STATE OF CALIFORNIA • DEPARTMENT OF TRANSPORTATION
TECHNICAL REPORT DOCUMENTATION PAGE
TR0003 (REV 10/98)

1. REPORT NUMBER	2. GOVERNMENT ASSOCIATION NUMBER	3. RECIPIENT'S CATALOG NUMBER
CA17-2996		

4. TITLE AND SUBTITLE	5. REPORT DATE
Research and Support for MTLS Data Management and Visualization	September 30, 2016
	6. PERFORMING ORGANIZATION CODE
	AHMCT Research Center, UC Davis

7. AUTHOR	8. PERFORMING ORGANIZATION REPORT NO.
Travis Swanston, Kin Yen, Stephen Donecker, Bahram Ravani, and Ty Lasky	UCD-ARR-16-09-30-01

9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. WORK UNIT NUMBER
AHMCT Research Center UCD Dept. of Mechanical & Aerospace Engineering Davis, California 95616-5294	
	11. CONTRACT OR GRANT NUMBER
	IA65A0560, Task 2996

12. SPONSORING AGENCY AND ADDRESS	13. TYPE OF REPORT AND PERIOD COVERED
California Department of Transportation P.O. Box 942873, MS #83 Sacramento, CA 94273-0001	Final Report October 2015 – September 2016
	14. SPONSORING AGENCY CODE
	Caltrans

15. SUPPLEMENTARY NOTES

16. ABSTRACT

This report documents the research project "Research and Support for MTLS Data Management and Visualization." The primary goal of the project was to support Caltrans District 4 in their upgrade and enhancement efforts for Mobile Terrestrial Laser Scanning (MTLS) data management. The research included investigating data storage, management, and organization. It also included developing software tools for automated data cataloging to a geospatial-enabled web-based portal to view MTLS data availability. This research expanded District 4's leading-edge MTLS capabilities.

17. KEY WORDS	18. DISTRIBUTION STATEMENT
Mobile Terrestrial Laser Scanning (MTLS), Mobile LiDAR, Light Detection and Ranging (LiDAR), Mobile mapping, Data management	No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.

19. SECURITY CLASSIFICATION (of this report)	20. NUMBER OF PAGES	21. COST OF REPORT CHARGED
Unclassified	65	

DISCLAIMER

Advanced Highway Maintenance and Construction Technology Research Center

Department of Mechanical and Aerospace Engineering
University of California at Davis

Research and Support for MTLS Data Management and Visualization

Travis Swanston, Kin Yen, Stephen Donecker, Bahram Ravani &
Ty A. Lasky: Principal Investigator

California Department of Transportation

Division of Research, Innovation and System Information

ACKNOWLEDGMENTS

ABSTRACT

This report documents the research project "Research and Support for MTLS Data Management and Visualization." The primary goal of the project was to support Caltrans District 4 in their upgrade and enhancement efforts for Mobile Terrestrial Laser Scanning (MTLS) data management. The research included investigating data storage, management, and organization. It also included developing software tools for automated data cataloging to a geospatial-enabled web-based portal to view MTLS data availability. This research expanded District 4's leading-edge MTLS capabilities.

EXECUTIVE SUMMARY

This research study focused on support for Caltrans District 4 (D4) in their upgrade and enhancement efforts for Mobile Terrestrial Laser Scanning (MTLS) data management. The research included investigating data storage, management, and organization. It also included developing software tools for automated data cataloging to a geospatial-enabled web-based portal to view MTLS data availability.

Caltrans D4 has been acquiring MTLS data for several years both through consultant services and through the Caltrans MTLS system. D4 has reached a point where the storage, management, and organization of this data have become both critical and challenging. Through this research, the Advanced Highway Maintenance and Construction Technology (AHMCT) Research Center has investigated D4's current data management situation and needs, developed targeted recommendations for D4's immediate use, and developed automated data cataloging tools to assist D4's MTLS data management.

Research Objectives and Methodology

There were two main objectives of this research study: first, to assess and understand Caltrans D4's data management situation and needs, and second, to develop software tools for automated data cataloging of MTLS data availability for a web-based GIS portal.

AHMCT researchers addressed the first objective by meeting with key D4 personnel, including representatives from Surveys and from Information Technology (IT). We held several meetings, which were also attended by the Division of Research, Innovation and System Information (DRISI), either in Oakland at the D4 district office or by teleconference and webinar. Through these discussions and the inspection of their file systems, we developed a clear understanding of the data size, the directory structure for each project, the contents of each project, and the physical configuration of the drives being used to store the data. Several problematic issues were identified through this process, and recommended actions were provided to D4 at the earliest possible time. AHMCT also provided assistance in consolidating and securing essential MTLS data that was spread across multiple external hard drives. The detailed findings and recommendations of this portion of the research are provided in this report.

The second objective was met by working with D4 to define their needs for an automated data-cataloging tool. The tool, named LidarCrawl, is written in the Java programming language, and works with the PostgreSQL database with the PostGIS extension. This tool catalogs MTLS data for a web-based portal to investigate MTLS data availability. LidarCrawl greatly enhances D4's ability to explore and manage its MTLS data. Software modules can be added to utilize the existing GIS IT infrastructures of other Caltrans districts or DRISI if the PostGIS database is not being used for their GIS database.

Results and Recommendations

This research resulted in a set of recommendations and executed actions to improve MTLS data management in Caltrans D4. As part of this effort, D4's remaining MTLS data that was stored on external drives was located and copied into the NetApp system, significantly reducing

risks to this vital asset, including corruption or complete loss of data gathered with costly and labor-intensive efforts.

The research also provided a software tool to automatically catalog existing MTLS data. This tool also provides a web-based viewing mode that allows a general Caltrans user to check the availability of MTLS data at a given geographic location. This tool greatly enhances the ability of Caltrans D4 to properly manage and discover their MTLS data.

The following recommendations are made based upon the results of this research study:

- Leverage the recommendations and practices developed for D4 by seeing how they apply in other districts working with MTLS and adjusting them as needed.

- Work with all Caltrans districts and Headquarters (HQ) to develop an MTLS data management plan and policies, including those for data retention, data storage, MTLS data distribution and sharing, and data security for sensitive data such as major bridge structures. Also, develop policies for the use of internet cloud-based storage and backup and recommendations for the standardization of project survey notes and reporting formats.

- Allow Caltrans to use commercial cloud-based backup services or pursue an exception to allow these services. The cost differential between commercial services and the state's system makes it unreasonable to use the latter for cloud-based backup of LiDAR data. Should Caltrans not be allowed to use commercial cloud-based services, an entirely different approach without cloud backup should be pursued. Doing so, however, would cause Caltrans to lose the known benefits of a cloud-based approach.

- Consider implementing the LidarCrawler tool in other districts to help them manage their MTLS data.

- Consider the possibility of expanding the LidarCrawler tool to help Caltrans HQ and districts manage other geospatial data.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

Acronym	Definition
AD	Active Directory
AHMCT	Advanced Highway Maintenance and Construction Technology Research Center
AHN2	Actueel Hoogtebestand Nederland
ALA	Alameda county
AVI	Audio Video Interleave
AWS	Amazon Web Services
CA	California
CalCloud	California Cloud Services
Caltrans	California Department of Transportation
CC	Contra Costa county
CIFS	Common Internet File System
CORS	Continuously Operating Reference Station
CPU	Central Processing Unit
CRREL	Cold Regions Research & Engineering Laboratory
CSV	Comma Separated Values
D4	District 4
DB	Database
DDR	Double Data Rate
DEM	Digital Elevation Model
DMS	Data Management Systems
DOT	Department of Transportation
DRISI	Caltrans Division of Research, Innovation and System Information
ECC	Error-Correcting Code
ESRI	Environmental Systems Research Institute
FME	Feature Manipulation Engine
GB	Gigabyte
GDAL/OGR	Geospatial Data Abstraction Library
GIS	Geographic Information System
GML	Geography Markup Language
GNSS	Global Navigation Satellite System
GRiD	Geospatial Repository and Data
GUI	Graphical User Interface
HP	Hewlett Packard
HQ	Headquarters
IMU	Inertial Measurement Unit
iSCSI	Internet Small Computer Systems Interface
IT	Information Technology
JAR	Java Archive
JPEG	Joint Photographic Experts Group
JSON	JavaScript Object Notation
KML	Keyhole Markup Language
LAS	Log ASCII Standard
LiDAR	Light Detection And Ranging
LRS	Linear Referencing System
MRN	Marin county
MS	Microsoft
MTLS	Mobile Terrestrial Laser Scanning
NAP	Napa county
NFS	Network File System

