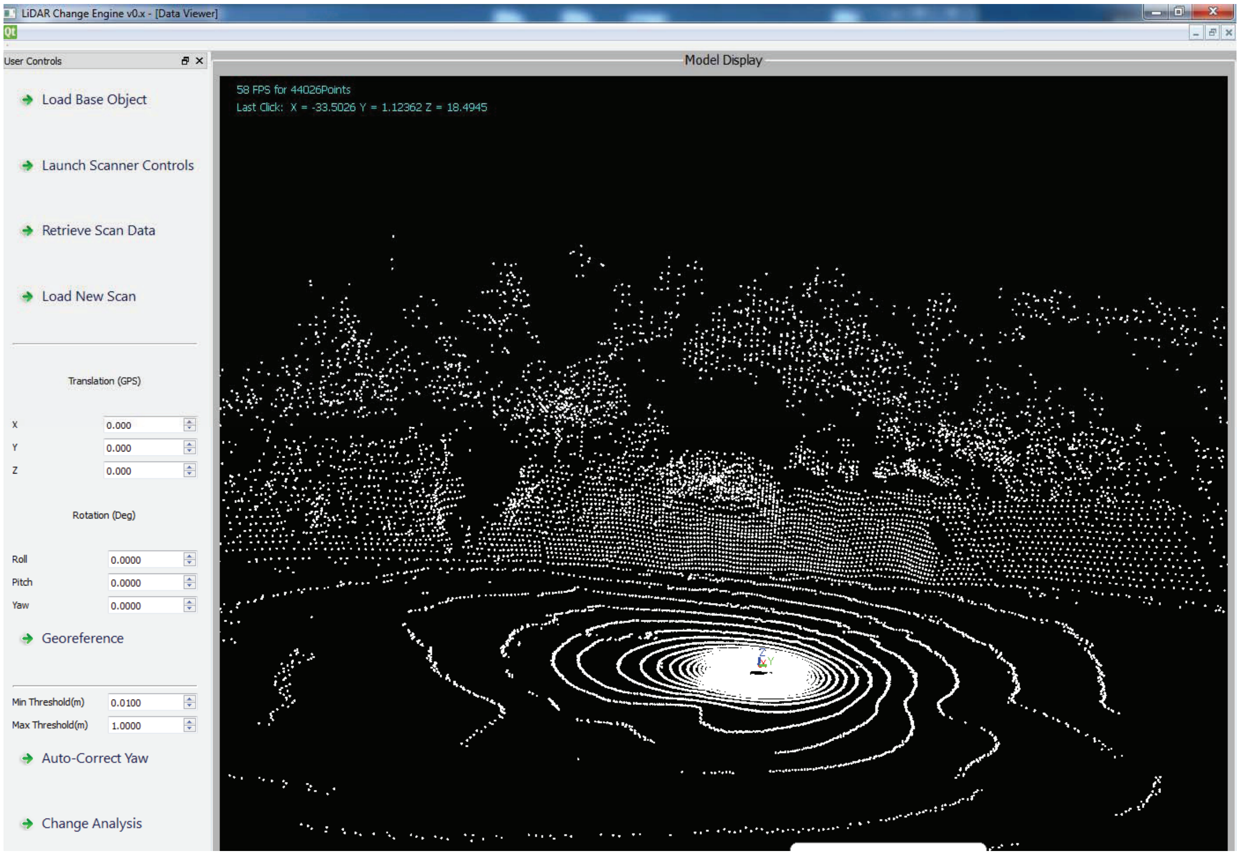
Figure 2.7: RMS of all scan alignments
(e.g., S1to2 = scan position 1 compared to scan position 2) versus distance threshold. Note the log scale used to plot the distance threshold.

2.2 PROGRAM DEVELOPMENT

The *Lichen* program consists of several key components, including a GUI with an interactive display of the point cloud data, an interface to the scanner, a geo-referencing module, and the change detection algorithm. Appendix A provides a user manual to the program. The program was designed to be modular such that each part could be updated separately as new advances take place and additional scanner SDKs are available. Note that the program supports data from any scanner; however, in order to optimize the workflow, a simplified, control module would need to be created for each scanner. Appendices B and C discuss data conversion details for using Lichen with a variety of scanners. Appendix C is most applicable to those using Leica Scanners, such as currently used by ODOT Geometronics.

2.2.1 Graphical User Interface

The graphical user interface (GUI) empowers the user with all operability needed to perform the real-time change analysis process. The GUI was designed to be a small, lightweight, powerful, and intuitive user interface conducive to efficient usage under field conditions (see Figure 2.8). It is laid out in a logical manner that progresses based upon the order of steps needed to perform a change analysis.



```
58 FPS for 44026Points
Last Click:  X = -33.5026 Y = 1.12362 Z = 18.4945
```

X 0.000
Y 0.000
Z 0.000

Roll 0.0000
Pitch 0.0000
Yaw 0.0000

Min Threshold(m) 0.0100
Max Threshold(m) 1.0000

Figure 2.8: The GUI consists of a large display window to improve visualization of the LiDAR data sets. The user controls are located in a fixed pane on the left, following a logical progression of how the user would use the program for in situ change detection.

The GUI was developed in the open-source environment Qt Creator (http://qt.nokia.com/) using the C++ programming language. This provides powerful functionality while also allowing for rapid GUI development and cross-platform portability. Each aspect follows the modular theme allowing specific buttons to be altered without impacting the general functionality of other buttons. This should promote more rapid future development without the need to rewrite extensive sections of code each time a modification or update is needed.

A customized version of the open source GLC Library (http://www.glc-lib.net/), based on OpenGL, was implemented for the point cloud visualization. This display is fully interactive so that the user can easily navigate the 3D scene.

2.2.2 Scanner Interface

A modular scanner interface was designed such that it will be able to incorporate additional scanners as software development kits (SDK) become available from the manufacturers.

DriveVZ400 (see manual in Appendix D) provides a suite of tools to operate the scanner as well as facilitate *Lichen*'s operation. This includes the ability to control the scanner directly or remotely from within the *Lichen* software. *Lichen* retrieves a monitor file, which is a subset of the entire scan, to enable display during data acquisition and rapid processing. For real-time change analysis, this monitor file is compared to the baseline scan for change detection. The program also generates an additional file with data retrieval information such as the project name, location, and the inclination data from the scanner.

This data can be retrieved manually from manufacturer software for other types of scanners.

2.2.3 Geo-referencing Module

Controls were developed to enable the user to enter scan geo-referencing parameters. The methodology used for geo-referencing is the same as that used for the baseline scan creation discussed in Section 2.1, except for the baseline model is generally used for alignment rather than adjacent scans.

2.2.4 Real-time Change Algorithm

The real-time change algorithm is the heart of the process of the *Lichen* workflow (Figure 2.1). This algorithm uses a point-to-point comparison by searching for the nearest neighbor in the baseline scan to a given point in the new scan. Based upon this criteria, points are classified by whether they are out of bounds (beyond the scope of detection), greater than the maximum threshold (change), less than the minimum threshold (no change), or in between (currently shaded based upon the degree of intermediacy). These values are provided by the user depending on the magnitude of change that would be of interest.

Prior to searching for neighbors, the baseline scan is divided into cube subsets based upon the chosen thresholds and the size of the scan. This spatial sorting optimization decreases the time necessary to complete the change analysis by reducing the number of points needed in the comparison. Figure 2.9 illustrates the procedure to search for a nearest neighbor. Upon completion of the new scan, the algorithm then locates and assigns the proper cube for each of the new scan vertices and checks to make sure they are within the bounds of the baseline scan cubes. If not, they are assigned an out of bounds code which is interpreted as meaning there is no data to compare the point against. If the point (*P*) is not out of bounds, it first looks in the same cube for a nearest neighbor in the baseline scan, comparing the distance between that point (*P*) and the current baseline point (*I*) to the current minimum distance to determine if point *I* is the closest point. (Note that for optimization, squared distances are used rather than distance).

The algorithm then searches through the remaining points in the cube. If, during this process, a point is found within the minimum distance threshold specified by the user, the algorithm immediately stops the search and returns that point rather than continuing the search. Should the point (*P*) be close to the boundary of a cube (e.g. point *P* in Figure 2.9), the algorithm expands the search in three dimensions to neighboring cubes located within the maximum distance to ensure that the closest point is found.

2D Closest Point Search Algorithm

Example for Point P

```
Search 5
If dist(P-A) < mindist, return A
else Search 2
If dist(P-B) < mindist, return B
else Search 3
If dist(P-C) < mindist, return C
else Search 6
If dist (P-D) < mindist, return D
else return closest(A,B,C,D)
```

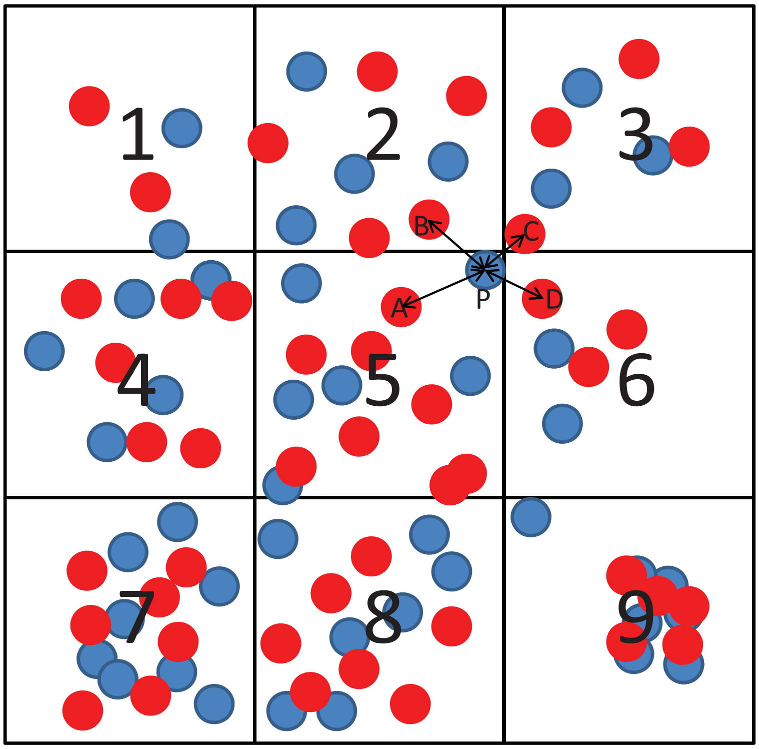


Figure 2.9: Schematic showing selection of point pairs from a point in the new scan (red) on the boarder of the cube to find nearest neighbor in the baseline scan (blue).
Note that for simplicity, the diagram was developed in 2 dimensions, but the actual algorithm functions in 3 dimensions.

This distance to the closest point is then used to calculate the color shading used by *Lichen* to display change to the end user. Points located out of bounds are shaded gray. Those found less than the minimum threshold distance from the baseline surface are shaded blue. If they are beyond the maximum threshold they are shaded white. Points in between the two thresholds are shaded an intermediary mix between blue and red. Future developments will enable the user to select their own coloring scheme.

A potential problem in change analysis resides in distinguishing between "overall change" and "error" in the data. This dilemma could be resolved in the field with the proposed work as opposed to returning to the office to discover this in post processing. With the real-time display and feedback, the operator can spot that "apparent error" and take additional measurements to verify whether it is in fact error or uniform change while still at the site, and, if necessary, use additional equipment to double check (see Figure 2.10). Without this, the questions must be discussed and evaluated in the office, where additional field visits will be required to verify. Because the data are constrained to GPS coordinates for the scan origin and level values from internal scan sensors, only the azimuth of the scan is dependent on the baseline surface. However, external digital compass measurements provide a reasonable estimate (within a few degrees) of this value, should significant change take place. Because of these constraints and external checks, overall movements and recessions across the site can be detected. In areas of concern, one could use adjacent scans for the reference surface (as is done in the baseline scan creation) to determine the azimuth adjustment.

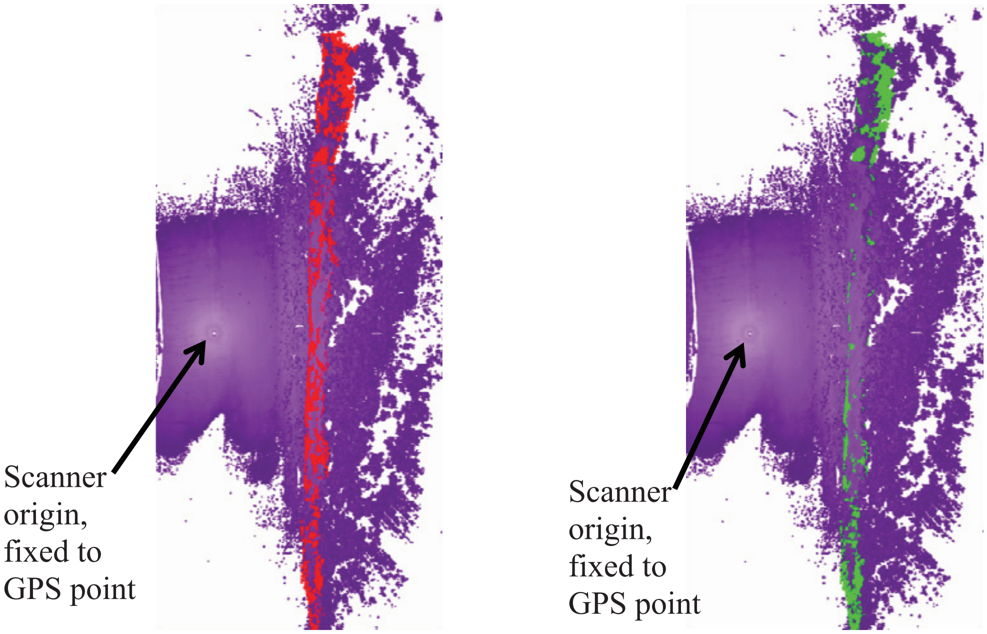

Scanner origin, fixed to GPS point
Scanner origin, fixed to GPS point

Figure 2.10: Left: 2011 scan of Johnson Creek landslide compared to 2007 baseline surface showing actual change. Right: same scans with an artificial overall change of 1m towards the East. Note that the scanner origin, rotation about X&Y axes do not vary between the left and right scans because of the constraints. However, there is a difference in the azimuth (0o27’) created from the overall change. In such a case, one could correct this by using the adjacent scans from the same survey for alignment.

2.2.5 Optimizations

Several optimizations have been included to improve the performance and operability of *Lichen*. Binary files are used to reduce file size and improve processing speed. Raw point clouds are very large (they can be several hundred megabytes to gigabytes in size) and using binary format helps ensure that lower-end hardware and mobile platforms can effectively handle these large data sets. The use of monitor files (reduced versions of the point clouds) also addresses this same issue.

Where possible within the code, we have removed or omitted steps which make redundant calls or calculations in an effort to reduce calculations and memory needed to operate *Lichen*. Areas that might cause memory leaks (the program consumes memory but is unable to release it when it is no longer needed) are identified and removed as soon as possible to improve stability.

In addition, the entire workflow of the *Lichen* user interface has been optimized to include as many degrees of automation as is currently possible. This reduces the possibility of user-derived error, improves the user computer experience, and expedites the real-time change analysis. At this time, it is not anticipated to make *Lichen* fully automated as the human user is an essential Q/C agent in both the baseline scan and real-time analysis processes.

2.2.6 Limitations

There are a few limitations to the usage of *Lichen* that users should be aware of so that they can be taken into account during the decision-making process. Many of these limitations are inherent to any change analysis. Incorrect usage of the software may give results that are out of context.

The greatest source of error in using *Lichen* involves the degree of accuracy that can be achieved in scan geo-referencing. GPS-based determination allows for the greatest mobility and flexibility in using the program in the field, but it also has significantly less accuracy than setting up over a control point. Users are encouraged to make quality control checks of their GPS measurements and understand that there are several possible results that will lead to incorrect conclusions regarding change between epochs.

1. Easting or X-axis precision: Let us use an example of a sea cliff running from north to south with the ocean to the west and the cliffs to the east. If GPS is used to fix the scanner position, then there will most likely be 1-3cm of positional uncertainty. If the true position is farther from the cliffs than the measured position, change analysis will incorrectly assign additional erosion of material than what has actually occurred. If the true position is closer to the cliffs than the measured position, change analysis will incorrectly assign additional deposition of material than what has actually occurred.

2. Northing or Y-axis precision: Using the same example given above, this positional uncertainty also means that the Y-axis of the new scan will differ slightly from its true position. This may falsely improve or degrade the alignment quality of the scan and introduce some amount of error into the change analysis.

3. *Lichen* does require some knowledge of the observed terrain to best interpret the results returned by the program. This is important mostly in regards to vegetated areas. By its nature, vegetation is always changing and creates quite a lot of noise in terrestrial laser scans. This in turn shows up in the change detection algorithm and is one reason why human operators are necessary to provide quality control for the returned results.

4. In areas of very high change, the position of the scanner's real-world coordinates become even more important. Alignment of the scans will become more difficult and less accurate as more time elapses between the scans. This is because scans are fixed in place and rotated to determine optimum alignment. However, what should also be apparent when scanning these areas is that significant change has occurred which would prompt the user to re-scan.

5. Currently, few manufacturers make available their SDKs to allow programmers to interface directly with each individual scanner. As a result of this limitation, *Lichen* currently only runs at optimal efficiency with Riegl VZ400 series scanners. Note that *Lichen* can still import data from any system.

3.0 DATA COLLECTION AND ANALYSIS RESULTS

3.1 SUMMARY/OVERVIEW

During the course of this project, data has been collected from several sites in Oregon to build baseline scan models to test the real-time change detection program as well as to expand our ability to monitor changes to key topographic regions. We chose several sites reflecting areas of public safety and economic concern from which to gather high resolution terrestrial LiDAR data. Starting in the fall of 2010 and continuing through to the present, data were collected from the Johnson Creek landslide and Beverly Beach (along Highway 101), US 20 Pioneer Mountain-Eddyville re-alignment project, and a mechanically stabilized earth (MSE) retaining wall located in Central Point. These data are summarized below and have been processed into triangulated object files, used as baseline scans for change-detection analysis during repeat field testing. Figure 3.1 shows an example dataset from Beverly Beach, and Figure 3.2 shows data from the Johnson Creek landslide. The examples in this chapter show a post-processing change results.

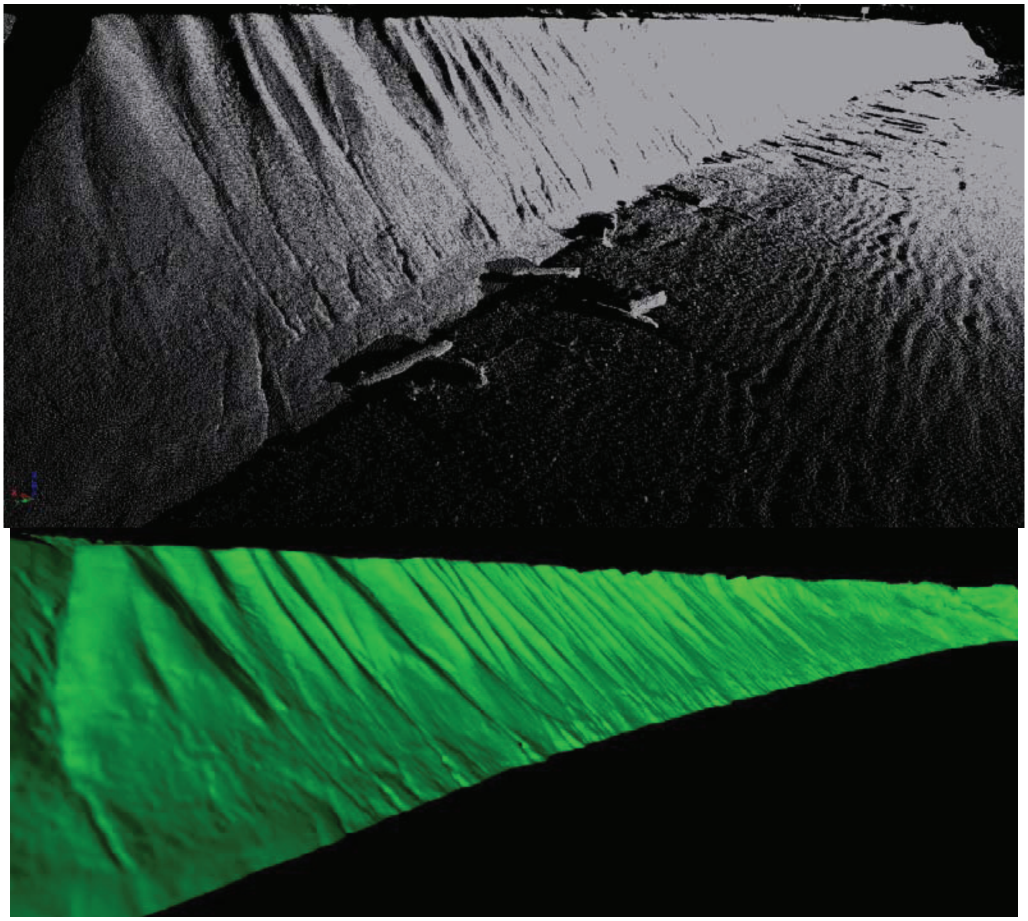


Figure 3.1: Example of a geo-referenced point cloud (top) and triangulation (bottom) at Beverly Beach State Park in January 2011.

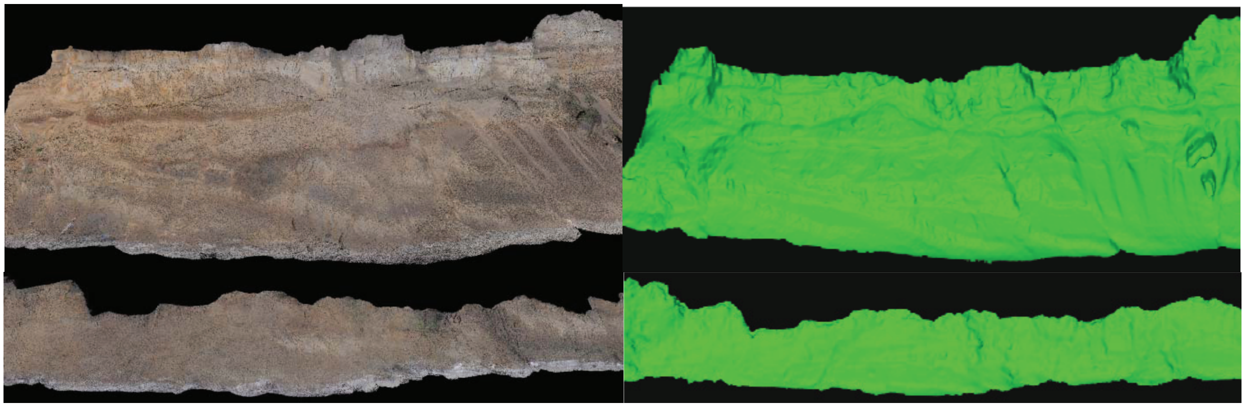


Figure 3.2: Two different sections of bare cliffs from Johnson Creek Landslide in June 2011. The left images are the geo-referenced and cleaned point clouds, while the right images show the resulting 3D triangulated models.