Acronym	Definition
NGA	National Geospatial-Intelligence Agency
NSF	National Science Foundation
OHS	Overhead Signs
PB	Petabyte
PDAL	Point Data Abstraction Library
PM	Postmile
POSPac	Position and Orientation System Postprocessing Package
QA/QC	Quality Assurance/Quality Control
RAID	Redundant Array of Inexpensive Disks
RAM	Random Access Memory
RINEX	Receiver Independent Exchange Format
ROW	Right of Way
RPM	Revolutions Per Minute
SATA	Serial AT Attachment
SCL	Santa Clara county
SF	San Francisco county
SFOBB	San Francisco / Oakland Bay Bridge
SHP or shp	Shape
SM	San Mateo county
SOL	Solano county
SON	Sonoma county
SQL	Structured Query Language
SSD	Solid State Drive
TB	Terabyte
UDOT	Utah Department of Transportation
UTC	Coordinated Universal Time
WFS	Web Feature Service
WKT	Well-Known Text

CHAPTER 1:
INTRODUCTION

Background

Mobile Terrestrial Laser Scanning (MTLS) is a technology that combines the use of a laser scanner(s), the Global Navigation Satellite System (GNSS), and an Inertial Measurement Unit (IMU) on a mobile platform to produce accurate and precise geospatial data [4]. The data is initially adjusted by post-processed kinematic GNSS procedures from separate GNSS base stations placed throughout the project area. Next, the GNSS solution is combined with the IMU information to produce geospatial data in the form of a point cloud. This point cloud is then adjusted by a local transformation to well-defined points throughout the project area to produce the final geospatial values. The final values are then compared to independent check point measurements [4]. MTLS data registered to a geodetic datum by a licensed land surveyor cannot be modified except under the direction of a licensed land surveyor. Thus, there needs to be a chain of custody for MTLS data.[1]

Figure 1.1: Caltrans MTLS system (Trimble MX8)

The California Department of Transportation (DOT) (Caltrans) has an MTLS system, the Trimble MX8 (see Figure 1.1). The successful implementation of the MTLS system into Caltrans business practices has resulted in an exponential increase in data collected from the system [3]. Current Information Technology (IT) infrastructure systems (both hardware and software) are insufficient to store and transfer this additional data. The Advanced Highway Maintenance and Construction Technology (AHMCT) Research Center's prior research, development, and implementation of the MTLS system within Caltrans provided historical knowledge and background related to MTLS systems that were critical to designing and developing data management systems for short- and long-term use in District 4 (D4), as well as other Caltrans districts.

---

[1] California Business and Professions Code, Section 8726(m), http://www.leginfo.ca.gov/cgi-bin/displaycode?section=bpc&group=08001-09000&file=8725-8731

1

Caltrans D4 recently executed substantial IT upgrades and improvements to its capabilities for handling MTLS data, including MTLS data storage solutions provided by NetApp. In this research, the AHMCT Research Center leveraged its experience implementing and supporting the Caltrans MTLS system in order to support Caltrans D4 in its MTLS data management efforts. This research expanded D4's leading-edge MTLS capabilities.

Problem

Before D4's IT upgrade, the D4's MTLS datasets were stored and backed up in several external hard drives. The installation of the NetApp data storage system (herein referred to as the NetApp system) in D4 would enable D4 Right of Way (ROW) Surveys to store MTLS data in a central location. The first challenge was to locate all raw and processed MTLS datasets on the external hard drives and copy the data to the NetApp system. Furthermore, a directory structure had to be developed to organize the MTLS datasets to allow future retrieval of the data. D4 ROW Surveys management would like to reuse the MTLS data when possible. Using existing software tools to catalog the MTLS data coverage into a Geographic Information System (GIS) database is time consuming and labor intensive. An automation tool to collect MTLS data project metadata and coverage into a GIS database was needed.

Objectives

This work supported Caltrans D4 in its IT upgrade and enhancement efforts for MTLS data management. We coordinated with the statewide investigation of overall geospatial data management to avoid the duplication of effort. Management of other forms of geospatial data or investigation of other Data Management Systems (DMS) were outside the scope of this project. The main objective was getting the D4 MTLS data organized and reduce the risk of data loss.

Research tasks included:

- The assessment of D4's IT infrastructure upgrade and NetApp system configuration at the outset of the research.

- The development of requirements for a geospatial-enabled web-based portal to view MTLS data availability.

- Software system design, including object diagrams, schema, testing, and the implementation of the customized software.

- The implementation and installation of software in D4.

- Recommendations for backup options.

- Documentation of project efforts and systems, including beta test efforts, testing procedures, workflow diagrams and quality assurance/quality control (QA/QC) documentation.

<u>Scope</u>

Project plans included:

1. Workflow and recommendations related to the use of the NetApp system for the storage of MTLS data. The secure transfer of existing MTLS data from portable storage to NetApp system storage was the highest priority to prevent any loss of data.

    a) The storage of raw data and Global Navigation Satellite Systems (GNSS) base station data.

    b) Design of directory structures for raw and processed data.

    c) Storage for processed data and exported data (images and point clouds).

    d) QA/QC sample data on the NetApp system storage.

    e) Development of an MTLS data flow diagram.

    f) Backup recommendations

    g) An evaluation of the compression feature provide by the NetApp system and recommendations for its use.

2. Developing software to automatically extract metadata from MTLS data to aid in MTLS data management.

    a) Develop a list of metadata that can be retrieved from the MTLS data.

    b) Investigate the availability of additional metadata such as the vehicle path, time and date of collection, scan coverage area, estimated errors, etc.

    c) Design and develop a tool to "crawl" over the data directory structure to extract and catalog metadata.

    d) Test and QA/QC metadata extraction software, and modify software as needed.

    e) Provide metadata extraction software source code to D4.

f) Provide metadata in an open, common format for use in GIS that aid in the visualization and retrieval of MTLS data from the NetApp system in Caltrans' internal network.

3. Perform brief literature and product reviews related to the storage and distribution of MTLS data.

a) Perform literature and product reviews of web-based MTLS data retrieval and the viewing of MTLS data (Light Detection And Ranging (LiDAR) and images):

- Environmental Systems Research Institute (ESRI, also Esri)

- UPLAN of Utah DOT (UDOT): http://uplan.maps.arcgis.com/home/ and http://udot.uplan.opendata.arcgis.com/datasets/2c718a780caf47279fbc23c3401af4d7_0

- OpenTopo: http://www.opentopography.org/

- GRiD: https://griduc.rsgis.erdc.dren.mil/griduc/corpsmap/

- AHN2: http://3dsm.bk.tudelft.nl/matahn

Research Methodology

AHMCT met with D4 personnel to identify their remaining MTLS data storage and management needs. Site visits and continued discussion helped AHMCT refine its understanding of D4's needs, as well as identify the configuration of relevant D4 systems at the outset of the research. AHMCT then used the existing configuration and D4's clearly stated needs to develop rigorous requirements to guide the remaining development and implementation.

Based on these requirements, AHMCT developed software to crawl MTLS data directories and files and automatically extract metadata about each MTLS project in the directory structure. The MTLS GIS data is stored in a PostGIS database and can be made available via a geospatial-enabled web-based portal to view MTLS data availability. This tool shows where MTLS scans have been performed, making it easy to select MTLS data files for viewing and post-processing using map-based visualization of the MTLS scan coverage areas.

Ancillary to the above primary research effort, AHMCT also reviewed existing MTLS data management and visualization software tools. AHMCT also developed MTLS workflow and QA/QC procedures to enhance D4's MTLS data management. Finally, AHMCT provided

recommendations for key issues in MTLS data management, including file formats, metadata, data set organization, directory structure, and data backup.

CHAPTER 2:
DISTRICT 4 CONFIGURATION AT START OF RESEARCH

Background

Previously, D4 MTLS data storage and backup filled nine 4 TB USB 3.0 Seagate external hard drives. These external drives are generally slower than internal workstation drives. Some of the disc drives were bare internal hard drives, specifically eleven 6 TB SATA Seagate internal 3.5" hard drives. One additional 4 TB USB 3.0 Seagate external hard drive contained an MTLS dataset for the Santa Clara County Overhead Sign MTLS project collected by a Caltrans consultant. Users relied on labels and Microsoft (MS) Excel to keep track of the data locations of MTLS project data. The process was cumbersome and unreliable. Large central data storage with high network bandwidth was needed. Caltrans D4 ROW Surveys received a Federal grant from the Highways for Life program. The grant enabled D4 to upgrade their IT infrastructure to better handle the large MTLS point cloud data. D4 Surveys and D4 IT, in cooperation with Caltrans Headquarters (HQ) IT, successfully procured and completed the IT infrastructure upgrade. The three major IT infrastructure upgrade areas included:

1. Data storage

2. Network bandwidth

3. Workstations

Network Data Storage Upgrade

NetApp System Configuration as Installed in D4

Both NetApp storage solutions and EMC Corporation Isilon[2] storage solutions were considered. The NetApp FAS8040 system was selected and procured. The detailed NetApp FAS8040 components procured are listed in Appendix A. A NetApp FAS8040 hybrid storage array is capable of housing a maximum of 720 disk drives with a 5760 TB maximum raw storage capacity. Detailed NetApp FAS8040 specifications and features are available at http://www.netapp.com/us/products/storage-systems/fas8000/fas8000-tech-specs.aspx. Additional detailed information on FAS8000 system design is available at http://community.netapp.com/t5/Tech-OnTap-Articles/FAS8000-Scale-Out-Storage-for-the-Enterprise/ta-p/85581. The total system cost was ~ $289K. The cost included system hardware, software, shipping, installation, extended warranty, support, and training.

The FAS8040 data storage system procured by D4 has 5×24 (120) 3 TB 7,200 revolutions per minute (RPM) disk drives (360 TB raw storage capacity). Some of the hard drives are used as spare hard drives and the remaining hard drives are divided into Redundant Array of Inexpensive Disks (RAID) groups. According to NetApp personnel, the current system configuration would have 98 usable hard drives (3 TB with ~ 2.5 TB usable storage space per

---

[2] http://www.emc.com/en-us/storage/isilon/index.htm

drive). The current NetApp system provides 245 TB usable data for data storage. Figure 2.1 shows the NetApp FAS8040 system installed in the D4 server room. The current data storage rack for the NetApp system has room for additional disk arrays. The FAS8040 can be expanded by adding additional disk arrays. In addition, more data storage space can be added by adding an additional rack.

Figure 2.1: NetApp System

Preliminary storage requirements for District 04 MTLS data sets

During a meeting held on October 15, 2015 between D4 Surveys, D4 IT, HQ Division of Research, Innovation and System Information (DRISI), NetApp application engineers, and AHMCT staff, the team decided to set up two volumes named **mtls_raw** (50 TB, for raw MTLS data as collected from the MTLS system) and **mtls_pro** (75 TB, for post-processed MTLS data). Currently, D4 has about 20 to 30 TB of raw MTLS data and 30 to 45 TB post-processed MTLS data. A 50 TB mtls_raw volume for storing raw MTLS collected data and a 75 TB mtls_pro volume for storing MTLS post-processed data were created based on the team's decision, the current MTLS data estimate, and anticipated future data collection. The storage volumes can be expanded to a maximum size of 100 TB using current NetApp hardware and software or reduced without any adverse effect on the data, and without system shutdown. The NetApp system has data redundancy to guard against data loss due to hard drive failure.

According to NetApp application engineers, the animation movie industry uses NetApp systems for their data. NetApp application engineers asked about the numbers of small files in the MTLS datasets. In their experience, certain animation movie industry applications generate a large number of small image files which would require certain NetApp system configuration changes. Caltrans MX8 MTLS datasets store the photo images in Audio Video Interleave (AVI) file format by combining hundreds of images into a single AVI file 1 GB in size. As a result, the NetApp system MTLS data volumes were configured using typical NetApp system settings for data storage. In other words, the NetApp application engineer did not make any specific NetApp system RAID and software configuration modifications for MTLS data. In addition, the NetApp system scalability was not affected.

The NetApp system's Snapshot feature functionality is enabled for the mtls_pro volume to provide rollback capabilities for the processed volume. The Snapshot feature adds about 5% data overhead to data storage usage. NetApp's Snapshot technology enables the user to automatically create "point-in-time" copies of file systems. Users can revert/restore files to the previous version. As a result, MTLS post-processing personnel may revert to previous version of the MTLS file(s) online using the Windows file manage without the involvement of D4 IT. Details of NetApp's Snapshot technology are available at http://www.netapp.com/us/media/ds-2477.pdf.

The current NetApp system configuration provides access to the two MTLS data volumes via the Common Internet File System (CIFS). CIFS is a native file-sharing protocol in Windows. D4 Surveys accesses MTLS data using their Windows workstations via shared directories. The NetApp system can also use other file-sharing protocol such as the Network File System (NFS) and the Internet Small Computer Systems Interface (iSCSI). The MTLS data stored on the NetApp system are catalogued and separated between raw MTLS and post-processed MTLS volumes. The raw MTLS data were copied with the help of D4 IT staff. Details of the recommended file structure and data storage usage are provided in Chapter 4.

Network Upgrade

The large data transfer between the NetApp system located in the server room and the data processing workstations located on the district office's 13th floor would put addition strain on the existing network backbone. To speed up the MTLS data transfer and reduce the network strain and latency, the network bandwidth from the server room to 13th floor workstations was upgraded. Without a network upgrade, the large MTLS data transfer would saturate the network backbone between the server room and the 13th floor and would negatively impact network access for other Caltrans D4 users located on the 13th floor.

The network upgrade consisted of two parts:

1) A 40 Gigabit fiber-optic network backbone upgrade from the server room to the 13th floor network closet. This included:

   a. Adding fiber-optic cable links from the server room to the 13th floor network closet

b. Adding 40 Gigabit fiber-optic network hardware in the server room and the 13th floor network closet

c. Adding a Cisco router to support the 40 Gigabit fiber-optic network

2) A 10 Gigabit "copper" network upgrade to workstations. This included:

a. Fourteen new Category 6 copper Ethernet cables (to support the 10 Gigabit network) were added from the 13th floor network closet to D4 Surveys and GIS workstations located on 13th floor

   i. One 10 Gigabit network link to the I-Room workstations

   ii. Eleven 10 Gigabit network links to D4 Surveys workstations for MTLS data post-processing

   iii. Two 10 Gigabit network links to D4 GIS workstations

b. Four additional 10 Gigabit network links were added to D4 IT workstations to support the administration of the NetApp system.

Figure 2.2: Networking equipment in 13th floor network closet
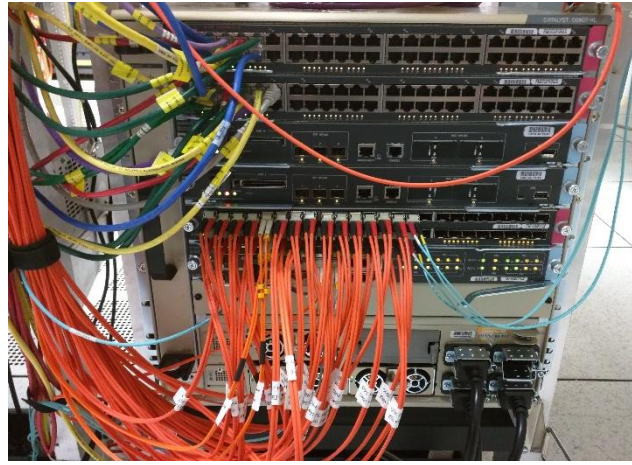
Figure 2.3: Networking equipment in the D4 server room

Workstation Upgrade

The existing standard D4 workstations did not support 10 Gigabit networking. Hewlett Packard (HP) X540 / Intel X540 10 Gigabit Ethernet network cards were added to support the 10 Gigabit Ethernet network interface. In addition, eight new HP Z840 workstations were procured to enhance the MTLS data post-processing speed and reduce staff time. Key HP Z840 workstation specifications are provided in Table 2.1.

Table 2.1: Key HP Z840 workstation specifications

CPU	Single Intel Xeon E5-2637v4 3.5 2400 4 Core
RAM	128 GB DDR4 4-2133 (8x16 GB) registered ECC
First Hard Drive	512 GB SATA SSD
Second Hard Drive	4 TB 7200 RPM SATA
Third Hard Drive	4 TB 7200 RPM SATA
Fourth Hard Drive	4 TB 7200 RPM SATA
Video Card	NVIDIA Quadro M4000 8 GB
10 Gb Network Card	X550 10 Gigabit network card (The Intel X450 has been obsoleted by Intel and replaced by the X550 for the eight new workstations)

CHAPTER 3:
DISTRICT 4 MTLS DATA MANAGEMENT NEEDS

Background

This research supported Caltrans D4 in their IT infrastructure upgrade deployment and enhancement efforts for MTLS data management and visualization. Caltrans' upgrades and enhancements include a NetApp system storage solution for MTLS data. At the beginning of the project, AHMCT researchers worked with D4 ROW Surveys personnel to identify work needed to best manage, locate, and secure existing and future MTLS data. Due to limited time and resources, the project focused strictly on MTLS data. Other point cloud data and CAD deliverable data were not considered in this research project. The two primary project objectives from D4 were securing existing MTLS data and developing a tool to allow users to catalog and find MTLS data for use in projects. This tool was meant for D4 Surveys and their customers. The tasks to achieve the project goals are listed below.