3.2 JOHNSON CREEK/BEVERLY BEACH[2]

3.2.1 Introduction

The Johnson Creek landslide (JCL) creates chronic maintenance problems for the ODOT where U.S. Highway 101 crosses the slide mass. The landslide is approximately 360 meters wide, 200 meters long, and terminates on the beach where ocean waves and subaerial processes are eroding its toe. To understand the rates and patterns of erosion along the bluff face of the JCL, DOGAMI and ODOT implemented a study in 2002 to document movement. This study had several key objectives including: 1) characterization of the geology of the landslide block, including the rates and patterns of slide movement; 2) examination of the processes that contribute to landslide movement; and, 3) examination of the spatial and temporal variability of mass wasting along the seaward edge of the slide block. Landslide Technology (*2004*) and Priest et al. (*2008*) performed a detailed analysis of the landslide including numerical modeling and an investigation of historical precipitation. Priest et al. (*2008*) also performed preliminary laser scan data analysis. Schulz and Ellis (*2007*) augmented the earlier subsurface exploration and monitoring of the landslide by installing vertical arrays of piezometers and soil moisture probes. The results of these studies correlated rainfall and pore-pressure generation with landslide movement, as well as significant impacts of toe erosion on stability.

3.2.2 Motivation

As discussed in Chapter 1 of this report, substantial work has been undertaken to document either erosion rates or landslide movement patterns and rates using LiDAR. In the case of the Johnson Creek Landslide, the combination of both erosion and landslide movement to the overall change across the site presents a new set of challenges, including difficultly in distinguishing between erosion and landslide movement. With respect to erosion, Young et al. (*2009*) provide a

[2] A substantial portion of this section has been developed into the following conference publication: Olsen, M.J., Allan, J.C., and Priest, G.R. (Accepted with revisions), "Johnson Creek landslide movement and erosion quantification through 3D laser scanning," submitted to *GeoCongress*2012, ASCE Geoinstitute, Oakland, Calif..

comparison of common methods for calculating sea cliff retreat and observed that significant differences in values can be obtained whether calculating retreat at the toe or crest of the cliff, or using an average across the cliff face.

For the purposes of this study, we focus on monitoring of a continually moving landslide. Our ultimate objective is to separate movements of the landslide from the active erosion caused by wave attack and sub-aerial processes, and to analyze the contributions of each process to overall changes taking place on the face of the Johnson Creek landslide (see Figure 3.3). Evaluation of such feedback mechanisms is important to document when attempting to deal with potential mitigation strategies. For example, Priest et al. (*2008*) estimate through limit equilibrium analysis that a 1 meter recession of the toe results in a 2.3 percent reduction in the factor of safety of the landslide. Hence, increased erosion will likely trigger an increase in landslide movement, which may then move the landslide toe back into the swash/wave erosion zone.

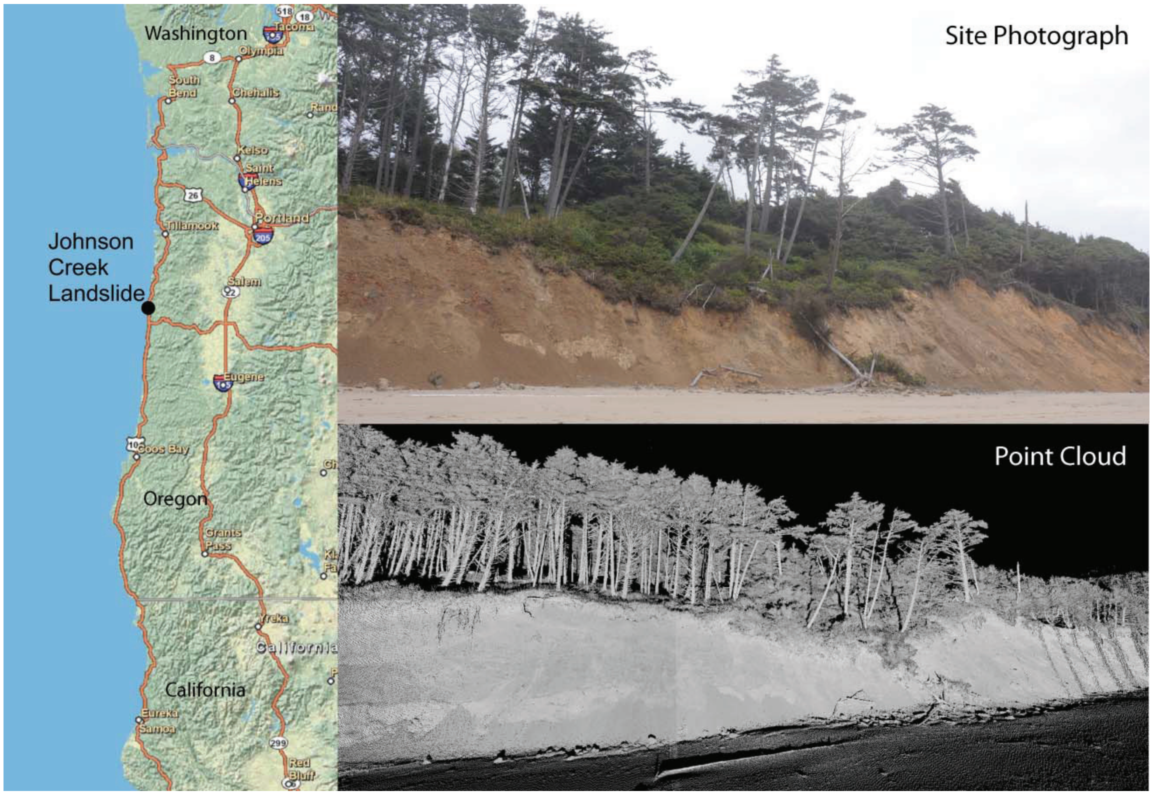


Figure 3.3: Location map, site photograph, and point cloud (August 2011) of the Johnson Creek landslide.

3.2.3 Methodology

Collins et al. (*2007*), Kayen et al. (*2010*) and Olsen et al. (*2008, 2009 and 2011*) provided an overview of typical laser scan acquisition and processing workflows, including discussion of common topographic data products. This section will only provide a brief overview and focus on details specific to this study.

3.2.3.1 Acquisition

Table 3.1 provides a summary of TLS surveys completed at the JCL site and overall data quantity and quality. The first TLS survey of the site, conducted in May 2004, was completed as a demonstration of the capabilities of TLS for landslide assessment. As such, only three discrete scans of the landslide were performed, which did not span the entire exposed face. ODOT Geometronics performed additional scans in October 2006 and June 2007. OSU recently completed surveys in December 2010, June 2011 and August 2011, which also included additional scans southward towards Beverly Beach State Park. Note that only scans relevant to JCL are summarized in Table 3.1. Future scan surveys will be acquired on a periodic basis, including scans on top of the landslide, rather than solely of the bluff face.

The significant increase in the number of data points and scans with time is a direct result of the rapid developments in laser scan technology. Airborne LiDAR datasets are also available through NOAA Digital Coast (*1997, 1998, and 2002*) and DOGAMI (*2009*). However, because of the look angle of airborne LiDAR, there is limited coverage on the cliff face in these datasets, reducing its utility for this study.

Table 3.1: 3D laser scan surveys completed of the landslide.

Date	Surveyor	# scans	#points ($x10^6$)	Typ. point spacing (cm)	Alignment Method	Alignment Error (RMSE, cm)	Length of survey (m)
05/2004	D. Wellman Surveying	3	2	5	Target	Unknown	150 (3 segments)
10/2006	ODOT	4	9	4	Target	Targets: 1 Points: 3	450
06/2007	ODOT	4	10	4	Target	Targets: 1 Points: 4	370
12/2010	OSU	8*	120	3	Hybrid	Points: 3	>600*
06/2011	OSU	8*	115	3	Hybrid	Points: 2	>600*
08/2011	OSU	7*	115	3	Hybrid	Points: 3	>600*

* = the scans used for this analysis were extracted from a much larger survey covering a longer section of the coast.

3.2.3.2 Processing

Overall processing of scan data consists of three steps: 1) The data must be geo-referenced; 2) the data must be cleaned of stray, non-representative points and filtered; and 3) surface models (DTMs) are created from the point cloud data, typically resulting in a 3D TIN.

Many methodologies exist for geo-referencing laser scan data (see Chapter 2). First, target based approaches use reflective (or identifiable) targets that have known coordinates in the scan window. The 2004, 2006 and 2007 surveys were all completed using the target based approach. Second, the scan can be directly geo-referenced by surveying control points where the scanner is set up. Third, software registration determines the least-squares, optimal alignment of a scan based on similar features in neighboring scans to merge scans. Olsen et al. (*2009; 2011*) presented a fourth (hybrid) approach for dynamic environments where control is difficult to establish and maintain, such as the coast. Scans are geo-referenced through RTK GPS coordinates, internal scanner leveling sensors, and a least-squares-based azimuth adjustment. This methodology was used for the 2010 and 2011 surveys, and RTK GPS corrections were obtained using the ORGN GPS network (http://www.theorgn.org/) to produce cm-level accuracy. Table 3.1 presents the alignment errors for each survey as: 1) RMSE for the positions of targets, and 2) RMSE of the points from the scan survey. Note that the RMSE comparing points in the scans are comparable between both survey methods. These cloud RMSE values tend to be higher than target values because targets are scanned in separately at higher resolution and can be better resolved.

Once the scans are aligned and merged, datasets are edited to remove artifacts, which can result from people or other objects passing in front of the scanner during acquisition. The data can also be filtered to remove noise or provide a more uniform point sampling throughout the scan, if desired. Digital Terrain Models (DTMs) can then be created by gridding or triangulating the point cloud. DTMs provide reference models to perform calculations and analysis, which can be difficult to analyze using the point cloud alone.

3.2.3.3 Analysis

A focus of this effort was to separate the outward landslide movement (towards the ocean) from the inward amount of erosion (away from the ocean). To estimate the amount of landslide movement along the landslide profiles, trees, stairwells, and houses on top of the cliff that were part of the landslide mass were used for differencing between the 2007 and 2011 surveys. When using trees to provide the landslide movement quantification, difference measurements were made as close to the base of the tree as possible so that tree rotations did not bias the results. Additionally, trees that were very close (<1 m) to the cliff edge were not used because localized instability generated large movements that were not representative of the actual landslide movement. Erosion rates (Δ_E) were calculated by using the maximum, measured erosion rate (Δ_M) found in the cross section analyzed. The effect of landslide movement (Δ_L) on the calculation of erosion rate can be determined from:

$$\Delta_E = \Delta_M - \Delta_L \qquad (12)$$

Positive values indicate cliff accretion or landslide movement towards the ocean; whereas negative values indicate erosion. Analyzing the rate of movement of the landslide in conjunction with the erosion rates provides a better estimate of the amount of material that is contributed to the beach from sea cliffs. In Figure 3.4, case 1 shows an example where no erosion would be detected without correction for the landslide movement. Here, the landslide movement would be incorrectly interpreted as cliff accretion. Case 2 shows an example where the erosion would be detected, but would be significantly underestimated without correction for landslide movement. Not only does this underestimate the actual erosion rates, but it also underestimates the eroded material volume. These differences could have significant effects when analyzing beach sediment budgets.

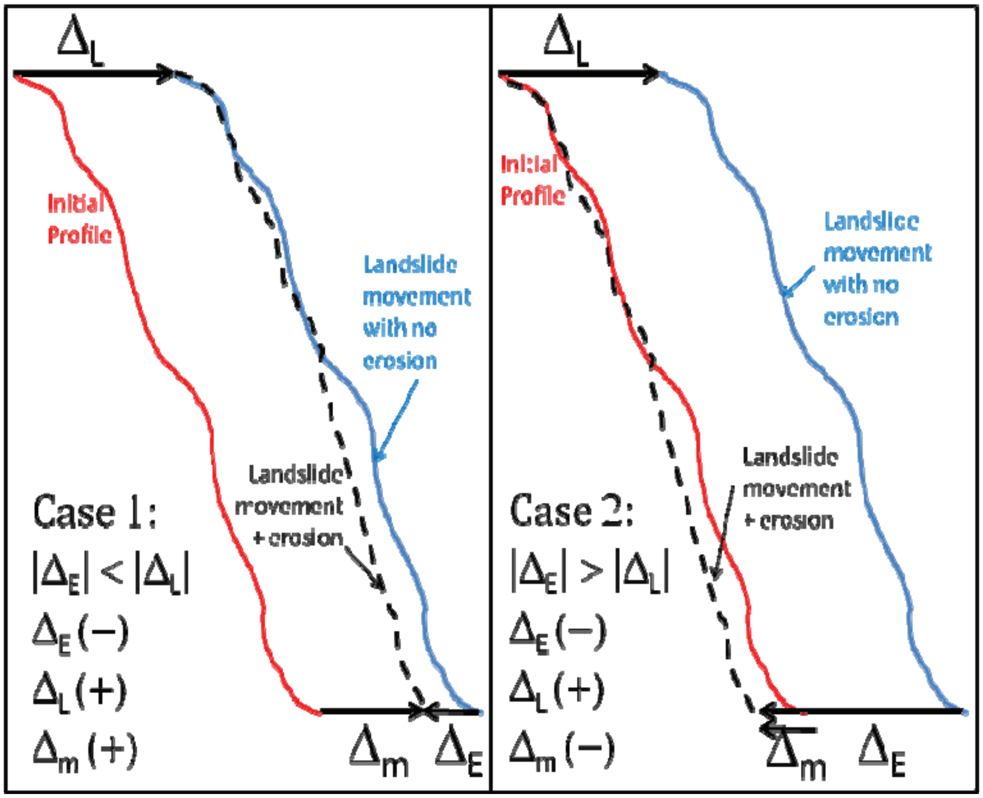


Figure 3.4: Conceptual diagram for correction of erosion rates for landslide movement.

3.2.4 Results and Discussion

3.2.4.1 Landslide Movement and Erosion Rates

Figure 3.5 shows the landslide movement, change rates and erosion rates for the JCL between the 2007 and 2011 surveys. Note the correlation between the areas of high erosion also showing significant landslide movements along the southern end of the landslide block, with those areas in the north where little change (erosion) is taking place. Additional data collected past the landslide extents were useful to verify the landslide boundary. Continual collection of data neighboring the landslide will allow for more in-depth analysis to compare nearby erosion trends. Thus far, very limited data are available in the 2004, 2006 and 2007 surveys to perform this type of assessment. Future monitoring will incorporate neighboring sections of sea cliffs for comparison.

The movement of the slide and the erosion of the toe are highly episodic, and as such, are not in short-term equilibrium. This is because there is a time lag between erosion during storm events and landslide movement. However, from long-term observations, it appears that the landslide and erosion tend to be close to equilibrium.

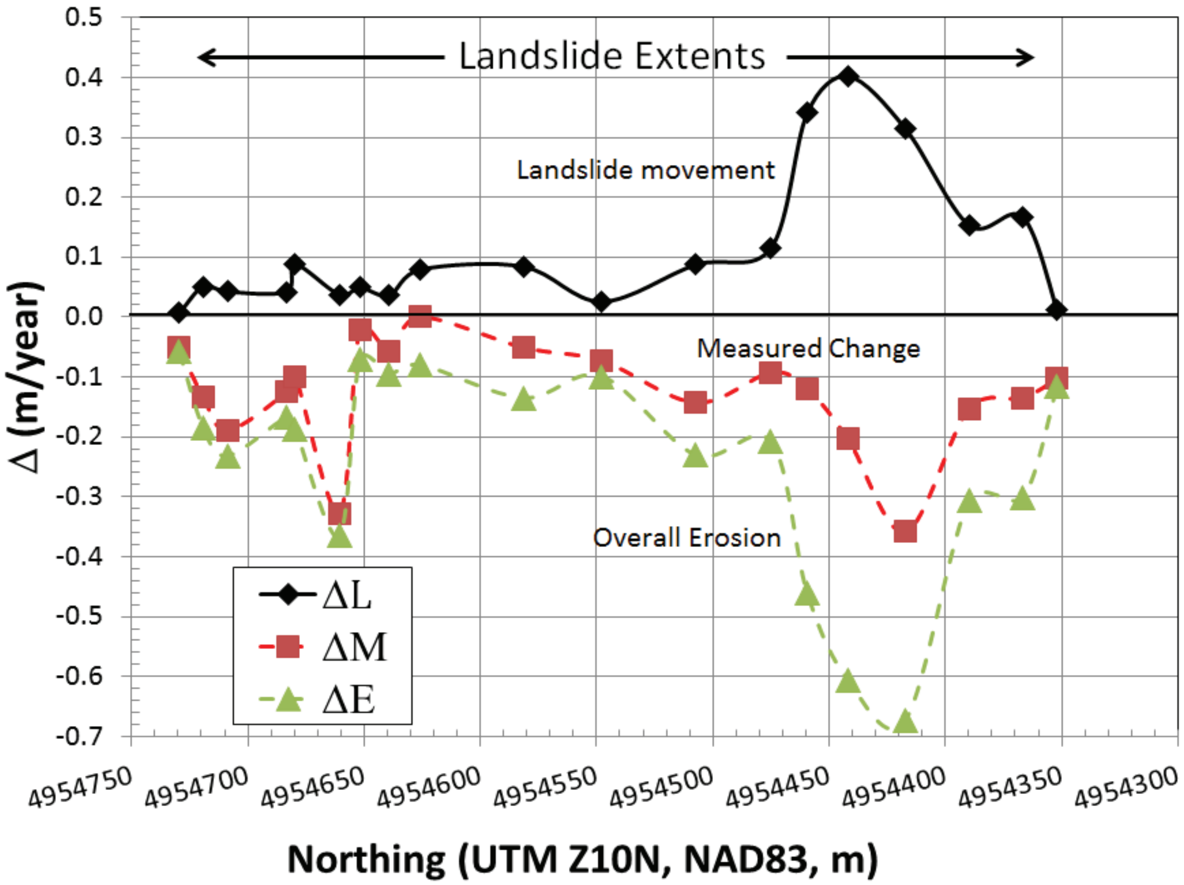


Figure 3.5: Landslide movement, change, and erosion rates calculated for the Johnson Creek landslide between the 2007 and 2011 surveys.
Positive values indicate movement towards the ocean and negative values represent movement away from the ocean.

3.2.4.2 Time Series Comparisons

Time series comparisons of the measured change (combined landslide and erosion) were performed using all of the LiDAR surveys. Figures 3.6, 3.7, and 3.8 show the observed change for the northern, central, and southern sections, respectively. Because of limitations in data coverage from the 2004 and 2006 surveys, the change values were calculated using the 2011 surface for the reference. Hence, orange values represent retreat of the cliff face from the ocean between the survey shown and the 2011 survey, while blue values represent overall advance of the cliff face toward the ocean.

Figures 3.6-3.8 indicate the same general trends of increased erosion to the south as previously shown (Figure 3.5) and have the advantage of showing the distribution of change across the cliff face. However, these figures only show net change, which does not include the landslide movement in the erosion rates. The variability of negative versus positive change shown in these figures highlights the fact that erosion calculations based on cross section analysis can be very sensitive to the location of the cross section, especially in the case of actively moving landslides.

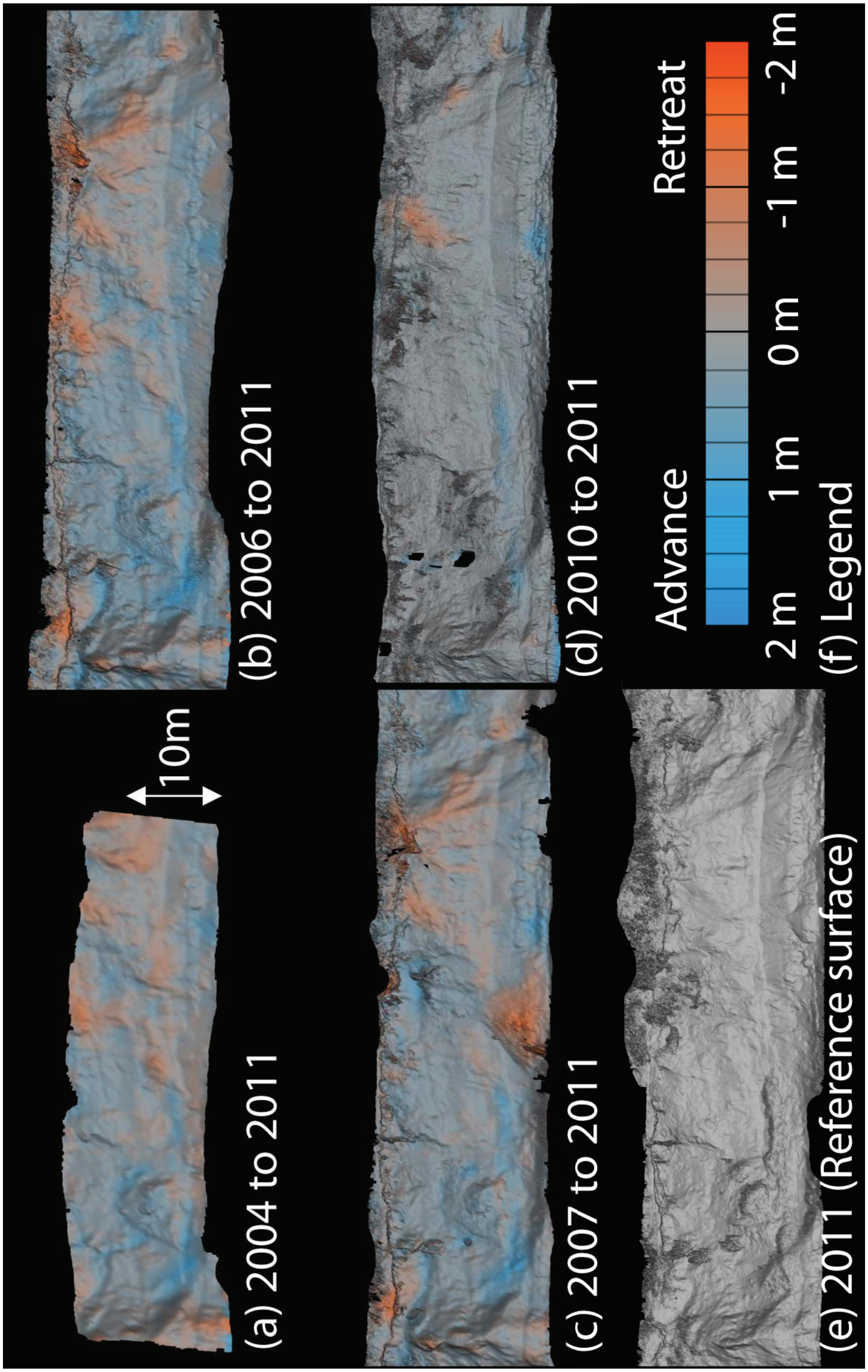


Figure 3.6: Change analysis between LiDAR surveys showing advance and retreat of the cliff face at the north section (Northing 4,954,580 m to 4,954,650m).