Implementation to Secure MTLS Data

1. A directory/folder structure was designed first to aid in data discovery and proper organization before copying the MTLS data onto the MTLS data volumes on the NetApp system.

2. The MTLS data was copied onto the NetApp system from several portable external hard drives. The MTLS data on the portable drives can serve as one backup.

3. A random sample of data were verified for integrity using the file compare software tool (FC.exe) built-in to Windows 7 to compare the files on the NetApp system to the original files. In addition, effort was made to ensure all MTLS data was copied onto the MTLS data volume on the NetApp system.

4. The built-in compression feature of the NetApp system was evaluated for use with MTLS data.

5. AHMCT developed and recommended a set of Access Control and Permission settings for raw MTLS and processed MTLS data volumes.

6. AHMCT performed a literature review and cost comparison of cloud-based backup options.

Implementation to Catalogue and Manage MTLS Data

Surveys staff must be able to efficiently identify and retrieve MTLS data to extract essential data for customers. The following tasks were designed to accomplish this goal:

1. Perform a literature review of available software and systems for LiDAR data management and visualization

2. Assess current D4 GIS data management and visualization systems in D4

3. Develop software to automatically extract metadata from MTLS data for integration into a GIS platform, which enables:

    a. A user to locate, manage, and retrieve MTLS data directly from the NetApp system.

    b. A user to perform GIS data analysis using ESRI ArcGIS tools to produce performance metrics if they are requested by Caltrans management.

4. Test MTLS GIS data extraction software in D4

5. Integrate MTLS GIS data to existing D4 GIS infrastructure:

    a. To allow D4 users to access the GIS data using ESRI software tools

    b. To provide MTLS GIS data visualization via Web browsers using D4 internal GIS infrastructure

    c. To use D4's existing GIS and database servers and IT infrastructure, eliminating the need for additional servers.

CHAPTER 4:
MTLS WORKFLOW, STORAGE, AND BEST PRACTICES

Introduction

Caltrans' MTLS data sources include:

1. Caltrans' Trimble MX8 MTLS system operated by Caltrans surveyors [3]

2. Caltrans contactors: Depending on the contract deliverables, the contactor provides Caltrans MTLS data in an open Log ASCII Standard (LAS) file format. In some cases, the contractor provides MTLS data files in the MTLS system manufacturer's proprietary format in conjunction with LAS files.

3. The Riegl VMX-1HA MTLS system operated by Caltrans surveyors. The Riegl VMX-1HA system is online after September 2016.

The LAS file format is an open standard file format that can be read by various point cloud data post-processing software. However, each MTLS system manufacturer uses its own proprietary data format that requires its own proprietary software for post-processing and ground target registration. The majority of the Caltrans consultants' MTLS systems are made either by Optech or Riegl.

Each Caltrans district is responsible for storing and processing its own MTLS data. The raw MTLS data size depends on the project size, number of lanes, vehicle speed, number of passes, number of on-ramps and off-ramps, as well as the specific MTLS system. A detailed MTLS data size estimation calculation is available in the Mobile Terrestrial Laser Scanning Workflow Development, Technical Support and Evaluation final report [3]. Using northern California MTLS deployment data (18 month period), the average number of projects is about 1.25 projects per district per month [3]. The average project size has been 5 centerline miles. The average MTLS data is about 50 GB/highway centerline mile [3]. The estimated average yearly raw MTLS data growth is about 3.75 TB per year for each district. The post-processed MTLS data is about 1.5 times of the raw MTLS data. Therefore, the combined growth rate for both raw and post-processed MTLS data is about 9.4 TB per year. For simplification, 10 TB per year MTLS data growth rate is used for the data backup calculation. This data growth estimate only accounts for pavement survey project data and may vary among the districts because of the district size and the number of capital projects. The MTLS data size growth estimate does not account for data of special projects such as the D4 overhead signs (OHS) project. D4 surveyors performed MTLS scans on all major routes in six counties for the OHS project within three months. The raw and processed MTLS data are stored in eleven Seagate 6 TB Serial AT Attachment (SATA) internal 3.5" hard drives. The addition of asset management MTLS projects would significantly increase the MTLS data storage demand.

Table 4.1: MTLS data growth rate estimate summary

Average # projects per month per district	Average # projects per year per district	Average project data size	Estimated raw MTLS data growth per year
1.25	15	250 GB / project	3.75 TB/ year

MX8 MTLS Workflow Summary

A high-level overview of the MTLS system data flow diagram is presented in Figure 4.1. After the data collection and transfer from the MTLS vehicle computers, the GNSS/IMU data is first post-processed with the local GNSS base station(s) and/or Continuously Operating Reference Station (CORS) data in the vicinity of the project site using Applanix Position and Orientation System Postprocessing Package (POSPac) software. The best estimated vehicle trajectory data (files named "sbet_Mission 1.out" and "smrmsg_Mission 1.out") resulting from the GNSS/IMU post-processing is then used to update the Trimble .GPS files and *.LAS 2.0 or *.LA20 files using Trimble Trident software. Trident is then used to "Colorize" the point cloud with the camera images (stored in AVI format). After the point cloud is registered to the ground control targets using Trident, feature extraction for the final deliverable can begin using Trident or other point cloud feature extraction software such as TopoDOT using the LAS files and camera images exported from Trident. The exported LAS files may also be used to create fly-through animation movies.

Some desirable MTLS metadata, such as coordinate epoch, vertical datum, and project request information, are not stored in the MTLS data files. Office of Land Surveys, MTLS users, and AHMCT researchers are developing a consensus based standardized data format and protocol to record these MTLS metadata for both Caltrans MTLS systems (MX8 and VMX-1HA). Such work is outside the scope of this research.
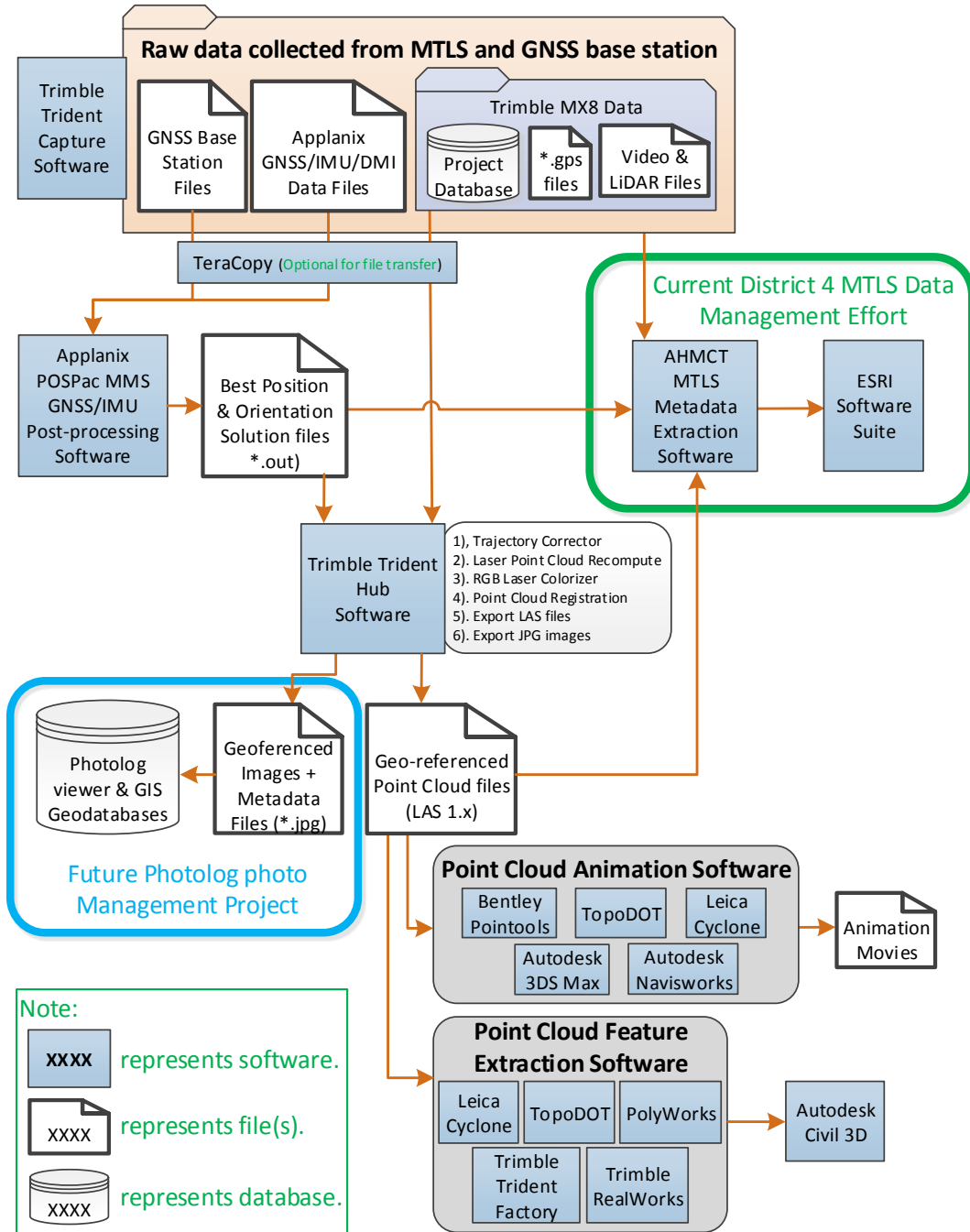
Figure 4.1: MTLS data post-processing data flow diagram

<u>**MX8 MTLS Data Structure**</u>

Previous MTLS research provided MTLS project name guidelines and recommendations. The MX8 MTLS data are stored in a single project folder using the following naming conventions. The majority of the D4 MX8 MTLS project folders were named using these conventions.

Project folder naming methods:

a) CoRte_PM Beg-PM End_YearMonthDay C or U

b) Postmiles (PM) rounded to full mile – no decimals

c) 2 or 3 letters (county) 3 digits (route)_3 digits (begin postmile)-3 digits (end postmile)_4 digits(year)2 digits(month)2 digits(day) C or U

d) e.g. ALA080_001-010_20121217U and SJ580_000-005_20130128C

    i. ALA – County

    ii. 80 – Route

    iii. 001 begin PM

    iv. 010 end PM,

    v. 20121217 – Date, year 2012, month 12 (December), day 17

    vi. U – Uncontrolled and C – controlled

Within the MTLS project folder, there are several sub-folders containing the camera and LiDAR data, as shown in Figure 4.2. The MTLS project folder includes the Applanix POSPac data folder, project database, GNSS base station data folder, ground control data, camera image data folders, and LiDAR data folders, as well as a folder containing the exported LAS point cloud and Joint Photographic Experts Group (JPEG) image files. Table 4.2 listed the files and file extension of the MX8 MTLS data files used to store the raw and processed MX8 MTLS data.

Table 4.2: MX8 MTLS data format summary

MX8 MTLS data files	Description	Proprietary or open file format
*.xxx	Applanix GNSS/IMU raw data	Proprietary
sbet_Mission 1.out smrmsg_Mission 1.out	Applanix POSPac best estimated vehicle trajectory data and estimated error output	Open
*.T02 RINEX	GNSS base station data	*.T02: Trimble proprietary Receiver Independent Exchange Format (RINEX): open standard
*.mdb *.tridb	Trimble MX8 project database	*.mdb: Microsoft Access (proprietary) *.tridb: SQLite (open)
*.LAS *.LA20	Trimble MX8 MTLS laser data	*.LAS: open LAS 2.0 format *.LA20: proprietary
*.gps	Trimble MX8 MTLS vehicle position data for each "run" record	Proprietary
*.AVI	Trimble MX8 MTLS camera image data	Open
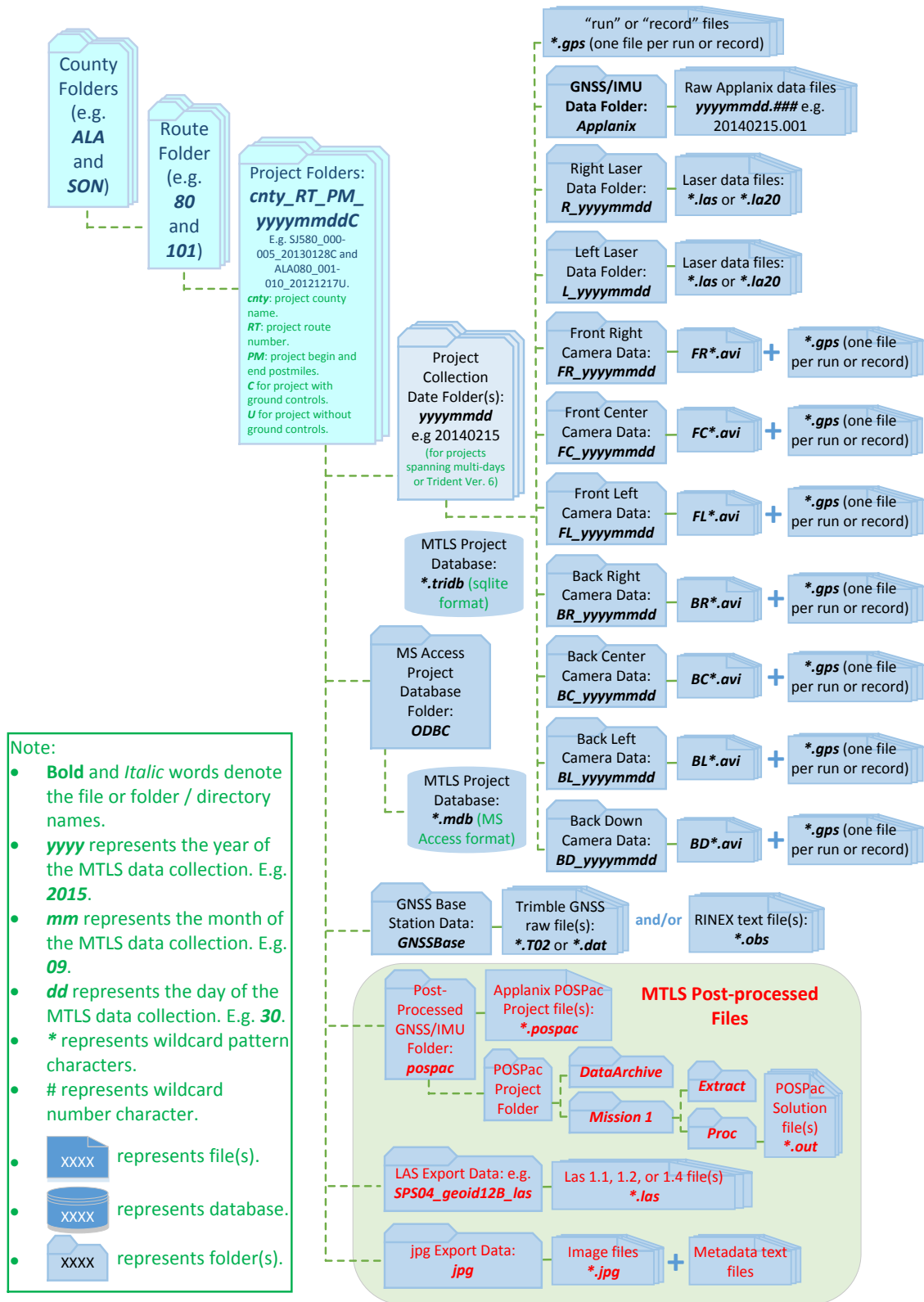
Figure 4.2: D4 Trimble MX8 / Trident MTLS data directory structure

Data Set Organization and Folder Setup

The directory structure for the **mtls_raw** and **mtls_pro** volume is based on county and route, as indicated in Figure 4.2. The directory structure was designed to aid in data discovery and proper organization in the raw and post-processed MTLS data volumes on the NetApp system. The root MTLS volume folder contains D4's nine county folders named using the standard Caltrans County abbreviation (ALA, CC, MRN, NAP, SCL, SF, SM, SOL, and SON), as shown in Figure 4.3. The county abbreviations located in D4 are listed in Table 4.3. Each county folder contains multiple folders named using the route numbers (e.g., 1, 4, 101, 580, 80, and 880) located within each county. MTLS pavement elevation project folders are copied to and stored in the route folders depending on the project's county and main route location. Since the D4 OHS MTLS projects cover multiple routes, they are copied to and stored in the county folders.
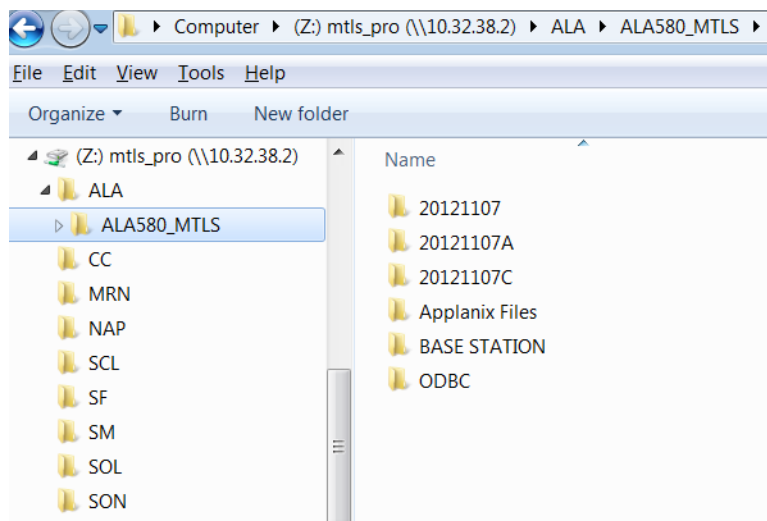
Figure 4.3: D4 root MTLS data directory structure

Table 4.3: Caltrans District 4 county code, name, and abbreviation

California county code	County name	County abbreviation
01	Alameda	ALA
07	Contra Costa	CC
21	Marin	MRN
28	Napa	NAP
38	San Francisco	SF
41	San Mateo	SM
43	Santa Clara	SCL
48	Solano	SOL
49	Sonoma	SON

18

NetApp System Administration and Access Control

D4 IT is responsible for the administration of the NetApp system, including monitoring system integrity and assigning access to the two MTLS data volumes using Caltrans Active Directory permissions and access. D4 Surveys management is responsible for identifying individuals who require access to the MTLS data volumes based on their job functions and responsibilities with Surveys. Access permission recommendations are provided in Table 4.4. Note that a single individual may be assigned to have more than one role or job responsibility. For example, a staff member may be responsible for both operating the MTLS system during data collection and post-processing MTLS data. In this case, they will be required to have read and write access to both MTLS volumes.