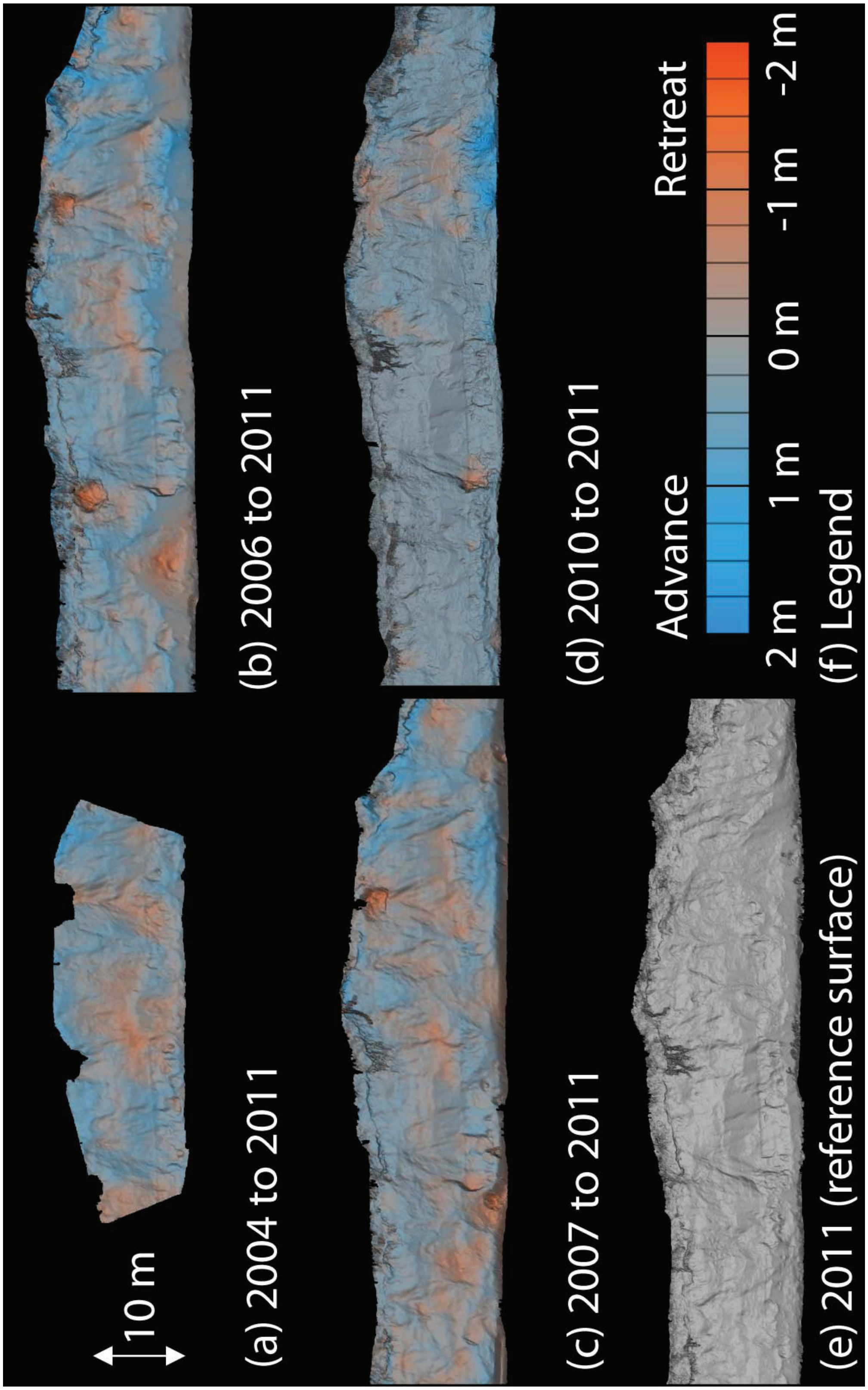


Figure 3.7: Change analysis between LiDAR surveys showing advance and retreat of the cliff face at the center section (Northing 4,954,450 m to 4,954,520m).

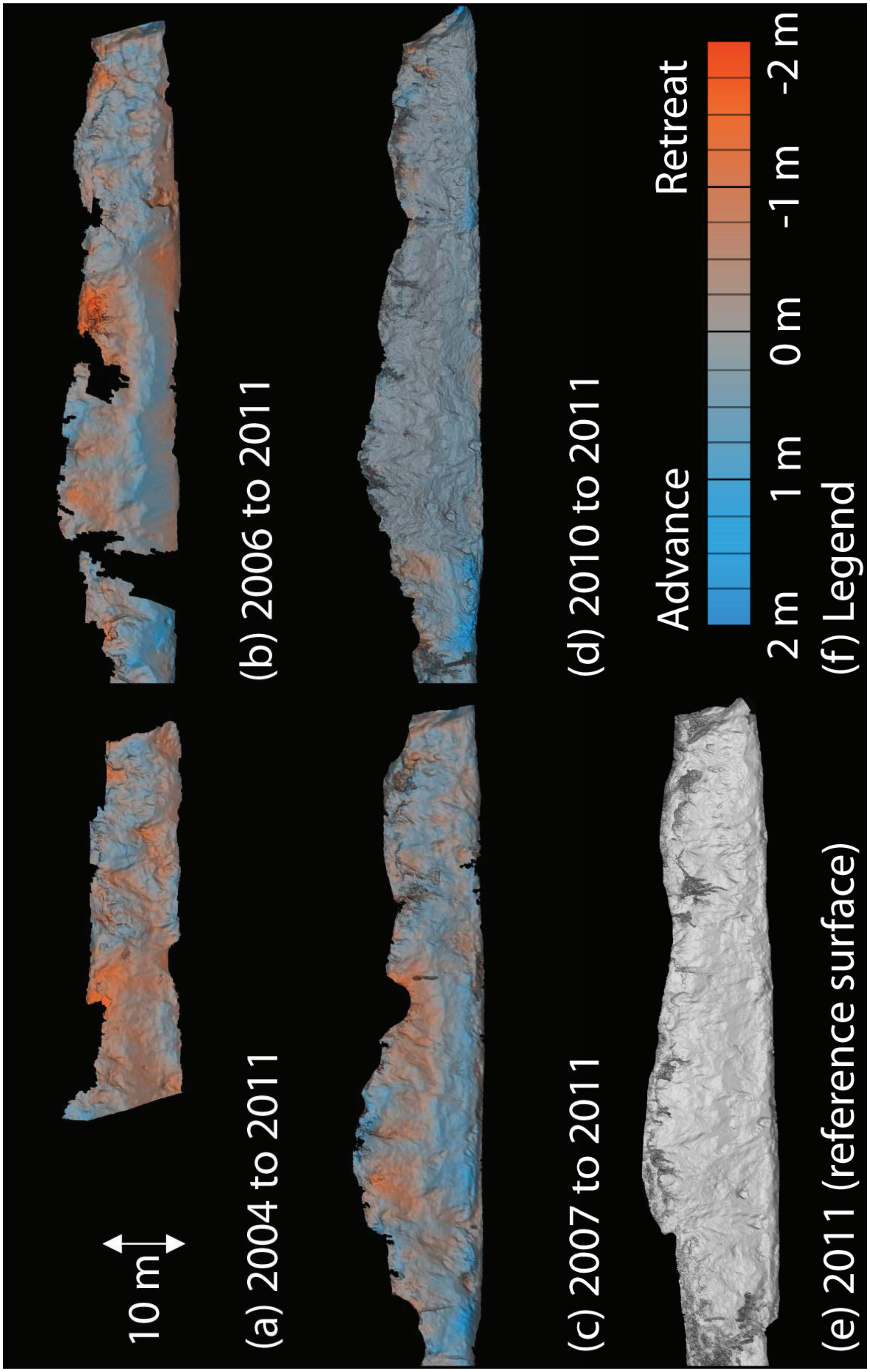


Figure 3.8: Change analysis between LiDAR surveys showing advance and retreat of the cliff face at the southern section (Northing 4,954,360 m to 4,954,440m).

3.2.5 Johnson Creek/Beverly Beach Conclusions

Several issues should be considered when completing landslide surveys, regardless of the technique used to acquire the data. The most important issues are:

(1) Survey network control should be set far outside the area of movement, or externally applied through GPS networks to ensure that control points are stable.

(2) Surveyors should collect additional data outside the area of interest. Not only can these data be used to validate the methodology and quality of data, but they also can be used to help verify the landslide extent and potential impacts on adjacent soils.

For the Johnson Creek landslide, the southern end shows more movement, overall recession, and erosion. The northern end shows slower movement in addition to slower erosion rates. This is consistent with previous findings of Priest et al. (*2008*), who show the inter-relationship of decreased stability of the southern part of the slide with increased erosion there.

Additional research will be needed to continue to monitor the site and determine long term change patterns. Further, the methodology will need to be refined by determining differential landslide movement at the top versus base of the slope. Perhaps this could be done through numerical modeling using several time-series laser scan datasets to calibrate and validate the model.

3.3 PIONEER MOUNTAIN – EDDYVILLE REALIGNMENT[3]

A realignment of the Pioneer Mountain to Eddyville section of US 20 is currently underway in the Coast Range of Oregon. Several superficial slope failures (Figures 3.9 and 3.10) have occurred on the steep 1.5:1 fill slopes and have been a significant cost and time delay to the project.

Straw wattles are common erosion control devices used to trap sediment during storm run-off, and have been installed on the steep slopes (1.5H:1V) for the US20 highway realignment project. Several shallow slope failures have occurred, often bracketed by straw wattles, which were hypothesized to contribute to the slope failures. To date, little is known about straw wattle placement and its effect on surficial slope stability. Prior studies have evaluated slope stability against slope height, slope angle, vegetation, rainfall, and other variables but have not assessed the influence of straw wattle placement on surficial stability.

[3] Additional soil testing and analysis related to the Eddyville site was funded through Granite Construction and have been compiled into a publication submitted for review:
Olsen, M.J., Rikli, A.M., Sillars, D.N., (Accepted with revisions). "Investigation of Straw Wattle Influence on Surficial Slope Stability," submitted to the Transportation Research Board (TRB) Annual Meeting, 2012.

To address this problem, TLS was used to produce 3D terrain models to analyze surficial slope stability of the site for the following purposes:

- Determine and verify fill slope inclination angles
- Measure embankment slope heights
- Observe straw wattle contour elevations and spacing
- Record failure depths to verify if superficial in nature, and
- Establish a baseline model of the slope to quantify (both spatially and temporarily) future slope failures through continued monitoring.

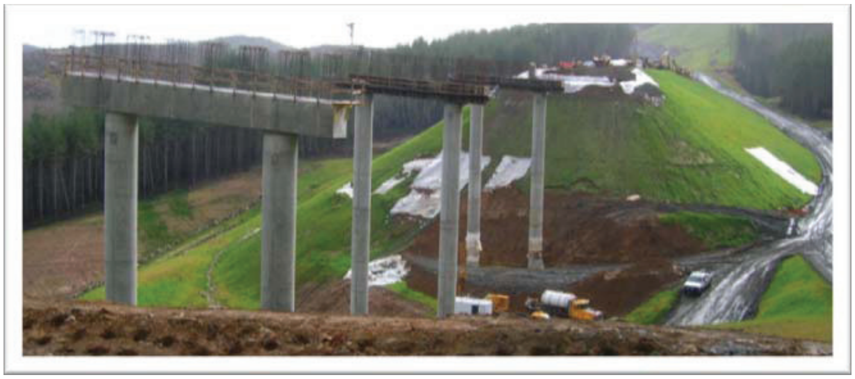


Figure 3.9: Construction of a bridge on the U.S. 20 Pioneer Mountain to Eddyville realignment project. The white tarps on the hillside cover the superficial slope failures, which have resulted in substantial cost and time delays to the project.

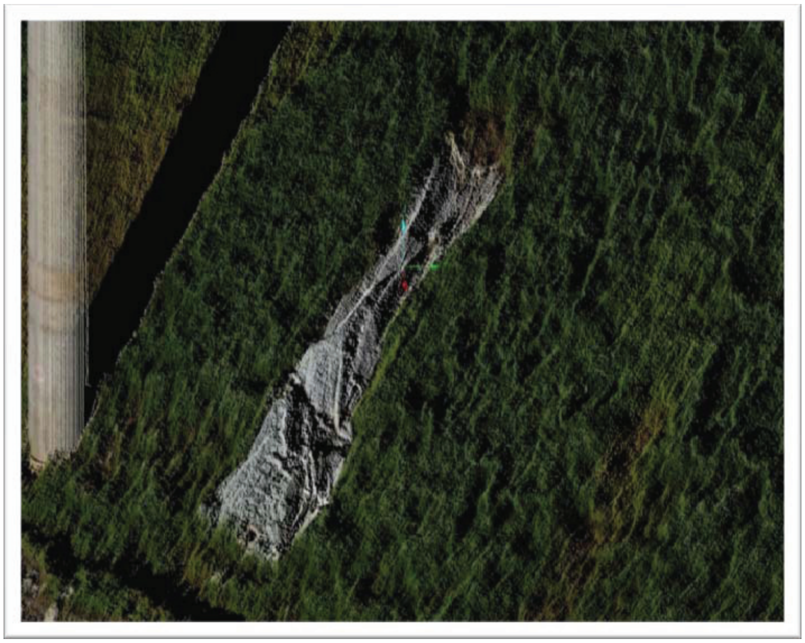


Figure 3.10: 3D laser scan point cloud of a shallow slope failure.

Three failure-laden fills were scanned with a 3D laser scanner. All fills showed a general slope angle of 1.5H:1V and heights of 30 to 40 meters, which are consistent with construction plans. Straw wattle spacing was confirmed to be 25 ft (7.6 m) and each wattle was installed at a consistent contour around the embankment slope. These elevations and spacing can be seen in Figure 3.11. Because of this, it was concluded that water was likely not moving laterally along the wattles and ponding in low spots. Thus, the hypothesis that water was backing up behind portions of the straw wattles at lower elevations was rejected.

The LiDAR scans also documented deformations caused by slope failures in each of the fills. These failures were, in general, a depth of less than 2 ft (0.60 m). This information, in conjunction with laboratory testing were used to back calculate soil strength parameters (ϕ'=33.7°, c' = 2.6 kPa). This cohesion intercept was much lower than that found in triaxial testing (c' = 18 kPa) for overall slope stability analyses. Using these strength parameters, it was determined that the slopes were at a factor of safety close to 1 under dry conditions. Hence, when seepage parallel to the face of the slope develops during rainfall, these shallow mud flows are likely to occur. While grass seed vegetation adds some strength at the surface, these slope failures bypass the roots completely.

Many failures observed were bracketed by straw wattles, however, from numerical modeling results using the geometry (slope and height) obtained from the scan data and strength parameters discussed previously, it did not appear that the straw wattles mechanically contributed to the failure of the slope. Future analysis will focus on hydrological impacts of the straw wattles and how they may alter the flow pattern to drive more water into the soil.

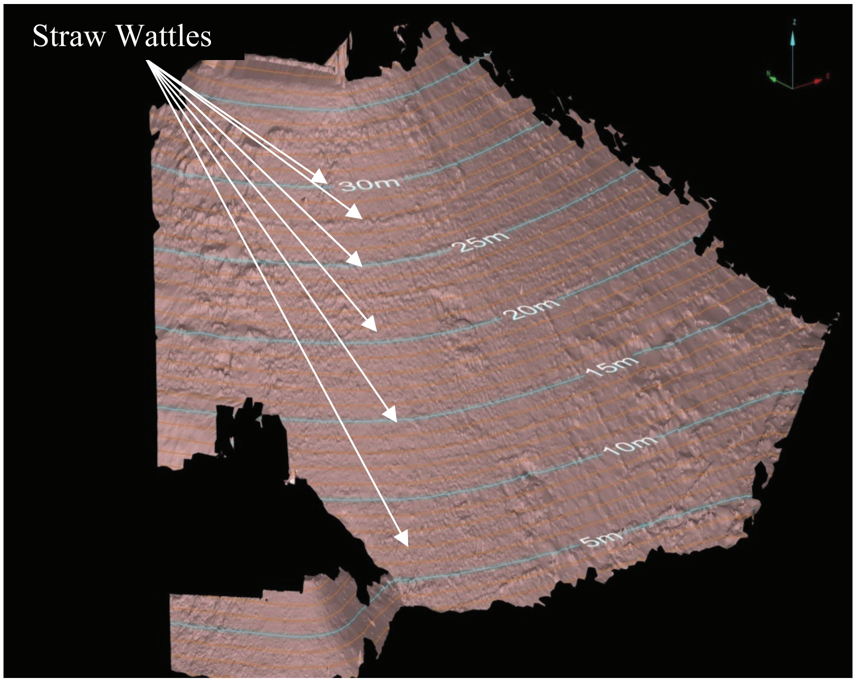


Figure 3.11: LiDAR-based DTM of fill slope, showing constant straw wattle elevation

In addition to examining the problems associated with the superficial slope failures, laser scanning was also completed on a large, deep slope failure that occurred on Cut 6 (Figure 3.12).

It is likely that if the failed soil is removed, the upper section of the slope will continue to slide. As a result, it was decided that the road would need to be shifted to the north (right in Figure 3.12). The scan data will be fully analyzed in a future study.

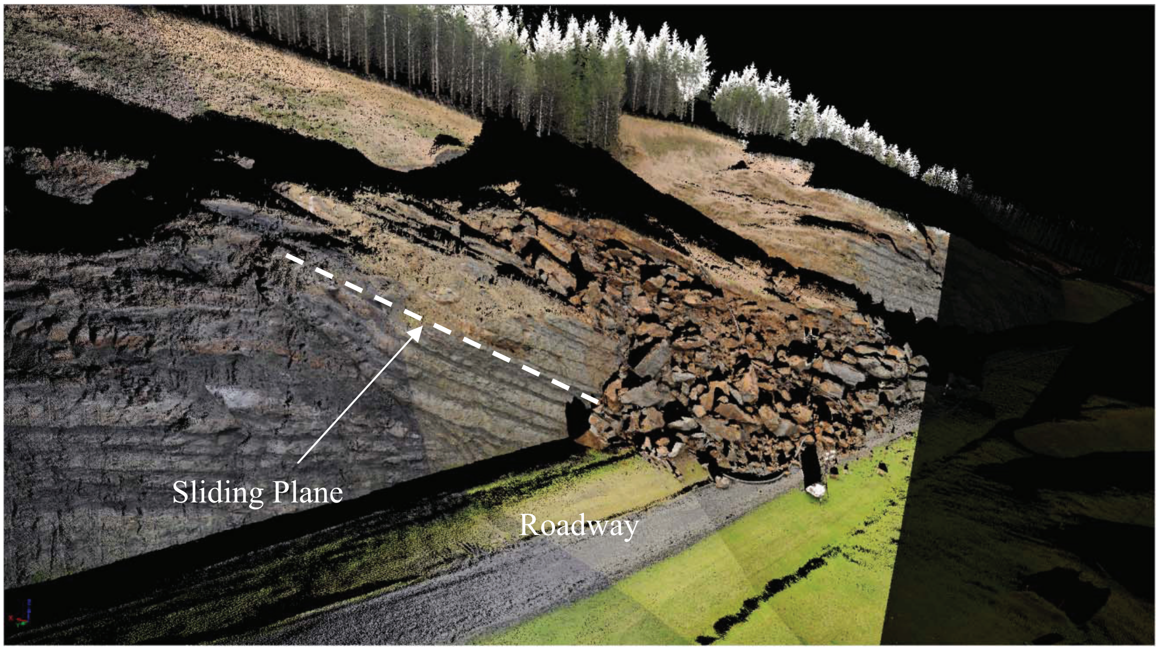


Figure 3.12: LiDAR scan of a deep slide on Cut 6, which occurred in January 2011

3.4 CENTRAL POINT MSE WALL EVALUATION

ODOT performed several scans of an older Mechanically Stabilized Earth (MSE) retaining wall in Central Point, Oregon, that was suspected to be moving because of observed cracking, concrete spalling, and other damage on the wall face. Change analysis was performed using the laser scan data provided by ODOT from surveys on December 2010, January 2011, and February 2011. No significant differences were observed between the December 2010 and January 2011 surveys. Figures 3.13 and 3.14 show the observed change between scans from January and February 2011. Note the "blue splotches" on the face of the wall. These were a result of a maintenance crew repairing sections of spalled concrete that occurred between the surveys as opposed to actual movement of the wall. While it does not appear that the wall is moving during this short time period, it is recommended that scans be completed on a semi-annual or annual basis. Also note that the error and sensitivity of the measurements between the surveys and modeling was estimated to be slightly less than 1 cm. Hence, very small movements (mm-level) could not be detected. This case study illustrates the potential application of laser scanning for MSE wall monitoring.

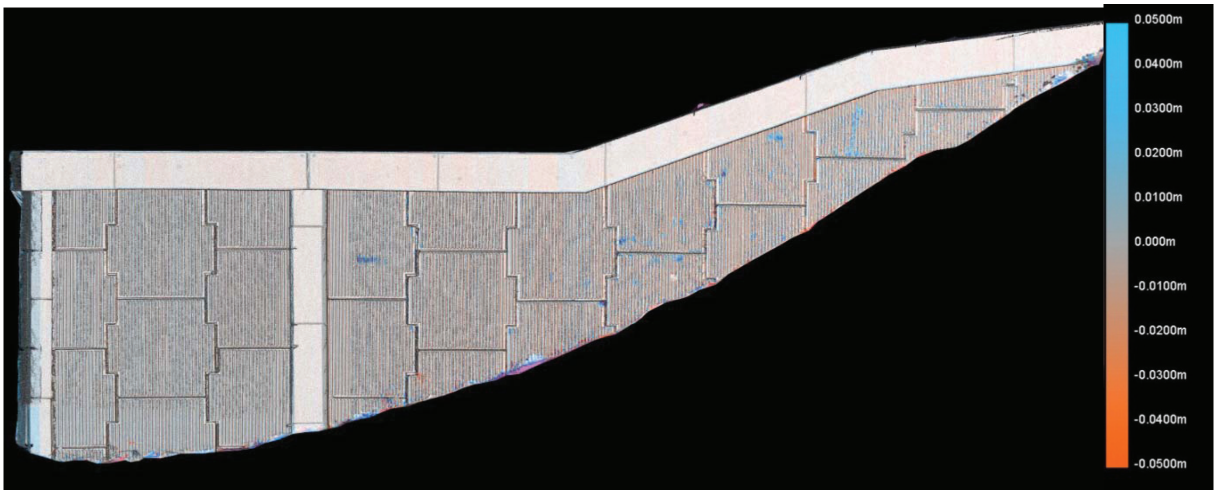


Figure 3.13: Comparison of MSE wall face between January and February 2011 surveys. Orange colors represent material loss, blue represents material gain.

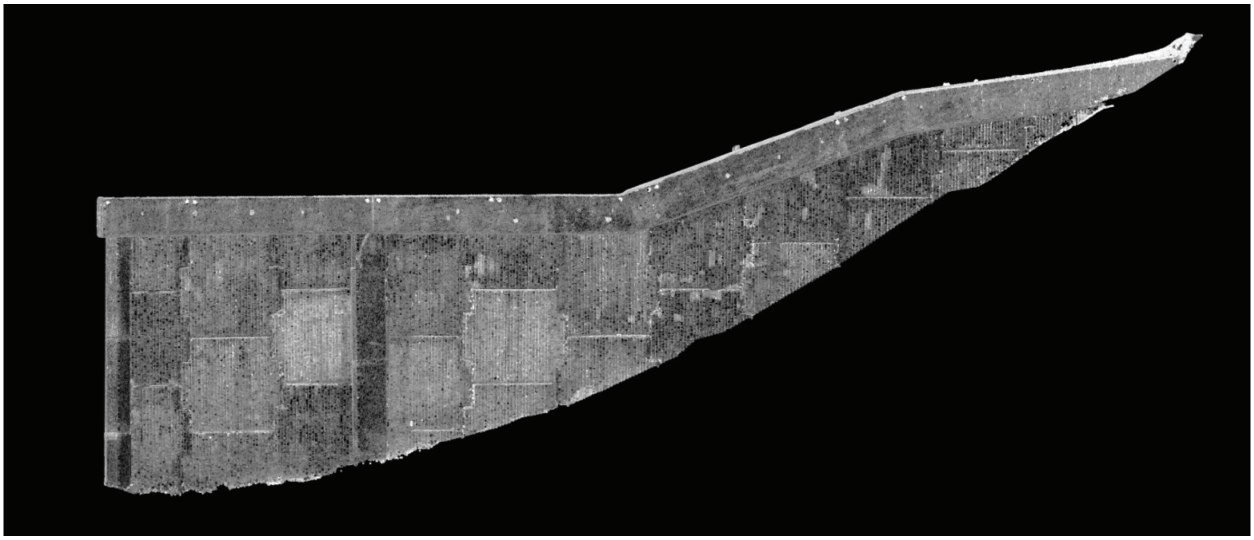


Figure 3.14: Intensity map of MSE wall face between January and February 2011 surveys. Note that the new concrete patches reflect light differently than the worn concrete.

4.0 TESTING

4.1 SUMMARY

Testing of the *Lichen* software was broken into three distinct phases:

I. Development of the code, initial testing and software bug correction were completed on campus at OSU.

II. Code refinement and workflow testing, as well as preliminary field trials and evaluation were performed at a field site on the OSU campus.

III. Final evaluation of the change analysis procedure and functionality was conducted under typical field conditions at the Johnson Creek landslide.

4.2 LAB TESTING

During initial development and refinement of *Lichen*, we implemented several lab tests in the Geomatics Lab located in Graf Hall on the OSU campus. Baseline scans were first collected in the lab room itself. Next, we strategically re-located objects around the room. Following a repeat scan, the effectiveness of *Lichen* to detect these objects was evaluated. These tests were conducted throughout the course of the entire project as code developments occurred. The focus of these tests was to:

1. Develop and improve the code to correctly geo-reference and sort scans,
2. Evaluate necessary computational resources and speed of the algorithm,
3. Verify accurate change detection in a controlled setting,
4. Simulate field tests by using previously acquired data, and
5. Evaluate and improve the workflow.

Numerous tests were conducted in the lab during code development to evaluate the accuracy and speed at which *Lichen* could perform change detection. Initial tests required 20-30 minutes of processing time for change detection. However, through refinements in code such as more efficient search algorithms, cut-off thresholds, improved data structures, the change detection time was reduced to one to two minutes for large (~1GB) scans. This shorter time interval of one or two minutes would have a minimal impact on most field applications where several minutes are required to setup the instrument and perform the scan for each position.

In addition to the laboratory scanning, we also used previously collected data from field sites to simulate a field experience (the only distinction being that the scan was not actually being collected at the time). These tests helped us ensure geo-referencing algorithms were operating properly and efficiently. They were also important to evaluate the suitability and scalability of the algorithms to field applications.

Fi to Fig illustrate the change detection abilities and limitations from a fixed position setup in the laboratory. For this test, several equipment cases on top of a desk were shifted following completion of the baseline scan. The middle case was rotated 90 degrees. The door was also rotated. A 5 mm thick poster board was moved from the floor and taped to the wall. (Note that it was not flush against the wall while on the floor). A 2 mm thick clipboard, a 1 mm thick piece of cardboard, and a 85 mm wide cardboard box were taped to the wall, as well. Fi shows the change analysis at a relatively fine scale. Note that most of the objects were detected well except for the 1 mm cardboard piece. There is also some noise around sharp edges due to scan artifacts termed mixed pixels, where laser beam strikes two surfaces close together so that it is unable to resolve them as distinct objects. Fig shows the change analysis focused on the larger movements of the cases. Note that due to the coloring scale, the objects representing smaller changes are not resolved, however there is less noise. Figure 4.3 is at the extreme limits of the measurement capabilities of the system. All objects, including the 1 mm cardboard piece are better defined; however, the measurement noise is also depicted as change.

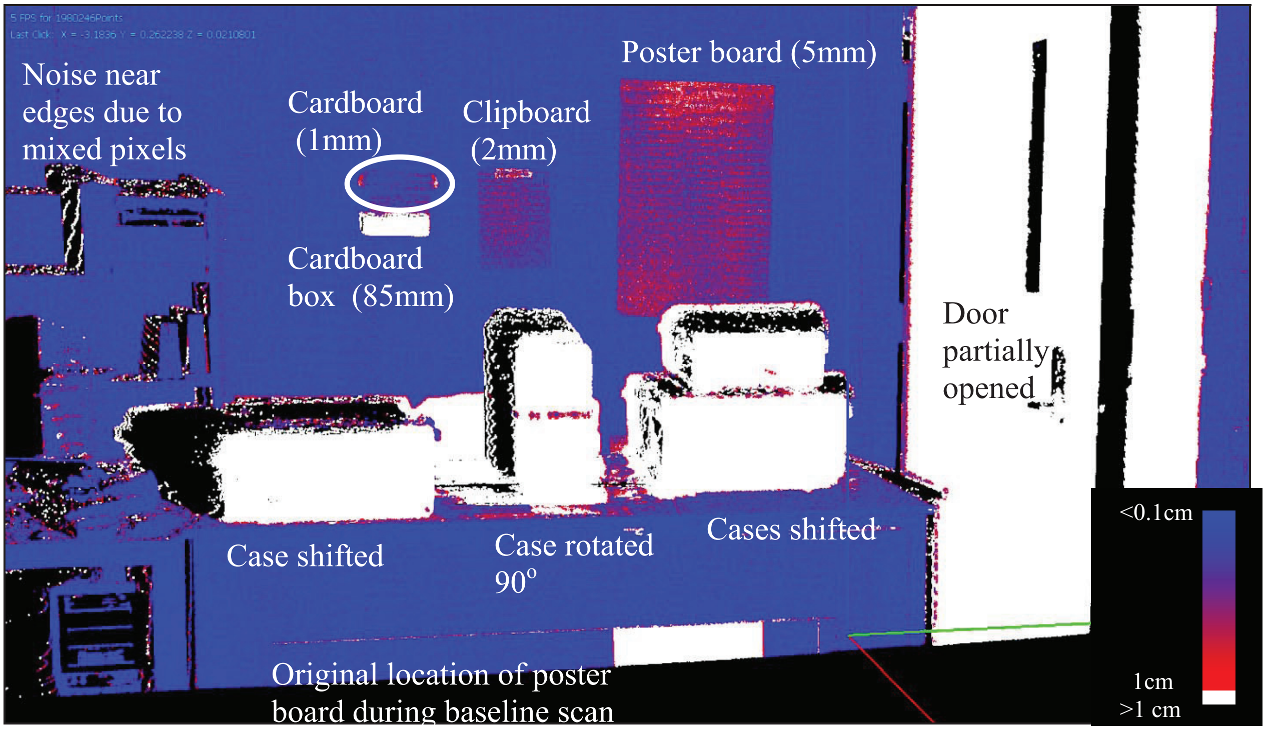


Figure 4.1: Change analysis performed on items shifted in laboratory with a fine scale.

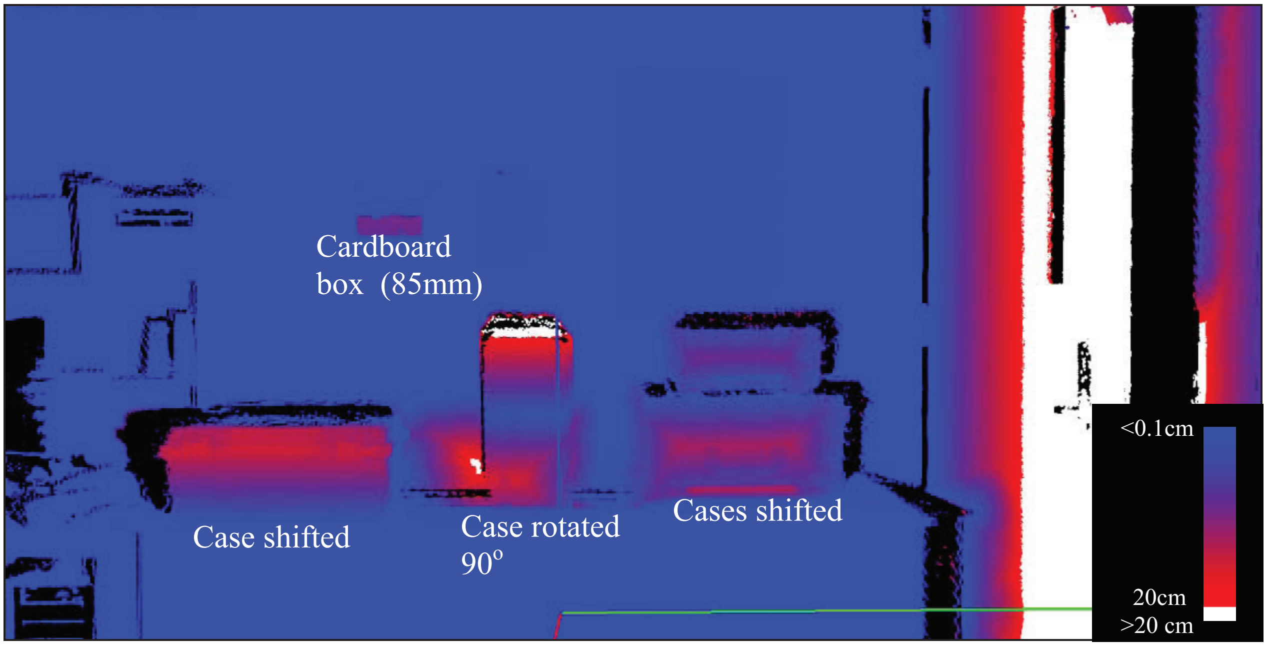


Figure 4.2: Change analysis performed on items shifted in laboratory for larger movements.

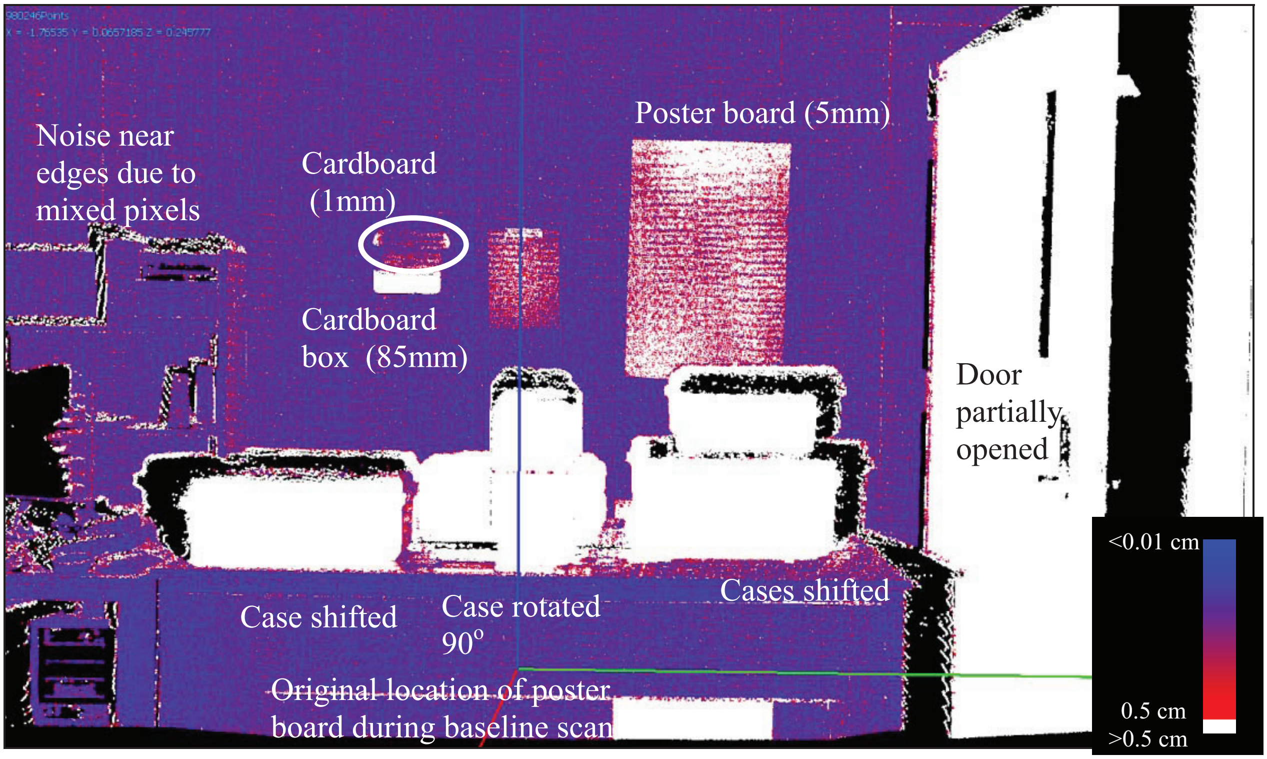


Figure 4.3: Change analysis performed on items shifted in laboratory with a very fine scale to highlight sensing limitations.
Note that while the smaller objects can be detected, there is more noise in the data from typical measurement error

4.3 LOCAL FIELD TESTING

To test the real-time change algorithm and program workflow more rigorously prior to full-scale site testing, we made use of an ongoing geotechnical research project being conducted at OSU's Hinsdale Wave Research Laboratory (Figu). This site is being used to test lateral load capacities of piles driven near slopes. This small-scale project offered an excellent opportunity to evaluate the change-detection algorithm under two potential field conditions. The first involves non-fixed scan position setups, most likely to be the case when monitoring highways, landslides, sea cliffs, and other large-scale objects where the necessary precision is within that offered by RTK GPS. The second method is fixed-point, time-series monitoring to enable finer-scale change detection by reducing uncertainty of the XYZ coordinates for the scanner origin. This would be used in situations requiring precision on the order of millimeters that may be achievable with total station control and long-term GPS monitoring.

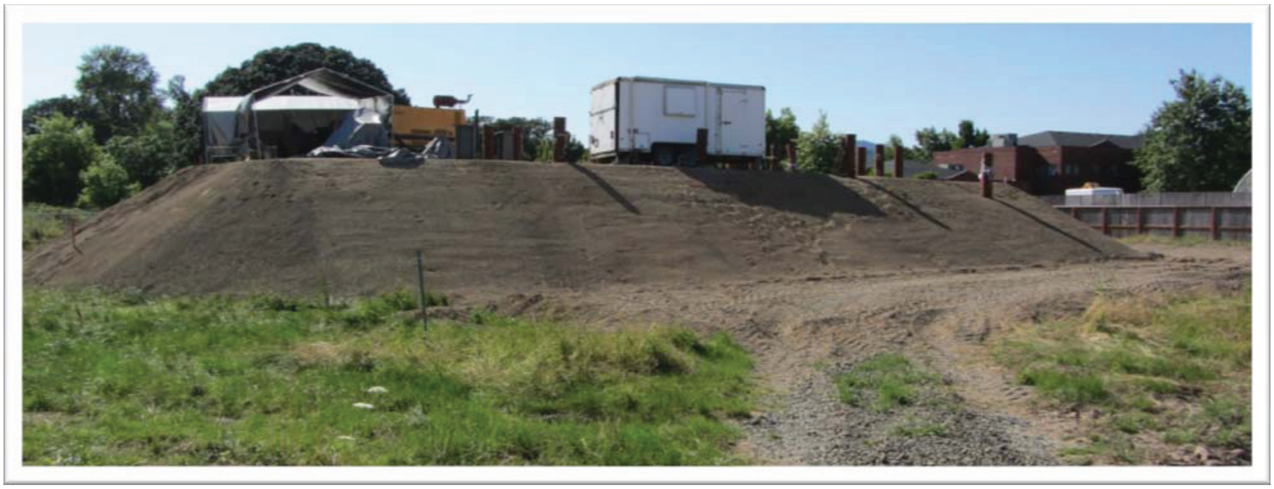


Figure 4.4: Photograph depicting the berm field site pre-testing.

Following completion of the berm and prior to any load testing, we traveled to the site and collected high-resolution LiDAR data along with RTK GPS for global positioning of the scan setups. We took several scans of the main face of the berm (east side) and one scan of the rear face (west side). Positioning of the scans still allowed alignment with the main face scan positions. Two separate triangulated models were generated, one for each face, to serve as the baseline models for change detection. Figure 4.5 depicts one of these models.

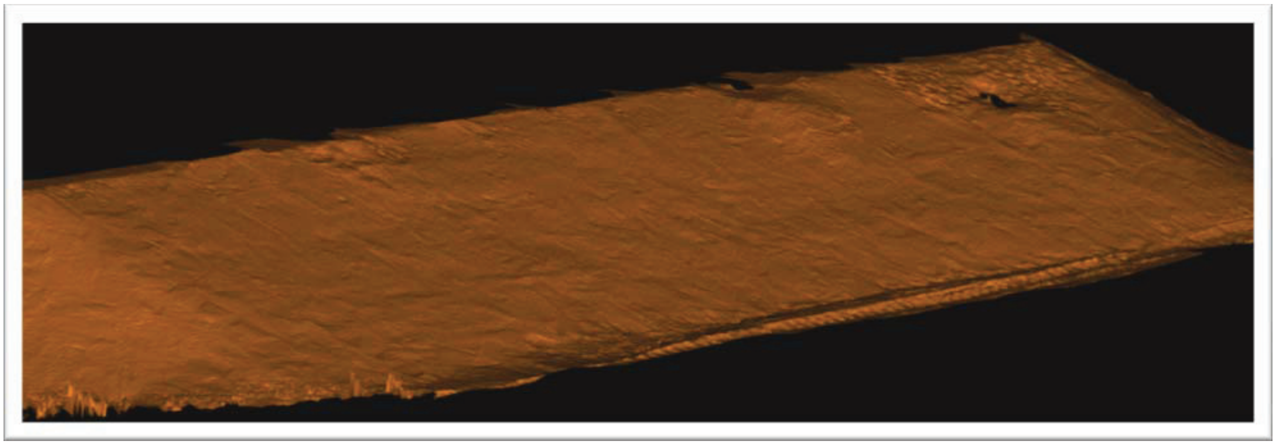
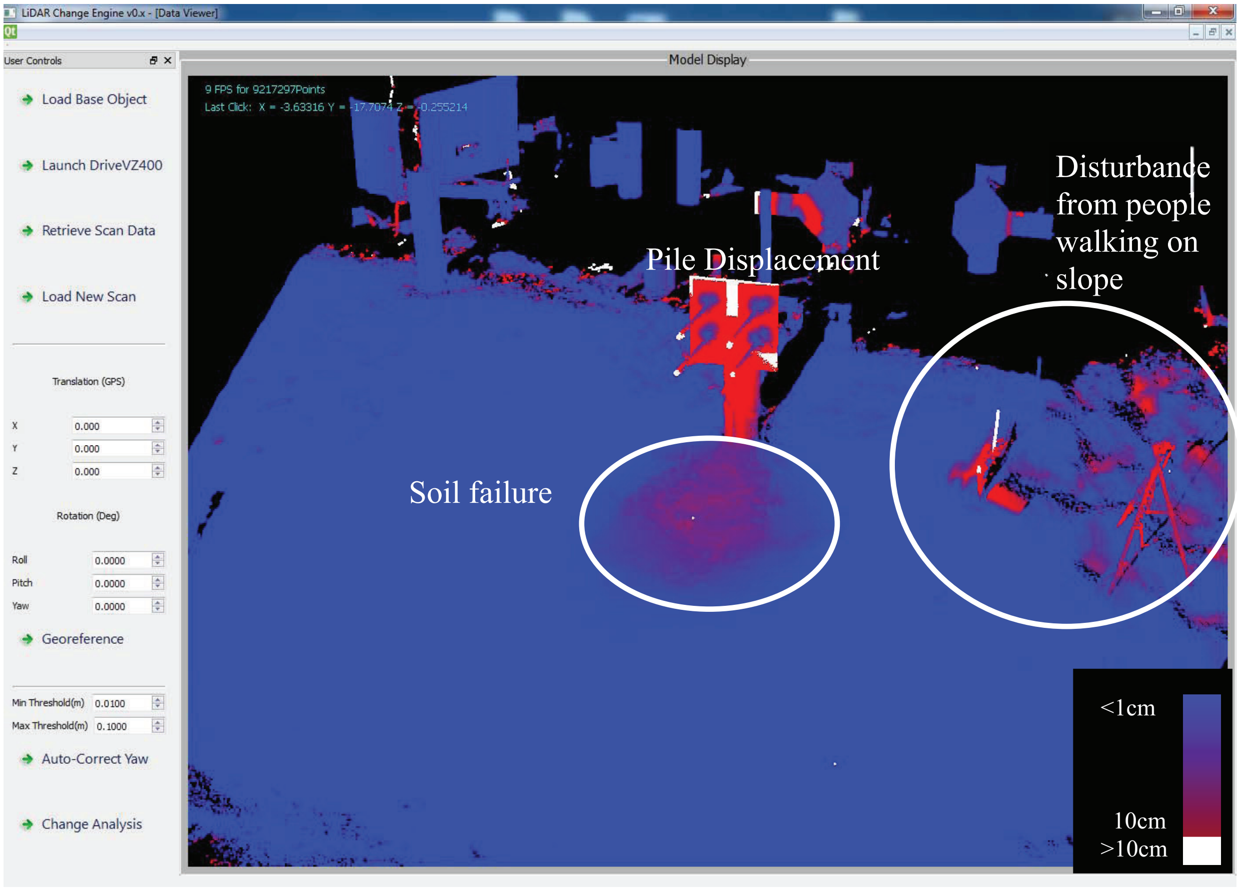


Figure 4.5: Rendering depicting the berm face as a triangulated model prior to testing.