Table 4.4: Caltrans District 4 MTLS data volumes access permission recommendations

Personnel role/Job description	mtls_raw volume	mtls_pro volume
Surveys manager	Read	Read
MTLS data collection operator	Read and write	None
D4 MTLS data post-processing staff	Read	Read and write
MTLS feature extraction staff	None	Read

Survey data of the major bridge structures in California such as the San Francisco–Oakland Bay Bridge are considered security sensitive. Special limited read permission may be required for such projects based on the data security policy set by authorities.

Workflow

1. The MTLS data collection operator copies the raw MTLS data from the MTLS system to a portable storage drive.

2. The MTLS data collection operator copies the raw MTLS data from the portable storage drive to the MTLS project's associated county and route folder located in the **mtls_raw** volume on the NetApp system.

3. The MTLS data collection post-processing staff member copies the post-processed MTLS data including exported LAS files to the MTLS project's associated county and route folder located in the **mtls_pro** volume on the NetApp system after the completion of the post-processing process.

NetApp System Online Compression Evaluation

AHMCT, D4 Surveys, and D4 IT have evaluated the NetApp system's online compression feature using MTLS and static laser scan data. The first compression evaluation used a 50 GB Leica Cyclone static laser scan data (*.imp) file. The NetApp system's online compression achieved a savings of 7.8% in data storage. The second compression evaluation used a 924 GB Trimble MX8 MTLS dataset (OHS_NAP_20140729 MTLS project), and the NetApp system's online compression achieved a savings of 12% in data storage (115 GB).

Table 4.5: NetApp system online compression test results

Point cloud file type	Original data size	NetApp system data size after compression	Percentage saving
Leica Cyclone data	50 GB	46.1 GB	7.8 %
Trimble MX8 MTLS data	924 GB	809.64 GB	12 %

Recommendations

- Since the online compression did not achieve a large storage savings in these limited tests, it was not employed in either **mtls_raw** or **mtls_pro** volumes.

- To reduce the exported LAS file size, the exported LAS files may be compressed into LASZip format. The Riegl MTLS post-processing software supports direct export to LASZip format. The current version of the Trimble Trident MTLS post-processing software does not support LASZip format export.

- Other popular compression formats, such as rar, 7zip, and zip, have been evaluated for MTLS datasets. The compression time for a MTLS project dataset ranges from 1 hour to several hours depending on dataset size. The storage space saving ranges from 9 to 25 percent. Due to the time-consuming nature of the compression and decompression processes, AHMCT researchers do not recommend the use of compression of the MTLS dataset (stored in the manufacturers' proprietary data format) at this time.

MTLS Data Backup

Current Status

Currently, the MTLS data are backed up using external hard drives stored in D4 and at AHMCT. Other backup methods should be explored and their costs compared. There is no standard Caltrans MTLS data retention policy for the raw and post-processed MTLS data. Future MTLS data management efforts should develop a statewide MTLS data retention policy based on input and consensus from all Caltrans districts and HQ. Such work is outside the scope of this research. The current NetApp system should provide sufficient storage capacity for D4 MTLS data for three to five years or more.

Caltrans D4 possesses approximately 20 TB of raw and 30 TB of processed MTLS data with an expected future data growth rate of about 10 TB/year total (raw and post-processed). This data is stored in a single location on a NetApp system appliance on-site at the D4 district office. Data integrity is integral to data warehousing. The NetApp system ensures internal data integrity through specific mechanisms, but the data is still ultimately housed in a single location. Thus, it is susceptible to localized catastrophic events. Mitigation involves the backup and storage of critical data in remote locations. MTLS data backup requires no different strategy from other digital data. A Caltrans MTLS data retention policy may mandate different data backup requirements for the raw and post-processed MTLS data.

The following options were studied and evaluated:

1. Replication to identical hardware in a remote location such as a D4 satellite office

2. Backup to major commercial clouds provided by Amazon, Google, and Microsoft

3. Backup to California Cloud Services (CalCloud)

4. Backup to Caltrans HQ

Table 4.6: Current MTLS data storage requirements

	Size (GB)
RAW MTLS Volume	20,000
PRO MTLS Volume	30,000
Total Yearly Growth (GB/year)	10,000

Assumptions

Table 4.6 summarizes the current data storage requirements for the MTLS data presently stored on the NetApp system hardware located in the Caltrans D4 district office. For the purposes of this analysis, all values were calculated based on a static data storage and backup size of 50 TB. The costs and durations can be scaled linearly for data sizes larger than 50 TB.

Cloud-Based Data Backup Cost

Table 4.7: Cloud storage fees (based on 50 TB of MTLS data)

Company	Storage ($/mo)	Storage ($/yr)	% higher from the lowest cost option
Amazon	$350.00	$4,200.00	0.00%
Google	$500.00	$6,000.00	42.86%
Microsoft	$500.00	$6,000.00	42.86%
CalCloud	$8,050.00	$96,600.00	2200.00%

All three major commercial cloud backup services charge a monthly fee for their services. Detailed pricing information links are available in Appendix D. Table 4.7 provides a summary of the data backup/archival cloud storage fees for the major commercial players (Amazon, Google, and Microsoft), as well as the CalCloud for state agencies. It summarizes the total cost per month to store 50 TB of data in deep archive (i.e., where the data is very infrequently accessed or modified). Detailed cost calculations are available in Appendix D. Some providers require a minimum commitment of 1-3 months for all data writes, however this seems immaterial given Caltrans' intended deep archive use case. The CalCloud service is the highest cost option. The service cost for CalCloud for two years would allow D4 to purchase a second NetApp system (~ $250,000) and install it in a remote satellite office, thereby addressing the location diversity issue. The yearly storage costs on CalCloud quickly approach the cost of redundant in-house storage hardware. The three commercial options are competitively priced. The Amazon service is

the lowest cost option. The prices are current as of this report but are subject to change without notice.

Cloud-Based Data Restoration Cost

All three major commercial cloud backup services also charge a fee for data transfer when uploading the data and restoring data. Table 4.8 summarizes the transfer costs of 50 TB between the commercial cloud services and a local data storage device. The backup upload fees are minimal. However, the full backup restoration fees are quite large compared to the typical monthly storage fee. The restoration fee is charged when data has to be restored to a local storage system from the commercial cloud. The use case of D4 MTLS data backup is for catastrophic disaster recovery and will rarely, if ever, be used. The primary NetApp system ensures that losses due to user error can be recovered with local data snapshots. In addition, NetApp system data redundancy protects against data loss due to hard drive failure. All three major cloud backup services' data restoration fees are very competitively priced. However, the recovery time due to each service's data transfer rate varies dramatically. Based on real-world tests,[3] the Google service can transfer the entire 50 TB archive in about seven days. The Amazon service, however, would take approximately 18 months. Table 4.7 shows that Amazon clearly has the lowest monthly storage fee, but the complete retrieval time of the entire MTLS datasets takes the longest (~18 months). Amazon does support higher transfer rates for additional fees, but the cost would increase by $10,000 to $100,000 depending on the data transfer rate. The low-cost data transfer rate option of the Amazon service is not practical, based upon the unrealistic disaster data restoration recovery time requirements and the clear need for relatively quick data restoration.