Field testing of the algorithm was accomplished during several repeat visits from non-fixed positions on either side of the berm. GPS coordinates were obtained to geo-reference the new scans and compare to the baseline surface model to determine what change had occurred.

An additional test was performed over the course of a live geotechnical experiment in which one of the piles was undergoing lateral displacement expected to total approximately 10 inches (25 cm) along the actuator. The scanner was setup at one location to simulate monitoring from a fixed position. A baseline scan was taken prior to the start of displacement testing. Additional scans were performed throughout the course of the testing. GPS coordinates were obtained during each scan so that we could simultaneously evaluate change from a non-fixed position. No significant differences were observed between in change coloring in Lichen between the two methods. However, note that conditions for GPS positioning were very good, resulting in <1cm RMS.

Using *Lichen* for both methods (fixed and non-fixed), we were able to identify and display areas in the field site showing minimal, moderate, and heavy change over the course of the experiment. The resulting color-change map (Figures 4.6 and 4.7) corresponds with intuitively expected values. The actuator-displaced pile showed the maximum change, along with the reaction pile. In addition, areas that were frequently and heavily disturbed throughout the test, such as the tripod-mounted video camera and the information sign it was recording, displayed appropriate positional shifts. Likewise, areas such as the control trailer and piles not being tested showed no real change. Scaling in between these two extremes was the soil being displaced by the movement of the pile, which showed varying degrees of movement corresponding to individual positions along the displacement bulge. On-site researchers with a geotechnical engineering background were excited to see the visual analysis as the testing proceeded. They were highly interested in seeing the final deformation maps as well as having access to time-series data to show the progression of earth and pile movement throughout the test duration.



9 FPS for 9217297Points

Last Click: X = -3.63316 Y = -17.7074 Z = -0.255214

Translation (GPS)

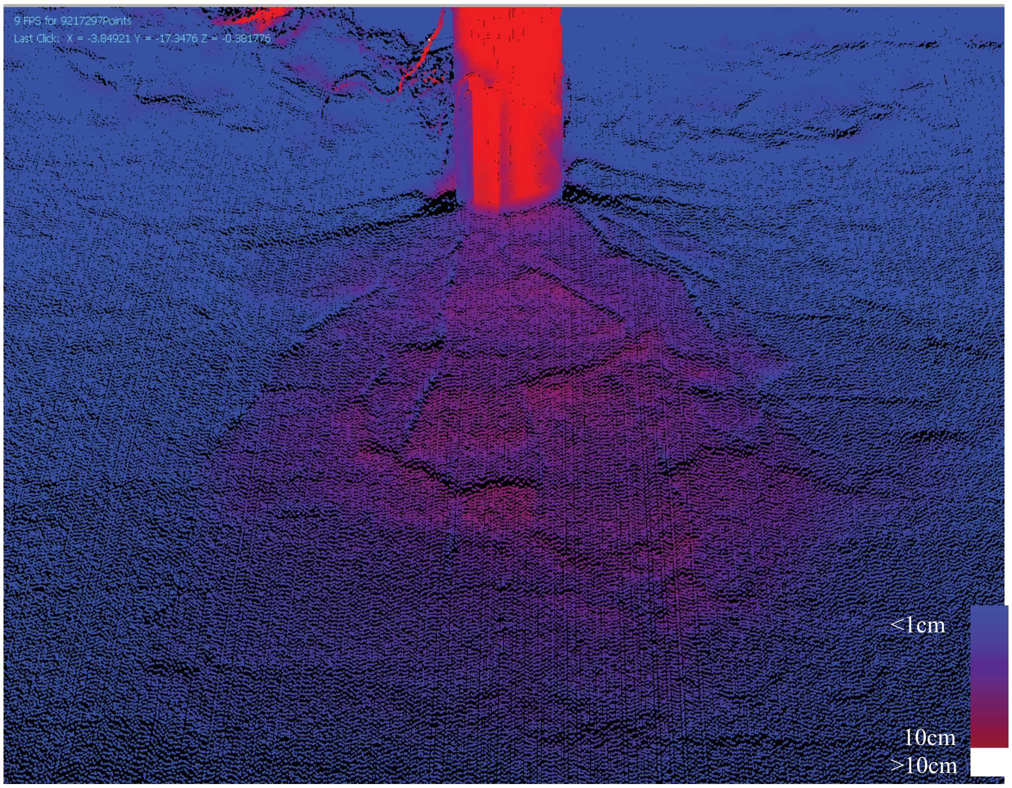
X	0.000
Y	0.000
Z	0.000

Rotation (Deg)

Roll	0.0000
Pitch	0.0000
Yaw	0.0000

Min Threshold(m)	0.0100
Max Threshold(m)	0.1000

Figure 4.6: Screenshot of *Lichen* being used for change analysis during the berm field test. This scan was taken at approximately nine inches (22.86) of actuator displacement. The white coloration indicates change beyond the maximum threshold of 10 cm, while blue coloring shows change less than 1cm. The color changes from blue to red as the amount of change increases.



```
9 FPS for 9217297Points
Last Click:  X = -3.84921 Y = -17.3476 Z = -0.381776
```

Figure 4.7: Close-up view of Figure 4.6 showing the soil bulge formed by the actuator driven pile's displacement.

Performing a more extensive test at a close, pseudo-controlled site allowed us to evaluate *Lichen*'s stability, workflow, and accuracy for anthropogenic deformations for which change is easier to understand than purely natural sources. For example, it was fairly obvious to detect the shift in position of manmade machinery, equipment trailers, and people over the elapsed months. Following each visit, stability issues in *Lichen* were addressed. Additional refinements were made to the GUI to improve the workflow.

The offsets calculated in Lichen were then verified against manual measurements made in the point cloud in software and showed good agreement (within 5%).

4.4 RESEARCH SITE FIELD TESTING

A stable, release-candidate version of *Lichen* was tested at the Johnson Creek landslide site to evaluate the real-time change analysis process under rigorous field conditions. Fresh scans were collected and compared to the previously created baseline scans (Chapter 3) directly in the field. These tests resulted in several successful walkthroughs of the complete procedure from several scan positions, including rapid conversion of the raw scans into binary files for processing using a laptop computer (Intel Core 2 Duo processor), which has weaker processing capabilities compared to desktop systems in order to maximize battery life and minimize physical size (Figure 4.8).

The only major tangible burden placed on users implementing *Lichen* in the field is the necessity of taking a mobile computing platform with them to perform the operations. This, however, is generally already required for several scan systems. The nature of the developed software adds only minimal additional time (<1 min. per scan) to the overall process of acquiring scans. Loading baseline and new scans is accomplished on the aforementioned laptop in <10 seconds, with conversion of the new scan via *BovConverter* occurring in <1 second. Figure 4.9 shows an example of such field results. At present, users must manually review their GPS/GNSS position and enter those values into *Lichen*. The time required for calculation of change scales upwards when smaller minimum cut-off thresholds are used for finer scale change observations. For example, if a user is interested in mm-level changes, the algorithm will require more time than for evaluating cm-level changes as it will need to compare each point against more points.

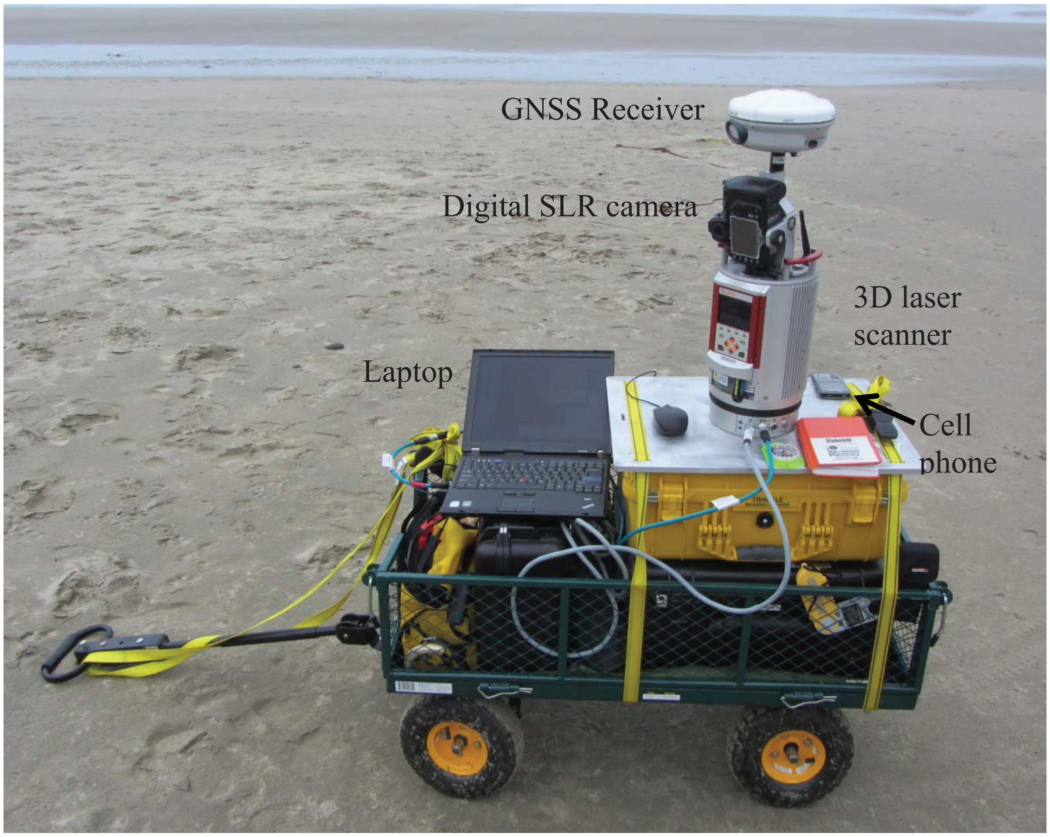
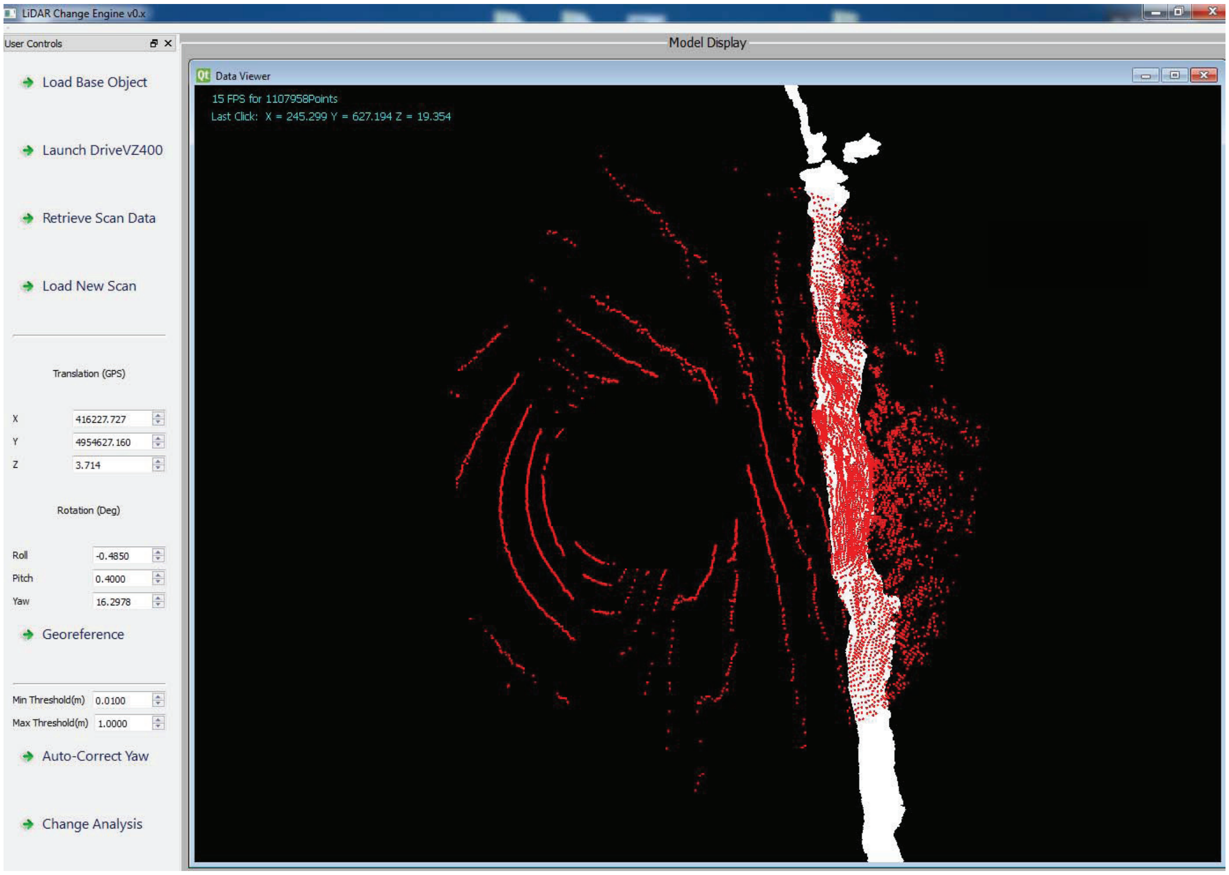


Figure 4.8: Photograph showing the TLS setup with a laptop to control the scanner and perform the real-time change analysis.



15 FPS for 1107958Points

Last Click: X = 245.299 Y = 627.194 Z = 19.354

Translation (GPS)

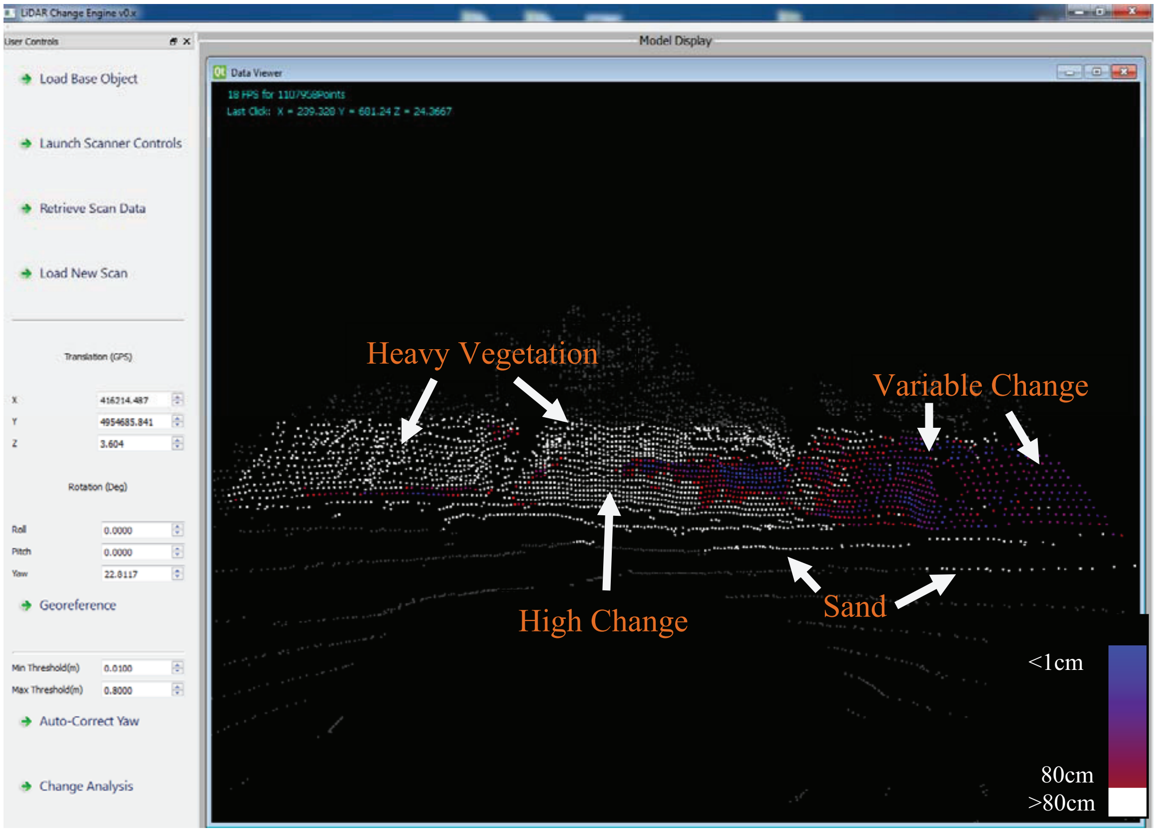
X	416227.727
Y	4954627.160
Z	3.714

Rotation (Deg)

Roll	-0.4850
Pitch	0.4000
Yaw	16.2978

Min Threshold(m)	0.0100
Max Threshold(m)	1.0000

Figure 4.9: Screenshot of the *Lichen* software operating at the Johnson Creek landslide. The baseline scan is visible as the white model. The red image is a monitor file that has been geo-referenced and filtered to a distance of 100meters from the scanner. Lichen has completed all steps prior to change analysis, including azimuth adjustment.



18 FPS for 1107958Points
Last Click: X = 239.328 Y = 681.24 Z = 24.3667

Translation (GPS)

X 416214.487
Y 4954685.841
Z 3.604

Rotation (Deg)

Roll 0.0000
Pitch 0.0000
Yaw 22.8117

Min Threshold(m) 0.0100
Max Threshold(m) 0.8000

Figure 4.10: Screenshot showing *Lichen* following a real-time change analysis at the Johnson Creek landslide on August 22, 2011.
The baseline model, collected in December 2010, has been hidden. Th new scan has been colored to represent the degree change detected (Figure 4.10). This is a good example of why human Q/C is essential. Being present at the location during the change analysis gives insights into what leads to the displayed results (e.g. being able to readily identify change resulting from vegetation shifts).

5.0 CONCLUSIONS/FUTURE WORK

5.1 ACCOMPLISHMENTS

The research group successfully developed an open-source based program to detect change between time-series terrestrial LiDAR scan data sets in real-time. *Lichen* was developed as a modular platform written in Qt/C++ with source code available. Currently, it has been designed as two main components: the main *Lichen* module and the scanner interface module for a Riegl VZ400 laser scanner. As additional manufacturers release SDKs to allow an end-user to interact with a laser scanner at the command level, additional scanners can be integrated into *Lichen*. If other users already have the ability to interface with their brand of scanner, then they can get output files compatible with *Lichen*'s workflow.

Additionally, *Lichen* is able to function over a wide range of hardware from workstation computers to mobile laptops. While there does exist an expected sharp decrease in performance as hardware platform capabilities decrease, it is not an undue burden on the operator to do field-based change analysis on lower-end hardware. Analysis and functionality using full-size files on mobile platforms imposes more limits on field workflow.

Lichen provides users the ability to utilize real-time change detection between scans taken at different times, while they are present in the field. There is no need to return to the office to perform post-processing on repeat scan data before any data analysis can commence. The process itself is fast and non-disruptive to current workflow methodology. In addition, *Lichen* users do not need to be experts in point cloud manipulation or processing as the software handles these technical processes in a semi-automated fashion. This enables LiDAR technicians to conduct evaluations in the field, potentially increasing project workflow.

Enabling real-time change detection in the field allows users to make judgments about the quality and relevance of the data they are collecting based upon project needs. Users making periodic scans of a region for monitoring purposes may have no need for amassing large quantities of redundant data which take up large amounts of storage space. When change of a magnitude significant to a user’s objectives is detected, it is made readily apparent and a high-resolution scan of the important regions may be performed. Users monitoring active processes may analyze their data in real-time to look for changes which may pose risks to human activities or to track progress on construction projects.

Lichen has been shown to be effective at detecting change larger than 5 centimeters in the field using the geo-referencing methodology described herein. Additional research is needed to determine if finer scale movements can be accurately detected. For scans acquired from the same location, *Lichen* was able to clearly detect changes as fine as 2 millimeters for fixed-position laboratory testing. However, note that these values will depend on the equipment used and materials scanned.

5.2 FUTURE DEVELOPMENTS

A wide range of improvements are planned for future versions of *Lichen* to improve software and hardware performance. The goal is to increase the usability, performance, and functionality of the program with each new iteration based upon developer and user feedback.

Planned updates will include continual improvement, as needed, to the way *Lichen* implements the automatic alignment of scan data as well as the way in which it establishes change between scans. Alignment between scans is crucial in providing the most accurate determination of what has actually changed. When improvements are discovered or deemed necessary, they will be tested and integrated into future versions of the software. We may opt to include additional change detection algorithms developed as they become available. Future research could also investigate change detection based upon 2D projections of the 3D point data. If possible, this would allow improved performance on lower-end platforms.

Additional optimizations aimed at improving performance will include enabling larger subset sampling of the full-resolution scan data to provide a more detailed change map of the region of interest. However, there is a fine balance between improved functionality and the resulting usability on hardware of different performance capabilities. It is also impossible to test functionality on every available configuration of hardware currently on the market. The most likely course of action is to develop guidelines to provide users a general set of settings that can provide expected performance for their general hardware category (e.g., workstation, desktop, laptop, other).

Other planned features include development of a tool that enables the user to click on a point to query the deformation at that location, export of color scale bars, controls for selecting the coloring scheme, and the ability to produce high resolution screen renderings directly in the program.

Future versions will also aim to include the capability of both automatic recommendation and selection of re-scan windows based upon *Lichen*'s analysis. This would be an alternative to user-definable or modifiable re-scan windows. This will add increased functionality to the software as well as improving the workflow while keeping the user as the quality control agent.

Future research will address quantifying achievable levels of change detection and comparing results to traditional measurements, which was beyond the scope of this project.

5.3 NEED FOR ADDITIONAL DATA

As the scope of this study was by necessity limited, future improvements to the software and user-operation would benefit greatly from the availability of additional data, particularly in highway and construction regions. Access to these areas is often restricted for both safety and construction work efficiency. We have made plans to test *Lichen* functionality at sites which stand to benefit from usage of spatial data and change detection in the future.

These areas could include critical infrastructure at risk of disruption due to earth movement, erosion, or man-made alterations. The ability to monitor these sites over time has the potential to provide advance warning to managers and policy makers so that appropriate actions to prevent or mitigate impacts may be taken.

Other data we believe to be important is on-site monitoring of projects such as earthwork movement, object stability, and progress monitoring. On-site data is necessary to check both the calibration of the software itself, as well as the correct method of scanner usage to achieve optimal results. For example, a large-scale earth clearing for a highway would not require positional accuracy beyond that of RTK GPS (1-3 cm), while monitoring bridge pillar stability may require the use of fixed control points (2-5 mm).

Having access to data in real-time during these processes has the potential to increase project efficiency and reduce construction costs. TLS systems can collect enormous amounts of data faster than traditional survey methods, and *Lichen* can process and deliver analysis results to the user in minutes. As development of *Lichen* continues, additional data sets will be used to enhance the efficiency and functionality of the real-time change analysis and expand the level of detail users can extract in the field.

It is important to note that for change detection, baseline models are needed. Hence, it is imperative to have repeat efforts to acquire accurate laser scan surveys at critical sites.

5.4 IMPROVEMENTS IN SCANNER SYSTEMS

The technology behind terrestrial laser scanners and LiDAR operation in general is rapidly advancing, providing faster and more accurate scanners with increasing range and versatility in operations. As the technology advances, the availability of hardware and software platforms becomes increasingly affordable to potential users. In addition, the data that is gathered from laser scanning projects is being increasingly made available for purposes beyond the scope of the original mission that collected it.

Consistent with the vision driving *Lichen*'s development is the newly released data standard "E57" which will bring much needed interoperability between the many different brands of scanners currently offered. The E57 standard does not replace a manufacturer's optimized raw data format, but will facilitate the transfer of data sets between different software packages without losing critical data to do so (*Huber 2011*). As different manufacturers adopt this standard, it will help reduce the barriers faced by users of LiDAR technology.

5.5 COLLABORATION

This project involved collaboration between students and faculty at OSU, ODOT Geometronics, and DOGAMI. This research has the potential to improve cross-disciplinary training at ODOT. For example, use of TLS for geotechnical studies performed by ODOT could create interdisciplinary teams to perform field work simultaneously, enabling surveyors from ODOT Geometronics to understand the needs of the geotechnical groups for data acquisition. Conversely, the geotechnical group will obtain a better understanding of how such data are collected and potential limitations. Currently, ODOT Geometronics collects the data, processes it, and turns it over as a finalized digital terrain model to the geotechnical group who then use it in the analysis. Upon viewing the wealth of data available from TLS, we hope that divisions will begin to see the value of working with these higher-resolution datasets offered by 3D laser scanning.

5.6 IMPLEMENTATION PLAN AND USE

5.6.1 Data Transfer

The insights obtained from this project have been, and continue to be, incorporated into several student PhD and MS theses for initial dissemination. Several journal and conference publications are also being prepared as part of this work.

A critical component to the long-term success of this project is transferring the program and results to ODOT and other agencies. We are currently working with ODOT Geometronics to provide them with both the data and program. We are planning on presenting this material during training sessions such as the annual ODOT-sponsored surveyor conference, so that ODOT's personnel become familiar with the technology and its potential applications. Potentially, real-time change detection can also be implemented throughout ODOT's construction and operations divisions. For example, for earthwork movement on a project, the real-time change analysis can be used as a tool to show construction managers and quality control units how much earth has been moved by comparison to baseline digital terrain models (DTMs). Additionally, the amount of cut/fill left to be completed could be shown more visually and accurately by using a 3D design DTM as the reference model. This would allow the quality control checks to be done in the field instead of requiring the data to be processed and quantities calculated in the office. Thus, any necessary fixes could be done while the equipment is still at the site.

Ultimately, a primary goal of ODOT Geometronics is to transition to full 3D design and machine automation. By increasing exposure of TLS to ODOT's engineers and construction crews, this will help transition them to working with 3D datasets and within 3D environments. It will also advance 3D scanning technology and move it closer to being usable in machine automation because of the developed real-time analysis procedures.

The *Lichen* program (Chapters 2 and 4) will also soon be provided as a freeware application so that users can download and use it. As other manufacturers provide an interface library and equipment, only the data acquisition portion of the program will need to be modified to accept the new scanner system, and most of the overall methodology and code will remain the same. Thus, the program will be widely applicable for many uses.

The data and findings generated from this research will be publicly available by dissemination to ODOT and DOGAMI.

6.0 REFERENCES

Alba, M., L. Longoni, M. Papini, F. Roncoroni, and M. Sacioni. Feasibility and problems of TLS in modeling rock faces for hazard mapping. *Int. Arch. of Photogrammetry and Remote Sensing,* ISPRS, Vol. 36, Part 33/W19, 2005.

Alba, M., F. Roncoroni, and M. Scaioni. Application of TLS for Change Detection in Rock Faces. *Proc., Laserscanning09,* ISPRS, France, 2009, pp. 99-104.

Alba, M., and M. Scaioni. Automatic detection of changes and deformations in rock faces by terrestrial laser scanning. Proceedings of Commission V Symposium, ISPRS, UK, 2010, pp. 11-16.

Barber, D., J. Mills, and S. Smith-Voysey. Geometric validation of a ground-based mobile laser scanning system. *Journal of Photogrammetry and Remote Sensing*, ISPRS, No. 63(1), 2008, pp. 128-141.

Bernardini F., and H. Rushmeier. The 3D model acquisition pipeline. *Computer Graphics Forum*, Vol. 21(2), 2002, pp. 149–172.

Besl, P.J., and N.D. McKay. A method for registration of 3D shapes. *IEEE PAMI*, Vol. 14(2), 1992, pp. 239-256.

Burns, W.J., and I.A. Madin. Protocol for inventory mapping of landslide deposits from light detection and ranging (LiDAR) imagery. *DOGAMI Special Paper,* No. 42, 2009.

Butkiewicz, T., R. Chang, Z. Wartell, and W. Ribarsky. Visual analysis for live LiDAR battlefield change detection. Proc. Defense and Security 2008*,* Vol. 6983*,* SPIE, FL, 2008.

Collins, B., and N. Sitar. Application of high resolution 3d laser scanning to slope stability studies. Proc., 39th Annual Symposium on Engineering Geology and Geotechnical Eng.*,* Butte, MT, 2004, pp. 79–92.

Collins, B., R. Kayen, T. Reiss, and N. Sitar. Terrestrial LiDAR investigation of the December 2003 and January 2007 activations of the Northridge Bluff Landslide, Daly City, California. *USGS, Open File Report,* No. 2007-1079, 2007.

Collins, B., and N. Sitar. Stability of steep slopes in cemented sands. *Journal of Geotechnical and Geoenvironmental Engineering,* ASCE, Vol. 137(1), 2011, pp. 43-51.

Dunning, S.A., N.J. Rosser, and C.I. Massey. The integration of terrestrial laser scanning and numerical modeling in landslide investigations, *Quarterly Journal of Engineering Geology and Hydrogeology,* Vol. 43(2), 2010, pp. 233-247.

Ghilani, C.D. *Adjustment Computations: Spatial Data Analysis,* 5th Ed., Wiley, NJ, 2010.

Girardeau-Montaut, D., M. Roux, R. Marc, and G. Thibault. Proceedings of ISPRS Workshop Laser scanning 2005, ISPRS Archives, Netherlands, 2005, pp. 30-35.

Glennie, C. Kinematic Terrestrial Light-Detection and Ranging System for Scanning. *Transportation Research Record,* No. 2105, 2009, pp. 135-141.

Huber, D. The ASTM E57 file format for 3D imaging data exchange. Proceedings of SPIE 7864, 78640A, doi: 10.1117/12.876555, 2011.

Jaboyedoff, M, D. Demers, J. Locat, A. Locat, P. Locat, T. Oppikofer, D. Robitaille, and D. Turmel. Use of terrestrial laser scanning for the characterization of retrogressive landslides in sensitive clay and rotational landslides in river banks. *Canadian Geotechnical Journal.,* NRC, Vol. 46(12), 2009, pp. 1379-1390.

Jaboyedoff, M, T. Oppikofer, A. Abellan, M.H. Derron, A. Loye, R. Metzger, and A. Pedrazzini. Use of LiDAR in landslide investigations: a review. *Natural Hazards*, doi:10.1007/s11069-010-9634-2, 2010.

Kayen, R., R.T. Pack, J. Bay, S. Sugimoto, and H. Tanaka. Terrestrial-LiDAR visualization of surface and structural deformations of the 2004 Nigata Ken Chuetsu Japan, Earthquake. *Earthquake Spectra,* EERI, 22(S1), S147-S162, 2006.

Kayen, R., J.P. Stewart, and B. Collins. Recent advances in terrestrial LiDAR applications in geotechnical earthquake engineering. Proceedings of 5th Int. Conf. on Recent Advances in Geotech. Earthquake Engineering and Soil Dynamics, ASCE, 2010.

Kemeny, J., B. Norton, J. Handy, and J. Donovan. *Three-dimensional digital imaging for the identification, evaluation and management of unstable highway slopes.* Final Report for Highway IDEA Project 119, TRB NAS, 2008.

Kemeny, J., and A.K. Turner. *Ground-based LiDAR Rock slope mapping and assessment*. Publication No. FHWA-CFL/TD-08-006, 2008.

Landslide Technology. Geotechnical investigation Johnson Creek Landslide, Lincoln County, Oregon. *DOGAMI Open-File Report,* No. O-04-05, 2004, p. 115.

Lim, M., D.N. Petley, N.J. Rosser, R.J. Allison, and A.J. Long, A.J. Combined digital photogrammetry and time-of-flight laser scanning for monitoring cliff evolution. *The Photogrammetric Record*, Vol. 20(110), 2005, pp. 109-129.

Olsen, M.J., E. Johnstone, A.P. Young, T.J. Hsieh, S.A. Ashford, N. Driscoll, and F. Kuester. Rapid Response to Sea cliff Erosion in San Diego County using Terrestrial LiDAR. *Proceedings of Sol. to Coastal Disasters Conference,* ASCE, Vol. 573, 2008.

Olsen, M.J., E. Johnstone, N. Driscoll, S.A. Ashford, and F. Kuester. Terrestrial laser scanning of extended cliff sections in dynamic environments: a parameter analysis. *Journal of Surveying Engineering*, ASCE, Vol. 135(4), 2009, pp. 161-169.

Olsen, M.J., and J. Donahue. A wave of new information: LiDAR investigations of the 2009 Samoan tsunami. *Proceedings Sol. to Coastal Disasters Conference,* ASCE, 2011, pp. 321-330.

Olsen, M.J., E. Johnstone,, F. Kuester, S.A. Ashford, and N. Driscoll. New automated point-cloud alignment for ground based LiDAR data of long coastal sections. *Journal of Surveying Engineering*, ASCE, Vol. 137(1), 2011, pp. 14-25.

Olsen, M.J., A.M. Rikli, and D.N. Sillars. Investigation of Straw Wattle Influence on Surficial Slope Stability. Transportation Research Board (TRB) Annual meeting, 2012.

Olsen, M.J., J.C. Allan, and G.R. Priest. Johnson Creek landslide movement and erosion quantification through 3D laser scanning. ASCE Geoinstitute GeoCongress 2012, Oakland, CA, 2012.

Priest, G.R., J.C. Allan, A.R. Niem, W.A. Niem, and S.E. Dickenson. Johnson Creek Landslide Research Project, Lincoln County, Oregon. *DOGAMI Special Paper,* No. 40, 2008.

Rosser, N.J., D.N. Petley, M. Lim, S.A. Dunning, and R.J. Allison. Terrestrial laser scanning for monitoring the process of hard rock coastal cliff erosion. *Quarterly Journal of Engineering, Geology and Hydrology*, Vol. 38, 2005, pp. 363-375.

Samarasekera, S. Into the Mainstream. Geospatial Intelligence Forum: LiDAR future, *http://www.kmimediagroup.com/files/GIF%209-1_FINAL.pdf* Accessed February 2011.

Schultz, W.H. Landslide susceptibility revealed by LiDAR imagery and historical records. *USGS Open File Report,* No. 2005-1405, 2005.

Schulz, W.H., and W.L. Ellis. Preliminary results of subsurface exploration and monitoring at the Johnson Creek landslide, Lincoln County, Oregon. *USGS Open-File Report*, No. 2007-1127, 2007.

Silvia, E.P., and M.J. Olsen. To level or not to level: Laser scanner inclination sensor accuracy evaluation. *ASCE Journal of Surveying Engineering*, undated.

Singh, R. Engineering Automation: Key concepts for a 25 year time horizon. 2008. http://www.oregon.gov/ODOT/HWY/GEOMETRONICS/docs/EngAutoKeyConcepts.pdf. Accessed March 2012.

Teza, G., A. Galgaro, N. Zaltron, and R. Genevois. Terrestrial laser scanner to detect landslide displacement fields: a new approach. *International Journal Remote Sensing,* Vol. 28(16), 2007, pp. 3425-3446.

Tobias, Lori. Pacific Ocean threatens to gobble up Oregon beach towns, The Oregonian, 2010. http://www.oregonlive.com/news/index.ssf/2010/01/pacific_ocean_threatening_to_g.html. Accessed January 31, 2010.

Trinder, J.C., and M. Salah. Airborne LiDAR as a Tool for Disaster Monitoring and Management. Proceedings of GeoInformation For Disaster Management*,* ISPRS, Turkey, 2011.

Turner, A.K., J. Kemeny, S. Slob, and R. Hack. Evaluation and management of unstable rock slopes by 3-d laser scanning. Proceedings of IAEG, 2006.

Vosselman, G., and H.G. Maas. *.Airborne and Terrestrial Laser Scanning.* Whittles Publishing, Scotland, 2010.

Vu, T.T., M. Matsuoka, and F. Yamazaki. LiDAR-based change detection of buildings in dense urban areas. Proceedings of Geoscience and Remote Sensing Symposium, IGARSS, IEEE International, AK, 2004, pp. 3413-3416.

Walter, V. Object-based classification of remote sensing data for change detection. *ISPRS Journal of Photogrammetry Remote Sensing,* Vol. 58, 2004, pp. 225-238.

Young, A.P. and S.A. Ashford. Quantifying sub-regional sea cliff erosion using mobile terrestrial LiDAR. *Shore and Beach*, Vol. 75(3), 2007, pp. 38-43.

Young, A.P., R.E. Flick, R. Gutierrez, and R.T. Guza. Comparison of Short-Term Seacliff Retreat Measurement Methods in Del Mar, California. *Geomorphology*, Vol. 112, 2009, pp. 318-323

Young, A.P., M.J. Olsen, N. Driscoll, R. Gutierrez, R.T. Guza, R.E. Flick, E. Johnstone, and F. Kuester. Comparison of airborne and terrestrial LiDAR estimates of sea cliff erosion in Southern California. *Journal of Photogrammetric Engineering and Remote Sensing*, ASPRS, Vol. 76(4), 2010, pp. 421-427.

Zhang Z. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, Vol. 13(2), 1994, pp. 119–152.

APPENDIX A

LICHEN V1.0 USER MANUAL

Programmed by: Michael J. Olsen, Alfred Flammang, Shawn Butcher, Rebecca Pankow, and Andrew Johnson (Oregon State University)

What is it?

Lichen (**Li**DAR **Ch**ange **En**gine V1.0) is a software program designed to enable real-time change detection between terrestrial laser scan data sets. The software was designed to fulfill two important purposes. First, it enables users to identify areas of high and low change or potentially significant user interest while collecting data in the field. If employed as a means to monitor sensitive areas, this program provides a fast, semi-automated means to analyze them, or enable analysis of areas that are not easily traversed manually. Second, the software may help reduce data redundancy for a technology which generates and requires a large volume of storage space. Users may determine thresholds appropriate for their application and when an area exhibits change above this threshold, only that region needs to be re-scanned and updated on the storage devices.

How does it work?

Operators enter the field with previously generated digital models of the regions of interest which serve as baseline scans. A new, lower-resolution scan is performed and then geo-referenced in the same coordinate system as the previous model. Once completed, the new scan will be compared to the pre-existing model to look for both change and the magnitude of change that has occurred. This information is displayed in a GUI (Graphical User Interface, see Figure A-1) to allow the operators to visually inspect the data to make a determination as to whether more detailed scans will be necessary at that position.

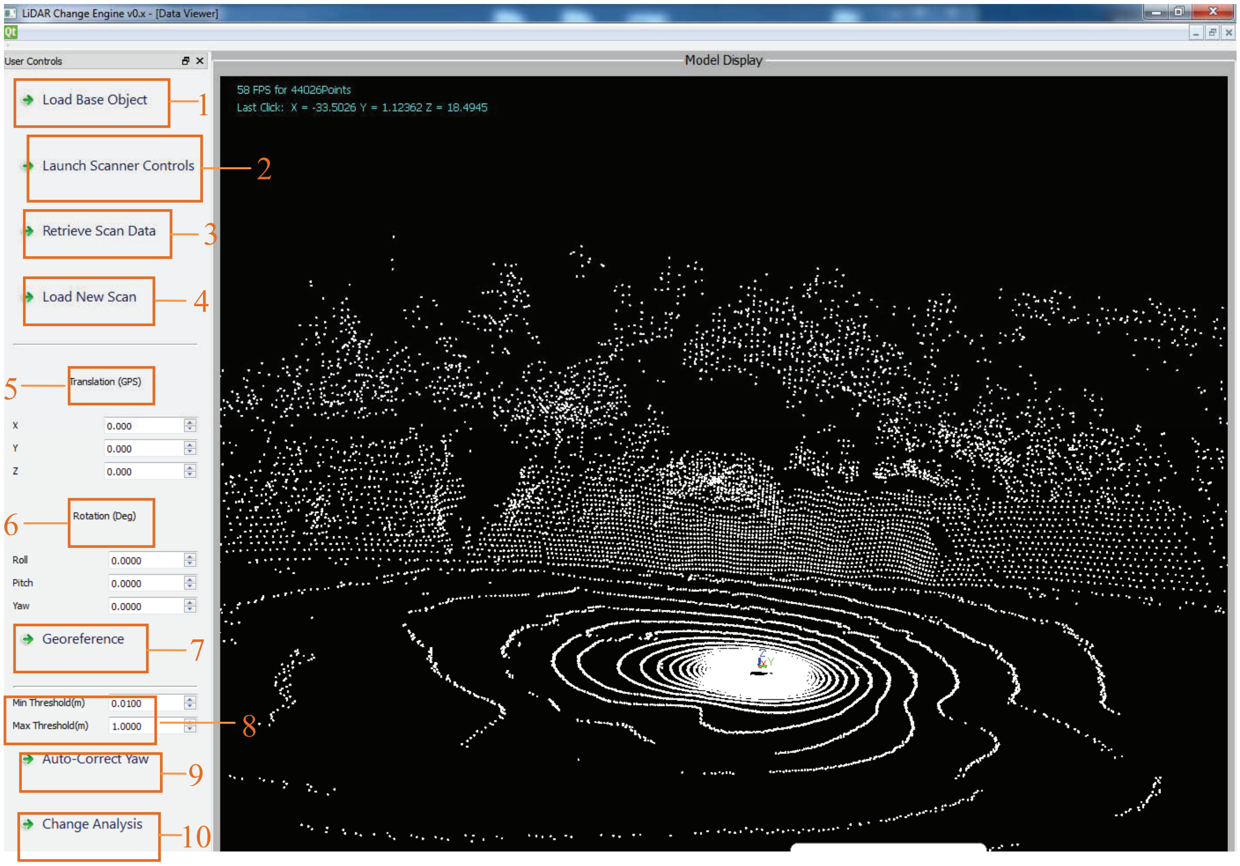
Prior to running the program

The user will collect and process baseline scans of the site following conventional methodologies. These baseline scans will need to be converted to the (***.bov**) format using either the PTX converter program (Appendix B) or the *BovConverter* program (Appendix C) for *txt*, *obj*, or Riegl***(*.rxp***)files.

Launching the program

To run the program, locate the directory in which the program is installed, such as "*C:/Program Files(x86)/Lichen"* and double click the executable *Lichen.exe*. This will launch the program's GUI from which all functions of the program may be accessed. The user interface has been laid out in a logical manner that follows the process a user will go through in order to implement the software for real-time change analysis.