Table 4.8: Cloud backup restoration costs and duration

Company	Upload			Retrevial		
	Cost	Rate (MB/s)	Duration (hrs)	Cost	Rate (MB/s)	Duration (hrs)
Amazon	Free	23.7	586.0	$4,550.00	1.45	9578.5
Google	Free	59.3	234.2	$4,700.00	122.3	113.6
Microsoft	Low	20.5	677.5	$4,450.00	27	514.4

Another primary data backup possibility is the purchase of another NetApp system similar to the existing unit located in the Caltrans D4 district office. This second system would then be installed in a separate location. At the time of this writing, similarly equipped hardware is priced at approximately $250k. Duplicate storage hardware has several advantages, including availability of existing administrative staff familiar with the primary storage unit, easy data/structure modifications, inexpensive backup verification, and high-speed data transfers.

---

[3] http://blog.zachbjornson.com/2015/12/29/cloud-storage-performance.html

Summary

Compared to the yearly costs of commercial cloud services, the purchase of another NetApp system would require more than twenty years for a return on the investment. Duplicating another similarly sized storage system(s) at another Caltrans district or HQ would incur a cost similar to buying another NetApp system. Limited Caltrans backbone bandwidths would have to be upgraded to make the duplicate system option feasible for data backup and recovery. The additional costs of hardware support, training, maintenance, and administrative support are not readily quantifiable. The State of California CalCloud storage solution is not a viable option due to its 16x cost over commercial solutions. The Google (Nearline) storage service is recommended for MTLS data backup and disaster recovery, based on the typical D4 use case of a one-time backup and the long-term archive of MTLS files.

CHAPTER 5:
REVIEW OF EXISTING WEB-BASED GIS AND LIDAR DATA VISUALIZATION SYSTEMS

Introduction

Both Caltrans D4 Right of Way and the DRISI Office of Data Services and Technology, Geospatial Enterprise Operations Branch have published GIS maps on their intranet and internet websites. Figure 5.1 shows the publicly available web graphical user interface (GUI) GIS maps on arcgis.com published by Caltrans D4 Right of Way. Figure 5.2 displays the internet web GUI GIS maps published by the DRISI Office of Data Services and Technology, Geospatial Enterprise Operations Branch. MTLS data footprints could be viewed through existing geospatial portals.
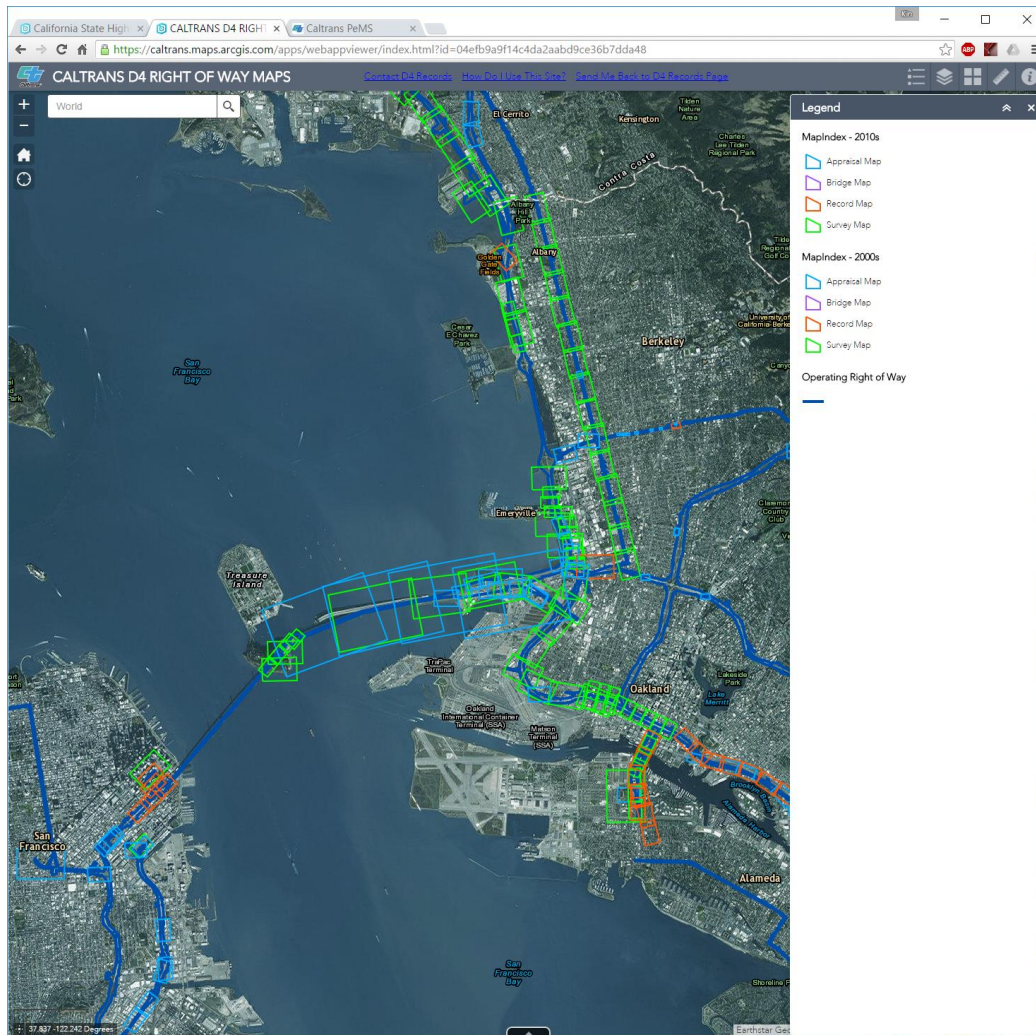
Figure 5.1: Caltrans D4 Right of Way maps on arcgis.com[4]
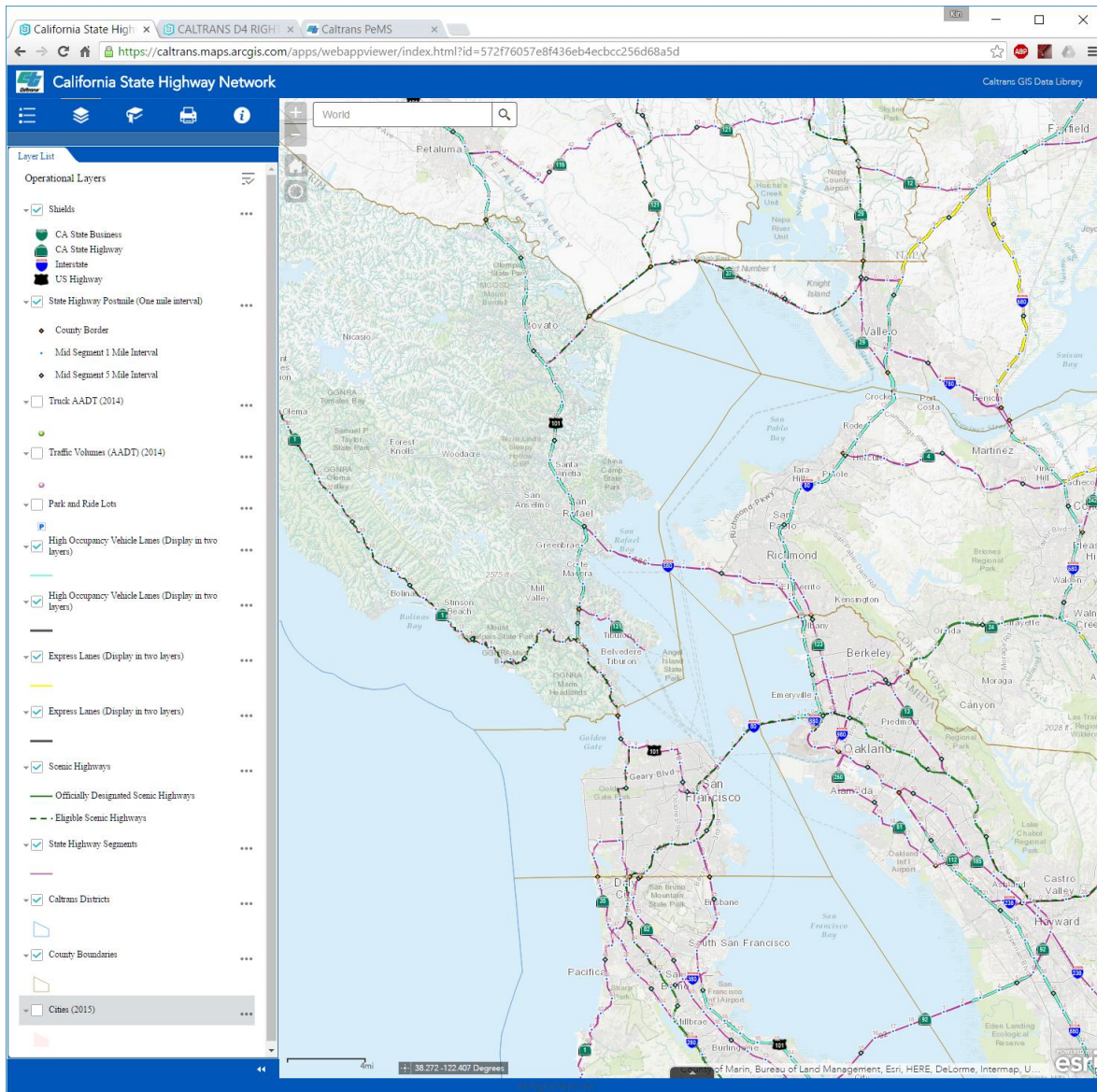
Figure 5.2: DRISI Office of Data Services and Technology, Geospatial Enterprise Operations Branch maps on arcgis.com[5]

ESRI continues to enhance and extend their software support for LiDAR point clouds. The Arcgis.com online services would likely support 3D point cloud viewing in a web browser in the future. There are two currently available open-source WebGL HTML 5 web browser-based 3D point cloud viewers, namely plas.io (http://plas.io/) and potree (http://potree.org/). They support both LAS and LASzip file formats as well as HTTP point cloud streaming (see http://www.pdal.io/workshop/pdal-introduction.html#what-is-pdal for details). Caltrans has not currently committed to pursuing web-based 3D point cloud viewing. The current research addressed technical feasibility; implementation planning is left for future work.