GUI ELEMENTS



LiDAR Change Engine v0.x - [Data Viewer]

User Controls

Model Display

58 FPS for 44026Points

Last Click: X = -33.5026 Y = 1.12362 Z = 18.4945

X 0.000

Y 0.000

Z 0.000

Roll 0.0000

Pitch 0.0000

Yaw 0.0000

Min Threshold(m) 0.0100

Max Threshold(m) 1.0000

Figure A-1: Lichen user interface

1. Load Base Object

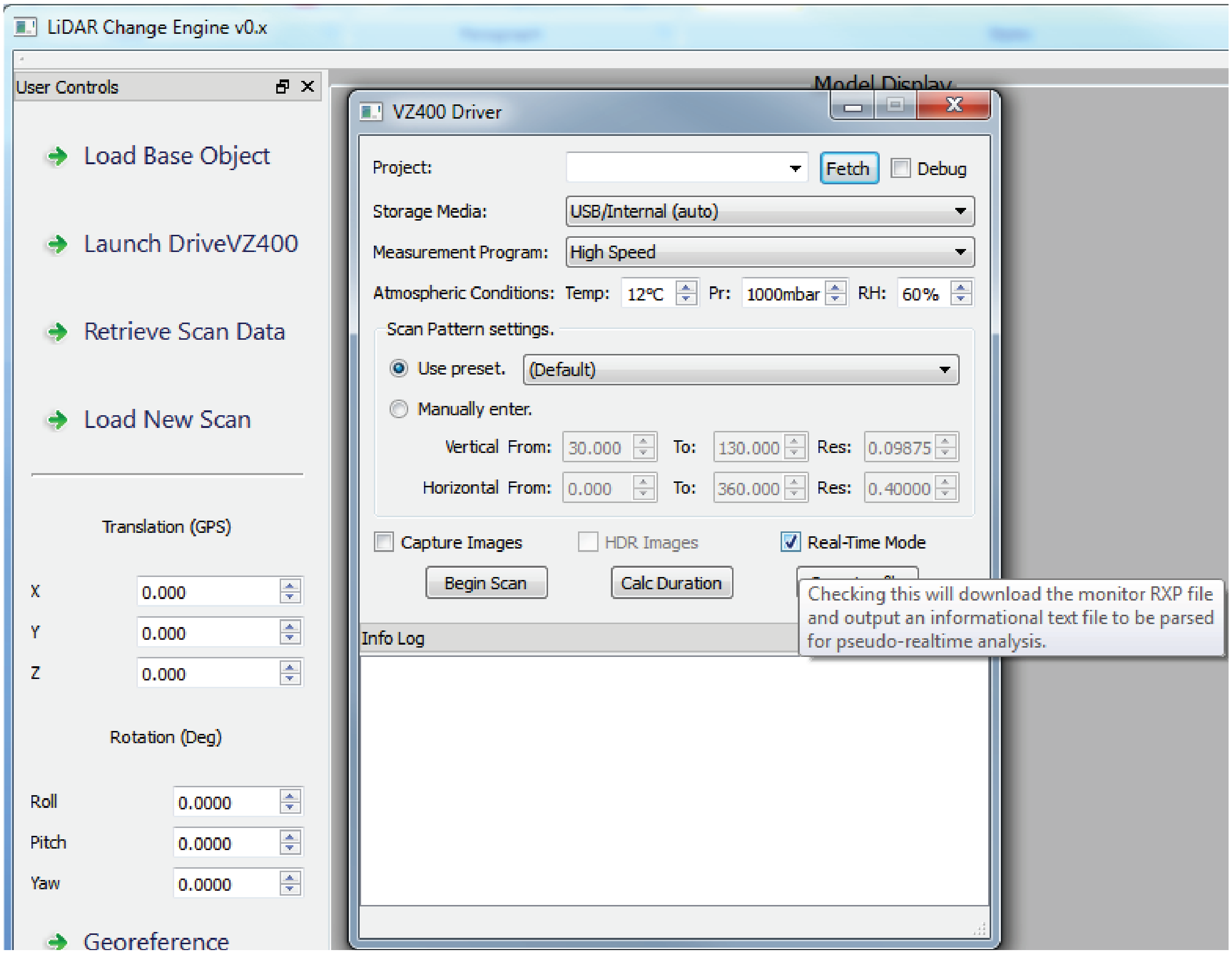
Executing this command will open a file navigation window, allowing the user to navigate to the baseline scan file used for change comparison. The navigation menu only displays binary objects with a file extension of (***.bov**). For ease of navigation, it is recommended that users transfer their baseline scans in the (***.bov**) format into the "Base Models" folder which is located in the *Lichen* program directory. If the baseline scans are not yet in the (***.bov**) format, please see the *BovConverter* program (Appendix C). When a (***.boh**) header file is not present, the program assigns X, Y, Z translation offsets of (0,0,0) meaning that *Lichen* will use the raw point values for the coordinate system. Upon loading the baseline scan, the GUI will display the base model in the display window.

Users of scanners other than a Riegl VZ-400 will skip steps 2 and 3.

2. Launch Scanner Controls

This button is a modular component that is independent of the main *Lichen* program code and is intended to execute a program to collect scan data. Currently, users of a Riegl VZ -400 scanner can push this button to execute the module DriveVZ400, which provides a GUI to operate the scanner. As additional scanner manufacturers release software development kits (SDK), this button will enable the user to launch a GUI to operate the scanner of their choice. The modular nature of this code will facilitate easy addition of different brands of scanners. To work around this issue, scans may be collected using the manufacturer provided software, converted to the (***.bov**) format, and then a user may proceed with the remaining steps to run *Lichen*.

DriveVZ400 operates separate from the *Lichen* program, giving users the capability to switch back and forth between *Lichen* and DriveVZ400 as needed (Figure A-2). The DriveVZ400 application may also be closed and restarted as needed. However, closing *Lichen* will also close DriveVZ400. Please see the DriveVZ400 Manual (Appendix D) for more detailed instructions on how to operate it. When running this program through *Lichen*, it is necessary to ensure that the check box next to "Real-Time Mode" is checked, which will generate a file that *Lichen* will automatically read to attain the Roll and Pitch values, as well as the file location for conversion to binary format.



Translation (GPS)

X 0.000
Y 0.000
Z 0.000

Rotation (Deg)

Roll 0.0000
Pitch 0.0000
Yaw 0.0000

Project:
□ Debug

Storage Media: USB/Internal (auto)

Measurement Program: High Speed

Atmospheric Conditions: Temp: 12°C Pr: 1000mbar RH: 60%

Scan Pattern settings.

● Use preset. (Default)

○ Manually enter.

Vertical From: 30.000 To: 130.000 Res: 0.09875
Horizontal From: 0.000 To: 360.000 Res: 0.40000

□ Capture Images □ HDR Images ☑ Real-Time Mode

Checking this will download the monitor RXP file and output an informational text file to be parsed for pseudo-realtime analysis.

Figure A-2: The DriveVZ400 program integrated into *Lichen*

3. Retrieve Scan Data

This command enables the semi-automated retrieval of data from new scans such as roll and pitch values, as well as launching the standalone *BovConverter* application (Appendix B). Use of this option is dependent on an ASCII text file output by DriveVZ400 operated in "Real-Time Mode." The name of this file is currently "**rtinfo.txt**" and will be found in the same directory as the *Lichen* executable.

Upon clicking the button, a file dialog opens that allows the user to navigate to the folder containing the text file. Once the file is selected, *Lichen* will automatically fetch the required information and import it into the GUI. It will also launch *BovConverter.exe* ready to convert the newly acquired scan. *BovConverter.exe* runs completely independent of *Lichen*, so it will need to be closed manually upon completion.

Users of scanners other than a Riegl VZ400 will skip this step and need to manually input the appropriate roll and pitch values (values of 0.0 would correspond to leveled scan data). You will also need to manually run the *BovConverter* application to convert your new scans. (**Note:** the current version of *BovConverter* converts (***.txt**), (***.obj**)**,** and (***.rxp**) files; it does not currently support other vendor's raw formats. Owners of Leica scanners can convert **(*.ptx)** files to the *bov* format using the *PTXreader* program (Appendix C). This will eventually be expanded to support E57 when available by manufacturers.)

4. Load New Scan

This command is used to open a file dialog and allow the user to designate which scan is to be used to compare to the baseline model for change detection. Once the user has selected the new scan in (***.bov**) format the GUI will display the new scan. Depending on the coordinates of the baseline scan, users may not see both scans until geo-referencing is performed in the next steps.

5. Translation (GPS)

This section of the GUI displays the current X, Y, Z coordinates of the new scan. The default values are set to (0, 0, 0). Users must retrieve the value from their GPS/GNSS device and enter them in the appropriate box. Simply click the mouse over each box or tab to the box and type in the value. There is no need to hit the Enter key.

This step is important in order to geo-reference the baseline and the new scans in a common coordinate system. The position of the new scan is defined by non-truncated GPS coordinates acquired in the field.

Important! Make sure that both sets of scan data are using GPS values from the same coordinate system. For example, if one scan uses UTM North Zone 11 and the second uses the Oregon Coordinate Reference System Salem Zone, the two scans will have vastly different X, Y, Z positions and not line up correctly.

6. Rotation (Deg)

This section of the GUI displays the current values for the roll, pitch, and yaw of the new scan. The default values are set to (0, 0, 0). If the user is operating a Riegl VZ400 laser scanner, the roll and pitch values will already be entered here when the user runs Step 3. For users of other instrumentation, simply mouse-click (or tab) to the appropriate box and enter the desired value. There is no need to hit the Enter key.

The yaw value is the rotation about the Z-axis and is used to orient scans in 3D space. This value is manually entered by the user (e.g. from a digital compass) and will provide the rough alignment of the **New Scan** to the **Base Scan**. The closer the measured value is to the actual value (optimally within a few degrees), the faster *Lichen* will run the automatic alignment algorithm.

7. Geo-reference

This command provides a manual way for the user to update the position of the new scan. It will translate the new scan to the values entered for X, Y, Z, Roll, Pitch, and Yaw. This is automatically performed during Step 9, but this allows the user to make adjustments if necessary. It utilizes the values currently displayed in the GUI.

8. Thresholds

There are two values in this box, labeled **Min Threshold (m)** and **Max Threshold (m)**. These two parameters are used for two different applications and represent a distance in meters. The first application determines the size of the cubes to subdivide each scan into to determine the distance to search for a neighbor when running the "**Auto-Correct Yaw**" command. The second function is to determine the thresholds to use for coloring the magnitude of change detected.

When aligning scans, the larger these thresholds are set to, the faster the two scans will align, but the less accurate the alignment will be. Larger thresholds detect points further out of alignment. However, if too small of a value is used initially, then scans which are not already closely aligned from the user-entered Yaw value will not be aligned properly as the algorithm will only search in the bounds entered.

Useful values are set as the default (Min = 0.01m, Max = 1.0m). For more precise alignments, users can run the defaults or relax the defaults, then run multiple iterations of the alignment algorithm using successively smaller thresholds until the desired level of alignment is achieved.

9. Auto-Correct Yaw

This command executes the *Lichen* program's auto-alignment algorithm. It runs multiple iterations until either no significant change in rotation is determined ($<0.001^{o}$) or it reaches the maximum number of iterations (1,000). It will then geo-reference the new scan to these derived values and update the display GUI to show the new position of the scans.

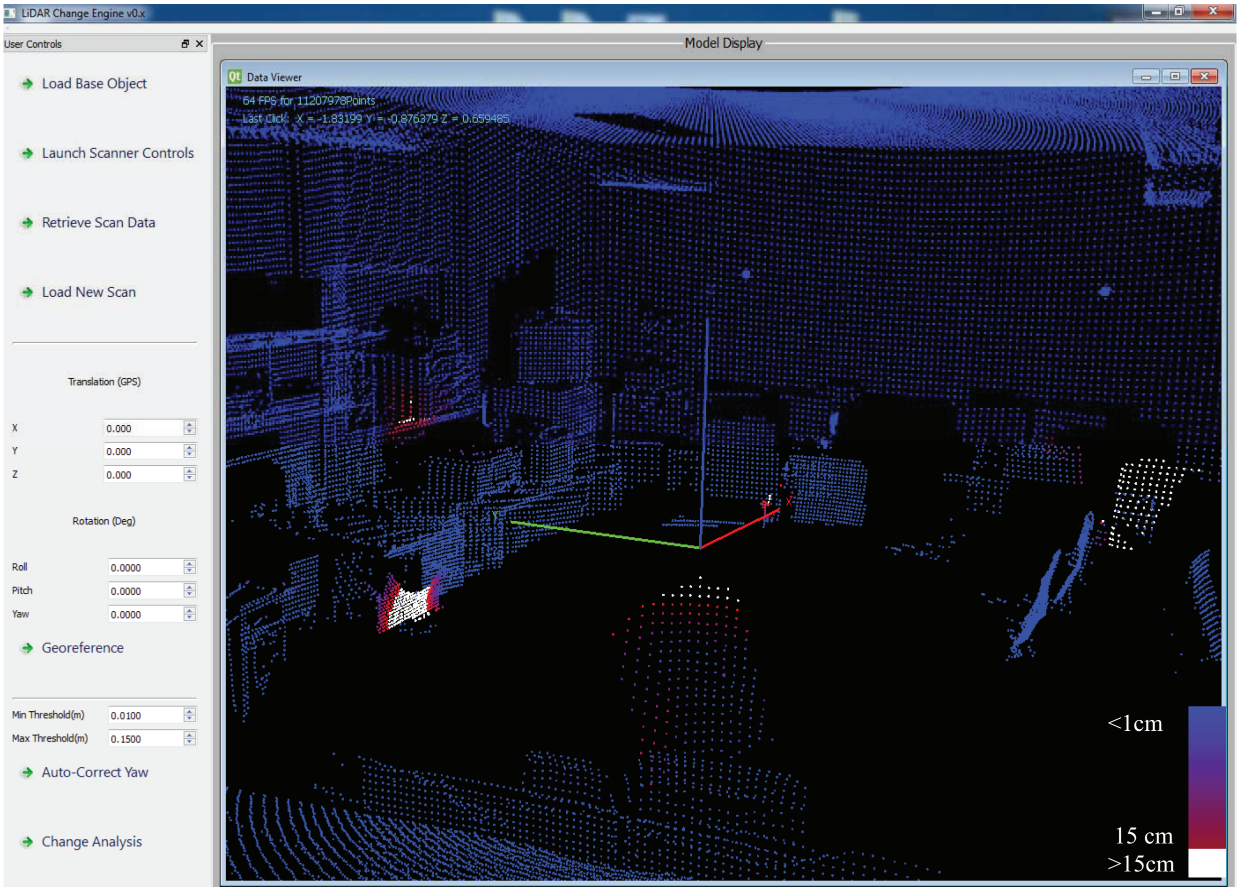
There are a couple of important considerations to bear in mind about this function.

1. The user is aligning one scan position to one section of a model. Scan signal returns farther from the scanner will be more spread out and less dense, creating more wiggle room on the edges of the scan as opposed to having flanking scans that create overlap and bound these edges.

2. The scans are temporally separated. This means that in areas exhibiting large-scale or rapid changes over time, there may be an alignment bias caused by morphological changes in the object face that causes better or worse alignment with the base model. If significant enough change has occurred, it may not be as accurate or even be impossible to achieve a "useful" alignment to the original surface. However, change of this magnitude would show up and indicate a re-scan is needed. In such cases, adjacent scans can be used for the alignment, as is done in baseline processing.

10. Change Analysis

This command tells *Lichen* to perform the change detection analysis on the new scan. It will compare the new scan against the baseline scan and determine what change, if any, has occurred (Figure A-3). It will then hide the baseline model and display a color-coded model in the GUI display window. Currently, grey shading indicates that there are no objects within range to compare against. White indicates areas that are displaying change greater than a threshold value. Red shading shows areas with a measurable change less than the threshold value. The user may navigate this display to visualize the change and determine if a re-scan of the area is warranted.



64 FPS for 11207978Points

Last Click: X = -1.83199 Y = -0.876379 Z = 0.659485

X 0.000

Y 0.000

Z 0.000

Roll 0.0000

Pitch 0.0000

Yaw 0.0000

Min Threshold(m) 0.0100

Max Threshold(m) 0.1500

Figure A-3: Result of change detection analysis performed in the lab.
The white regions are office chairs moved more than 15 centimeters, while the red regions are equipment cases which were opened or rotated.

Final Notes

The display uses a modified version of the GLC_library (version 2.0) provided by Laurent Ribon (http://www.glc-lib.net/). The GUI was developed using Qt version 4.7.

APPENDIX B

BOV CONVERTER V1.0 USER MANUAL

What does it do?

The BovConverter program is a small program used to convert data from (*.**txt**), (***obj**), and (***.rxp**) data into a binary format, resulting in smaller file sizes and improved processing efficiency for *Lichen* analysis. It can be run as an independent application in single use or batch mode, or it may be launched from within the *Lichen* application. This program has been adapted from sample code of the *RivLib* library.

Upon completion, *BovConverter.exe* may generate the following new files:

> **yourfilename.mon.bov** This is an input file for use in the *Lichen* application. This is only output if you use DriveVZ400 with RT checked (real-time mode) from within *Lichen* or as a standalone application.
>
> **yourfilename.bov** This is an input file for use in the *Lichen* application. This file is generated after converting an Object file, ASCII text file, or by converting a full-size raw (***.rxp**) file from a RiScan or RiProject directory. **Warning!** It will take substantially longer both to convert the full-size raw file and to load/run it from within *Lichen*. This is because the raw file may be several hundred megabytes to over a gigabyte in size. Depending on the processing power and hardware resources of the computer running the programs, the computer may take varying amounts of time or be incapable of processing such large data files.
>
> **yourfilename.boh** This is a header file that contains the translation offsets for the processed LiDAR dataset, which is necessary to avoid truncation when converting double precision values (~15 digits) to floating point values (7 digits). *BovConverter* will generate this file automatically when you convert a XYZ ASCII text file or object file. Users of some software generally require these translations (e.g. POP matrix in RiscanPro) to reduce a set of coordinates from the global level to the project level in order to prevent truncation of values. The offsets are subtracted from every point in the dataset. These offsets are applied if the X, Y, or Z values are greater than 1,000. Below is an example of a typical conversion.

Table B-1: Typical Conversion Example

	Global Value (GPS)	Project Value	Translation Offsets
X	456,973.354	973.354	456,000
Y	543,563.972	563.972	543,000
Z	76.234	76.234	0

What is required to use it?

There are three possible file formats that can be used as input data for BOV Converter processing.

1. Standard Object file format (***.obj**) usually resulting from triangulation of point cloud datasets. Most often these will be the baseline scans that will be used for change comparison and will consist of an Object file such as **yourfilename.obj**.

2. ASCII text files **(*.txt**) which consist of X,Y,Z coordinates. Headers are designated by the '#' or 'X' characters. Commas, tabs, or spaces can be used as delimiters. Values after the X, Y, Z coordinates on the line are ignored.

3. Proprietary V-line series file format (***.rxp** for scan) or (***.mon.rxp** for monitor file). These are the files output by the Riegl VZ400 scanner. These can be retrieved from the RiProject or RiSCAN folder where the project was created on your computer, the scanner's internal hard drive, or a usb flash drive. For example:
C:\Riegl Scans\yourprojectname.RiSCAN\SCANS\ScanPos001\SINGLESCANS\yourfilename.rxp

Alternatively, using the program DriveVZ400 will place a smaller monitor file in the same directory as the DriveVZ400 executable. The monitor file is a subset of the full scan file data, which is much smaller, allowing for quicker browsing and usage. This file will look like **yourfilename.mon.rxp**

How do I use it?

There are three ways to utilize the BOV Converter program. The first method is manual operation for single or multiple input files, and will typically be used to convert the baseline scan(s) that you want to use for the change comparison. The second method is semi-automated and will be launched from within the *Lichen* user interface. This method is intended to simplify the workflow when the user is in the field by removing the need to open and navigate multiple programs and file directories.

1. Manual Scan Conversion

Open the directory in which you have placed the BOV Converter program (*BovConverter.exe*). The default location for this is a folder within the *Lichen* program directory such as *C:\ProgramFiles (x86)\Lichen\bov*.

Use the mouse to drag and drop the files that you wish to convert directly onto the executable named *BovConverter.exe*. This will bring up a console window that will ask you if you wish to filter the scans. When converting multiple scans simultaneously, they will be processed in reverse order (e.g. the last scan selected will be processed first). Additionally, any filtering options selected will be applied to each scan run in batch mode, so processing baseline scans in the same batch as new scans is generally not recommended.

Enter "**0**" to **skip** filtering and then press the Enter key. Filtering of baseline scans is generally done as part of LiDAR post-processing. In general, you will not want to filter your base scan.

Enter "**1**" to filter the resulting output file. In general, you will want to filter the new scans that you will be comparing to the baseline scan. This is because in most circumstances, the new scan will collect data far beyond the scope of your comparison, either in terms of distance or directionality. Entering "**1**" will apply a 360 degree double circular (e.g. range) filter on the dataset based upon two parameters the user will be asked for, namely:

a. Select the **minimum** distance from the scanner in **meters**. This will discard all points from the scanners position to the specified distance. Selecting a value of 0 will result in only filtering data beyond the **maximum** distance entered. Typical values will be between 0 and 50.

b. Select the **maximum** distance from the scanner in **meters**. This will discard all points farther away from the scanner than the **maximum** distance selected. This method is a good way to eliminate data beyond the scope of the project or to limit it based upon scanner's accuracy specifications for distances. Typical values for topographic scanners will be 100 to 150. For closer projects, shorter distances will be appropriate.

2. *Lichen*-launched Scan Conversion

In order to use this method, you must have launched the DriveVZ400 program from within *Lichen*, and you must have performed a scan with the "Real-Time Mode" check box checked. This is because in "Real-Time Mode", DriveVZ400 generates a file containing the location of the scan performed.

From within the *Lichen* program, select the "**Retrieve Scan Data**" button. This will automatically launch *BovConverter.exe* and tell it the appropriate scan to convert without the need to find the scan file itself or drag and drop it.

Select your filtering preferences (described previously in Manual Scan Conversion) and proceed. Once the conversion is completed you may close the BovConverter console window and load the newly converted scan for comparison.

Advanced details: What does the bov data format look like?

The bov format is simply 3 floating point values (x,y,z coordinates) for each point. It can be read in using C++ using the following data structure and function:

```
typedefstructfloatXYZpoint
{
        floatx,y,z;
} fXYZ;

fXYZ* readbin(const char* fname_in, unsigned long intnumpts)
{
        int ret;
        numpts=0;//reset point count to zero
        longlsize;
        fXYZ* mypt = NULL; //set-up pointer to data


        FILE* fin = fopen(fname_in, "rb");
        if (fin == NULL)
        {
                //could not open file
                fprintf(stderr,"Could not open input file %s.  Skipping to next!
                        \n",fname_in);
                fprintf(stderr,"\n");
        }
        else
        {
                //get file size and number of points
                intmyptsz = sizeof(fXYZ);
                fseek(fin,0,SEEK_END);
                lsize = ftell(fin);
                rewind(fin);
                numpts = lsize/myptsz;

                printf("file size: %i\n",lsize);
                printf("numpts: %i\n",numpts);

                //allocate memory
                mypt = newfXYZ[numpts];

                //read in data
                fread(mypt,myptsz,numpts,fin);
        }

        fprintf(stderr,"%i points successfully loaded! \n",numpts);

        returnmypt;
}
```

Note that the header file (*boh*) contains three double values, representing offsets to avoid coordinate truncation. The program reading the file will need to keep track of these offsets and apply them to other datasets for appropriate comparison.

```
typedefstructdoubleXYZpoint
{
    doublex,y,z;
} dXYZ;

dXYZ* readbinheader(constchar* fname_in)
{
    longlsize;
    dXYZ* myoffsets = NULL; //set up pointer

    FILE* fin = fopen(fname_in, "rb");

    if (fin == NULL)
    {
        //cannot open file
        fprintf(stderr,"Could not open input file %s.  Skipping to next!
            \n",fname_in);
        fprintf(stderr,"\n");
    }
    else
    {
        //Determine number of points
        intmyptsz = sizeof(dXYZ);
        fseek(fin,0,SEEK_END);
        lsize = ftell(fin);
        rewind(fin);
        numpts = lsize/myptsz;  //should only be 1!

        printf("file size: %i\n",lsize);
        printf("numpts: %i\n",numpts);
        myoffsets = newdXYZ[numpts];

        //read in offsets
        fread(myoffsets,myptsz,numpts,fin);
    }

    fprintf(stderr,"%i points successfully loaded! \n",numpts);

    returnmyoffsets;
}
```

APPENDIX C

PTX READER V1.0 USER MANUAL

Programmed by Michael J. Olsen (Oregon State University), Kevin Ponto (University of California, San Diego), and Jason Kimball (University of California, San Diego) under the supervision of Falko Kuester (University of California, San Diego).

This user manual assists owners of Leica scanners to convert their *ptx* data to the *bov* format for use in Lichen.

What does it do?

The *PTXreader* program serves two primary purposes:

1. Extract data from scans in a Leica PTX format, including:

    a. XYZ coordinates with Intensity (I) and RGB color values as a text file.

    b. XYZ coordinates into a bov file format (see Appendix C).

    c. A png image file of the 3D scan projected to a 2D grid (spherical) with RGB photographic color (Figure C-1).

    d. A pgm image file of the 3D scan projected to a 2D grid (spherical) with grayscale colors to represent the range of the points from the scanner (Figure C-2).

    e. A pgm image file of the 3D scan projected to a 2D grid (spherical) with grayscale colors to represent the intensity of the points from the scanner (Figure C-3).

2. Filtering of scan data based on edited 2D images. The user can then edit the images by erasing unwanted sections of the scan in traditional 2D photographic editing software. The user can then run PTXreader to apply those edits to the point cloud. Many tools are available in 2D editing software that can enable efficient processing workflows, particularly for large datasets, which require significant computational resources to work with in 3D.

How do I run this program?

For using the data with *Lichen*, the user does not need the full capabilities of *PTXreader*. The user runs the software by:

1. Dragging and dropping the desired ptx file into the ptxreader.exe.

2. Selecting mode “e” for extraction.

3. Typing “b” for “.bov” file output.

The program then writes the “.bov” and “.boh” files for the scans.

For more information on additional features of *PTXreader* and how it can be used, see:

Olsen, M.J., Ponto, K., Kimball, J., Kuester, F., and Seracini, M. (2010).*2D open-source editing techniques for 3D laser scans*, Proceedings of Computer Applications and Quantitative Methods in Archaeology - CAA'2010. Javier Melero, Pedro Cano, & Jorge Revelles, p. 47-50.

Additional data output examples:

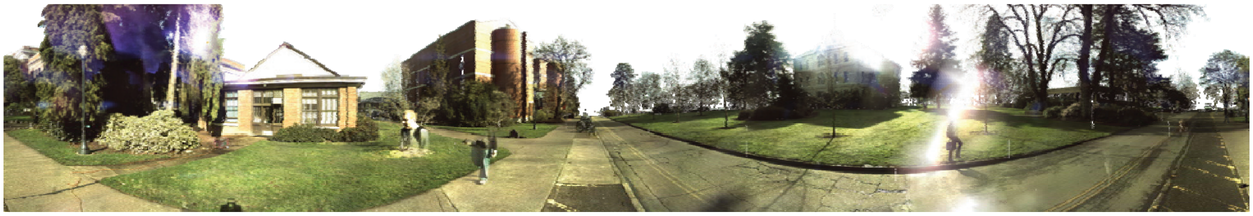


Figure C-1: Photograph output

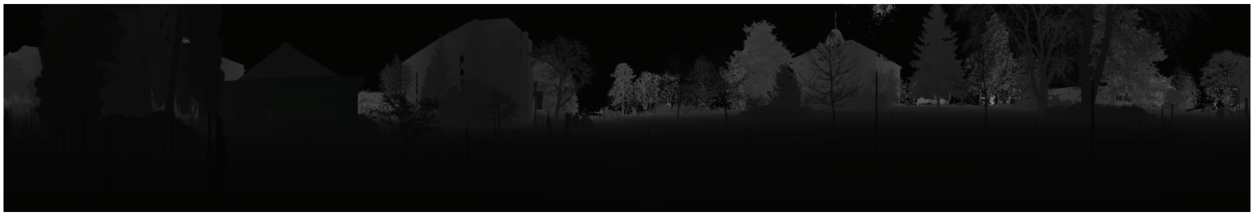


Figure C-2: Range output

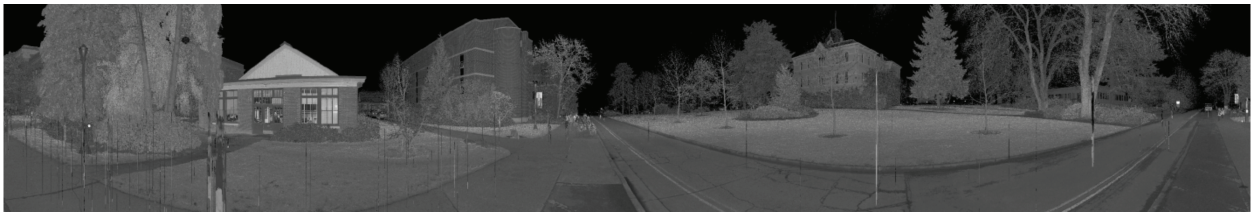


Figure C-3: Intensity output

What is the PTX format?

The PTX format is an ASCII format created by Leica Geosystems for scan data export. An important part of this file format is that it preserves the spherical grid structure of the point cloud.

The file format consists of ASCII text, as follows:

Number of Columns
Number of Rows
Rotation Matrix line 1 (3 values)
Rotation Matrix line 2 (3 values)
Rotation Matrix line 3 (3 values)
Translation vector (3 values)
Combined Transformation Matrix line 1 (4 values)
Combined Transformation Matrix line 2 (4 values)

```
Combined Transformation Matrix line 3 (4 values)
Combined Transformation Matrix line 4 (4 values)
X Y Z I R G B
X Y Z I R G B
….
….
X Y Z I R G B
```

Where there are Number of Columns * Number of Rows lines of X Y Z I R G B. RGB values are scaled as integer values from 0 to 255 and Intensity values (*I* are scaled as floating point values from 0 to 1). A scan grid cell with no data is designated by:

0 0 0 0.5 0 0 0 rather than X Y Z I R G B values.

Note that there can also be multiple scan grids within a single file.

APPENDIX D

DRIVEVZ400 V1.0 USER MANUAL

5/19/2011

Programmed by Evon Silvia, Graduate Research Assistant, School of Civil and Construction Engineering, evon.silvia@gmail.com under the supervision of Michael J. Olsen (Oregon State University).

Introduction

The DriveVZ400 tool is a convenient, lightweight scan collection interface that can run by itself or within other programs (Figure D-1). It cannot process data—it can only record. It can be used on any windows-based device with networking capabilities and all of the information needed to record a scan is conveniently displayed on one screen. It also supports automatic or command-line operation.

Several other viable options exist to operate the RIEGL VZ-400 scanner.

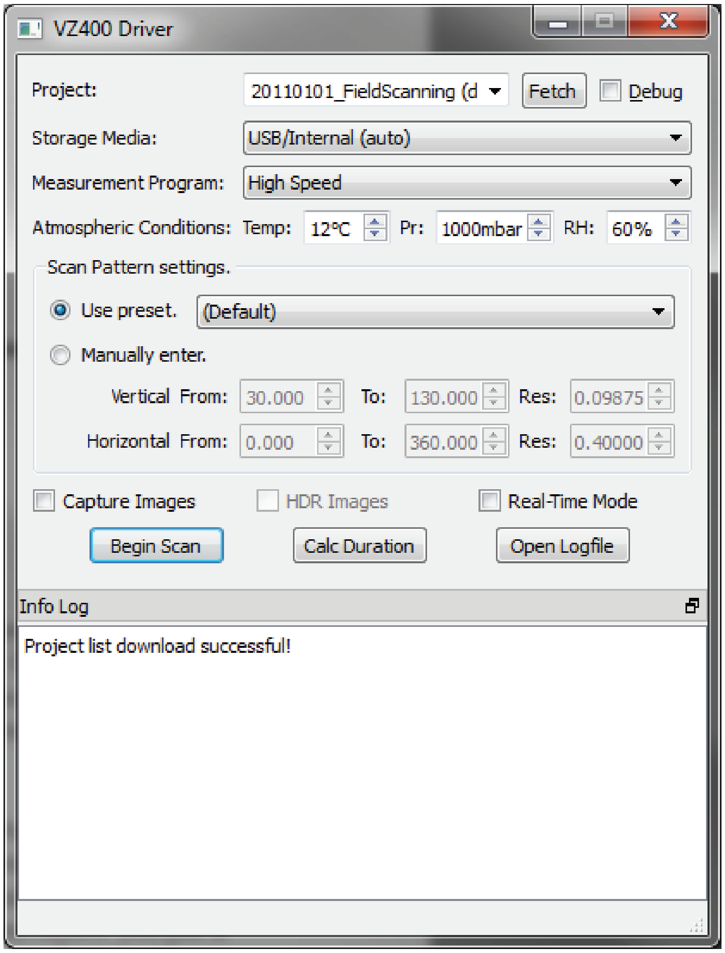
On-board interface. This is the most straightforward manner, but the menu structure can be counter-intuitive and inconvenient. Simple operations like entering a project name require pressing buttons repeatedly. Every time you press a button the scanner's setup has been disturbed a small amount, which can be a problem in high-precision work.

RiSCAN PRO. RiSCAN Pro (available through Riegl) can be installed on a computer to run the scanner either wirelessly or through an Ethernet connection. This program can open, display, edit, and process the raw data produced by the scanner. It can also conveniently record new scans based on the previous scans.

Telnet. The scanner can be communicated with using telnet, a legacy computer terminal communication protocol. Through telnet the scanner is sent text-based commands. These raw commands are interpreted by the scanner. A list of these commands and their use is in the scanner's manual. RiSCAN PRO and DriveVZ400 both use these commands in a manner invisible to the user.

Connecting to the scanner

Open the executable either by double-clicking it or running it from the command-line—see the next section for instructions using the script codes. The interface should appear.



VZ400 Driver

Project: 20110101_FieldScanning (d

□ Debug

Storage Media: USB/Internal (auto)

Measurement Program: High Speed

Atmospheric Conditions: Temp: 12°C Pr: 1000mbar RH: 60%

Scan Pattern settings.

● Use preset. (Default)

○ Manually enter.

Vertical From: 30.000 To: 130.000 Res: 0.09875

Horizontal From: 0.000 To: 360.000 Res: 0.40000

□ Capture Images □ HDR Images □ Real-Time Mode

Info Log

Project list download successful!

Figure D-1: The DriveVZ400 main screen.

Connecting to the scanner is fairly straightforward and can be done in one of two ways:

- Wireless. Once it has powered up the scanner becomes a WiFi hotspot, establishing wireless network connections just like other WiFi wireless access points or routers. The network name is the same as its serial number (S#######) and uses WEP encoding. Check with the lab manager to get the network password if you don't have it. This connection can be a little slow when downloading raw data files.

- Ethernet (wired). The VZ400 has two Ethernet ports, one on the base requiring a special adapter (100MBit) and one on the scanner head itself (1GBit). Most modern computers can switch cable type automatically, but to be safe you should use a cross-over Ethernet cable.

You can browse files on the scanner using Windows Explorer using either the serial number or the IP Address 10.0.0.1 (wireless) or 192.168.0.234 (wired) as the network address. Note that DriveVZ400 defaults to using the scanner's serial number. If it cannot connect and you know that you have connected, try opening the IP address in Windows Explorer first. Sometimes the serial number does not register right away. If your scanner has a different serial number, you must edit the code.

For more detail on connecting to the scanner, check the scanner owner's manual.

Using the program

You only need to be connected to the scanner to use [Fetch] and [Begin Scan]. Otherwise you do not have to maintain a connection to the scanner while the program is running. Note that the conversation with the scanner is stored in "C:\Temp\logfile.txt" for debugging purposes and can be opened after scan completion with [Open Logfile]. At any time you can pop-out the Info Log by double-clicking its title bar. Return it by double-clicking again.

The interface is organized in a logical flow: project setup, site conditions, scan settings, and options. Only the scan settings need change between scans for most sites.

Project setup

Project name

All projects are stored on the scanner or a USB drive plugged into it. Type in a new project name in the entry box, or download a list of existing projects from the scanner and select one. Some notes:

- All projects have a ".riproject" extension. The program will not recognize projects without that extension.
- Existing projects will have a tag appended to their name indicating where the project was detected. These tags are not part of the project name.
- "(internal)" indicates that the project was found on the scanner's internal drive.
- "(usb)" indicates that the project was found on the USB drive.
- The Debug option is meant for offline troubleshooting purposes and should not be used in normal operation.

Storage Media

Select where you want the project and scan data to be stored. "USB/Internal" is recommended for nearly all projects—it attempts to save on the USB but will save on the internal drive if no USB drive is detected.

Measurement Program

"High Speed" works for most applications. Using "Long Range" increases scan time by approximately 2.5 times, but can increase measurement range.

Atmospheric conditions

If you brought a thermometer and barometer with you, enter in the atmospheric conditions and the scanner will automatically correct measured values. Temperature is in degrees Celsius, barometric pressure is in millibars, and relative humidity is in %. Relative humidity has a minimal effect (±0.25ppm if you default to 50%), so do not worry too much about it if you are unable to measure that.

Scan settings

This is the meat of the program. All angles are measured in degrees and define the scan window to be recorded by the scanner. Vertical angles are measured from the vertical axis going down and horizontal angles are measured counter-clockwise from the scanner's power port, up to 720°. See the next section for details on increment precision. Usually you need to do no finer than thousandths of a degree. While running a scan, verify that the parameters received by the scanner are the same as the ones entered.

Common presets are provided based on scan time—add more by editing the VZConfig.cpp code file. The "(Default)" scan does a scan at the coarsest possible resolution. 2min and 5min are good for medium- and high- density 360° scans.

To get an estimate of how long a scan with the given parameters should take, click [Calc Duration]. Note that actual scan time varies from this value.

Scan options

Image Capture

The scanner has a calibrated camera mounted on top of it for collecting images aligned with the scan just taken. Enable this option to have the camera collect a full set of images. The number of images taken depends on the CAM_OVERLAP property, which for now must be set using Telnet or RiSCAN PRO.

There is an option to capture image sets for High Dynamic Range (HDR) image processing. This feature is still under development. Use at your own risk!

Real-Time Mode

This feature will enable additional features that are needed for real-time analysis. Specifically, the scanner will record highly-accurate inclination values prior to scanning, a low-resolution "monitor file" of the scan will be downloaded to the same directory as the executable, and a text file "vz_info.txt" will be output in the same directory as the executable (by default) that can be parsed by a real-time analysis program (see below).

Go!

Click [Begin Scan] to send all of the settings to the scanner. Verify that the settings were sent correctly by monitoring the text output in the Info log.

Once the scanner has begun scanning, you are free to change the settings in preparation for the next scan. Each new scan in a project opens a new scan position. The scan can be aborted at any time by clicking [Abort] next to the progress bar.

Scan retrieval

The scanner produces several files with each scan, all of them being time-stamped with the date and time (in GMT) of the scan as its name. The files are stored in a *.riproject folder with the project's name in front, and in the corresponding ScanPos### folder for the current scan position. Note that the RIPROJECT folder will be either stored on the scanner's internal hard drive or on the USB drive plugged into it, depending on your settings.

These are the files usually stored in the ScanPos### folder upon scan completion:

- YYMMDD_HHMMSS.rxp – the binary raw scan data itself, with housekeeping and monitor data included.
- YYMMDD_HHMMSS.mon.rxp – a small sample of the raw scan data. Useful for real-time change analysis and verifying that the correct area was scanned.
- YYMMDD_HHMMSS.png – a small image file that displays the scan in a polar grid. Useful for easy visual inspection of the scan without having to load the raw data.
- YYMMDD_HHMMSS.pat – an ASCII file with the scan parameters.
- YYMMDD_HHMMSS.prv – a "preview file" for use by RiSCAN.
- YYMMDD_HHMMSS_Image###.jpg – scan photos taken along the horizontal range of the scan. Some parameters can be read by an ASCII text reader.

Command-line operation

The DriveVZ400 program can be run from the command line for automated operation, with or without user input. This is accomplished by a series of arguments separated by spaces following the executable in a form similar to the following:

```
C:\programs\DriveVZ400.exe m=1 n=20110222_test a=17,1080,41 p=42.134,43.570,.02,321.897,456.123,.02 i=0 r=1
```

The arguments above follow a specific format. It is important to note that there must be no spaces within a particular argument for the program to recognize it. The actual order of the tags does not matter, with the exception that the run mode (m,M) must always come first. Here is a rundown of each tag:

- m,M: run mode. This tag must come first! Options:
    - 0—Simply launch the program and ignore all other arguments.
    - 1—Load the arguments into the program and wait for the user.
    - 2 (still to be implemented)—Load the arguments and start the scanner automatically.
- n,N: project name. This is the project name where all of the raw scan data will be stored, usually on a USB drive plugged into the scanner. Remember: there can be NO SPACES, even in folder or project names.
- a,A: atmospheric conditions. This is the air temperature (°C), atmospheric pressure (millibars), and relative humidity (%) of the current environment, separated by commas.
    - Temp range: -50 to 100°C, 1° precision (12° default)
    - Pres range: 600 to 1200mbar, 1mbar precision (1000mbar default)
    - RH range: 0 to 100%, 1% precision (60% default)
- p,P: scan preset pattern. These are preset patterns with all of the parameters set up for you. You'll probably default to the 2min for now. This should be somewhat case-insensitive. Options:
    - "2min"—360° overview at 0.05° increments
    - "5min"—360° overview at 0.03° increments
    - "default"—extremely fast 360° overview... good for testing code
- p,P: (alternate) scan pattern. This is the scan window parameters that set up the actual scan range, separated by commas: vstart,vstop,vincr,hstart,hstop,hincr.
    - vstart: vertical start position. 30 to 130°, 0.001° precision
    - vstop: vertical stop position. same.
    - vincr: vertical angle increment. 0.001 to 0.098750°, 0.000001° precision theoretically (0.0001° practical)
    - hstart: horizontal start position. 0 to 360°, 0.001° precision
    - hstop: horizontal stop position. 0 to 720°, 0.001° precision
    - hincr: horizontal increment. 0.001 to 0.499°, same precision as vertical.

- i,I: images requested. true/false toggle of whether or not images should be recorded. Options:
    - 0—do not record images after the scan
    - 1—if possible, record images after the scan
    - 2 (under development)—do HDR photography
- r,R: operate in real-time mode. For *Lichen*, you will always have this marked 1. Options:
    - 0—operate normally, recording data as usual.
    - 1—record roll/pitch values prior to the scan, download the monitor file, and write an output file to the executable's directory.

Real-time mode

If the user elects to run in real-time mode, an ASCII file named “vz_info.txt” will be created in the same directory as the executable. This file can be parsed by another program upon scan completion to retrieve information needed for real-time analysis. The tags and information in the following line will always remain the same. There will always be a blank line after each tag set.

- #VZ_PROJ: the following line will be a string with the project’s name on the scanner’s storage device.
- #VZ_PATH: the following line will be a string with the path to the downloaded monitor RXP file.
- #VZ_INCL: the following two lines will be floats with the roll and pitch values for the current instrument setup.

Calling another program using Qt

There are a couple ways to run one program from within another within Qt. QDesktopServices::openURL() is a very useful function that opens standard files like *.txt or *.doc files using the default program registered with the operating system, but it doesn’t really work for the DriveVZ400 program since there’s no file type associated with it. This section assumes you are familiar with Qt. If you are not, please refer to http://qt.nokia.com/products/for more information.

QProcess is a class in Qt intended for running programs in another thread. Here is some sample code that calls the DriveVZ400 program using the arguments described above:

```
QProcess* vz = new QProcess(this);
QStringListargs;
args      << "m=1"
<< "n=20110222_test23"
<< "a=19,1043,70"
<< "p=35,45,.05,155,165,.1"
<< "i=1"
<< "r=1";
vz->start("C:/programs/DriveVZ400.exe", args);
if(vz->waitForStarted(-1))
          this->userUpdate("Program started!");
qApp->processEvents();  //execute any events collected in the mainEvent queue (such as displaying the line I just
output) before executing the child program
if(vz->waitForFinished(-1))
{
          this->userUpdate("Program finished!");
          this->userUpdate(QString("%1").arg(vz->exitCode()));
}
deletevz;
```

`QProcess::start()` runs the child program in a separate thread, so the host/parent program will still be running at the same time under another process. The `QProcess::waitForStarted(-1)` and `QProcess::waitForFinished(-1)` functions lock the parent program until an exit signal is received from the child program. These functions are not required, but they can be helpful if your parent program cannot continue without the scan information.