Websites have been built to provide a web GUI for finding and downloading point cloud data directly. The National Land Survey of Finland has opened all of its topographic data for use free of charge via their website (https://tiedostopalvelu.maanmittauslaitos.fi/tp/kartta?lang=en). Aerial LiDAR point cloud data is available for certain regions of Finland. The open data provided by the National Land Survey of Finland is licensed under a Creative Commons Attribution 4.0 International License.[6] However, no technical detail is available about the software tools and system architecture used by the Finnish website. The Geospatial Repository and Data (GRiD) Management System,[7] OpenTopography,[8] UPlan,[9] MnTOPO,[10] and Actueel Hoogtebestand Nederland (AHN2)[11] have also created websites to display their LiDAR data availability and made the point cloud data available for download. Some technical details are available on the software tools and system architectures used by the websites via their publications and software repositories at open-source hosting sites.

Geospatial Repository and Data (GRiD) Management System

According to the GRiD Management System website,[12] the Geospatial Repository and Data (GRiD) Management System was developed through a partnership between the U.S. Army Corps of Engineers Cold Regions Research & Engineering Laboratory (CRREL) and the National Geospatial-Intelligence Agency's (NGA) InnoVision directorate to efficiently warehouse and distribute both three-dimensional data, such as LiDAR data, and related and derived two-dimensional geospatial products, such as imagery and digital elevation models (DEMs).

For a data provider, GRiD offers data storage, support for multiple data formats, and Oracle-enabled secure user access control. For a data user (registration and login required), GRiD provides an efficient web-based download mechanism, user-defined filters, and built-in terrain analysis algorithms. GRiD currently hosts about 107 TB of LiDAR data (mostly airborne) from various public agencies. Figure 5.3 shows the GRiD web GUI. Based on our teleconference meeting with GRiD personnel, GRiD is willing to host any Caltrans data. However, an interagency agreement may have to be developed first. GRiD has 1.5 Petabytes (PB) of data storage. They are considering using Amazon Web Services (AWS) in the future. Their website system was built using custom-developed software and mostly open-source software, including the Point Data Abstraction Library (PDAL), the Geospatial Data Abstraction Library (GDAL/OGR), and the Python programming language.

---

[6] http://creativecommons.org/licenses/by/4.0/

[7] http://lidar.io/

[8] http://www.opentopography.org/

[9] http://uplan.maps.arcgis.com/home/index.html

[10] http://arcgis.dnr.state.mn.us/maps/mntopo/

[11] http://www.ahn.nl/index.html

[12] https://rsgis.erdc.dren.mil/griduc/about/
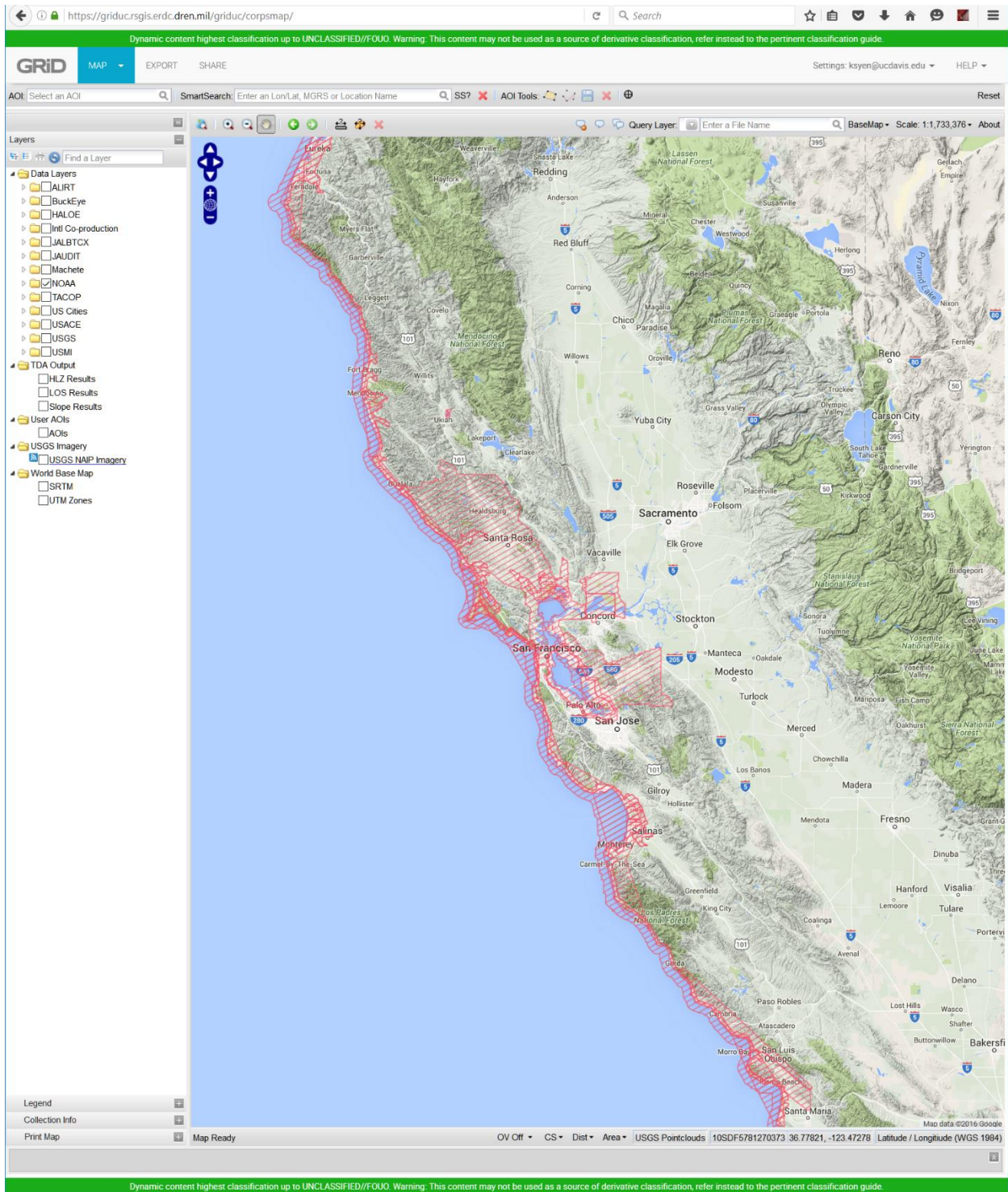
Figure 5.3: GRiD website GUI

OpenTopography

According to the OpenTopograpy.org website, OpenTopography, a National Science Foundation (NSF) data facility, is available to support NSF-funded projects that have previously acquired or are in the planning phases of acquiring LiDAR topography data. The OpenTopography facility is located at the University of California, San Diego's San Diego Supercomputer Center. It currently hosts about 200 LiDAR datasets (21 TB) covering an area of 184,000 km^2. In addition, it has 95 raster datasets (2.2 TB) covering an area of 130,000 km^2. Figure 5.4 shows the OpenTopography web GUI.

The OpenTopography system was largely built using open-source software, including libLAS, GDAL/OGR, and the free open-source ESRI Geoportal Server. Details of the OpenTopography IT infrastructure system are available in an overview paper titled "OpenTopography: A Services Oriented Architecture for Community Access to LIDAR Topography."[13]

---

[13] http://dl.acm.org/ft_gateway.cfm?id=1999327&ftid=987533&dwn=1&CFID=646483885&CFTOKEN=26190286 See also http://www.opentopography.org/publications for thesis, conference, and journal papers
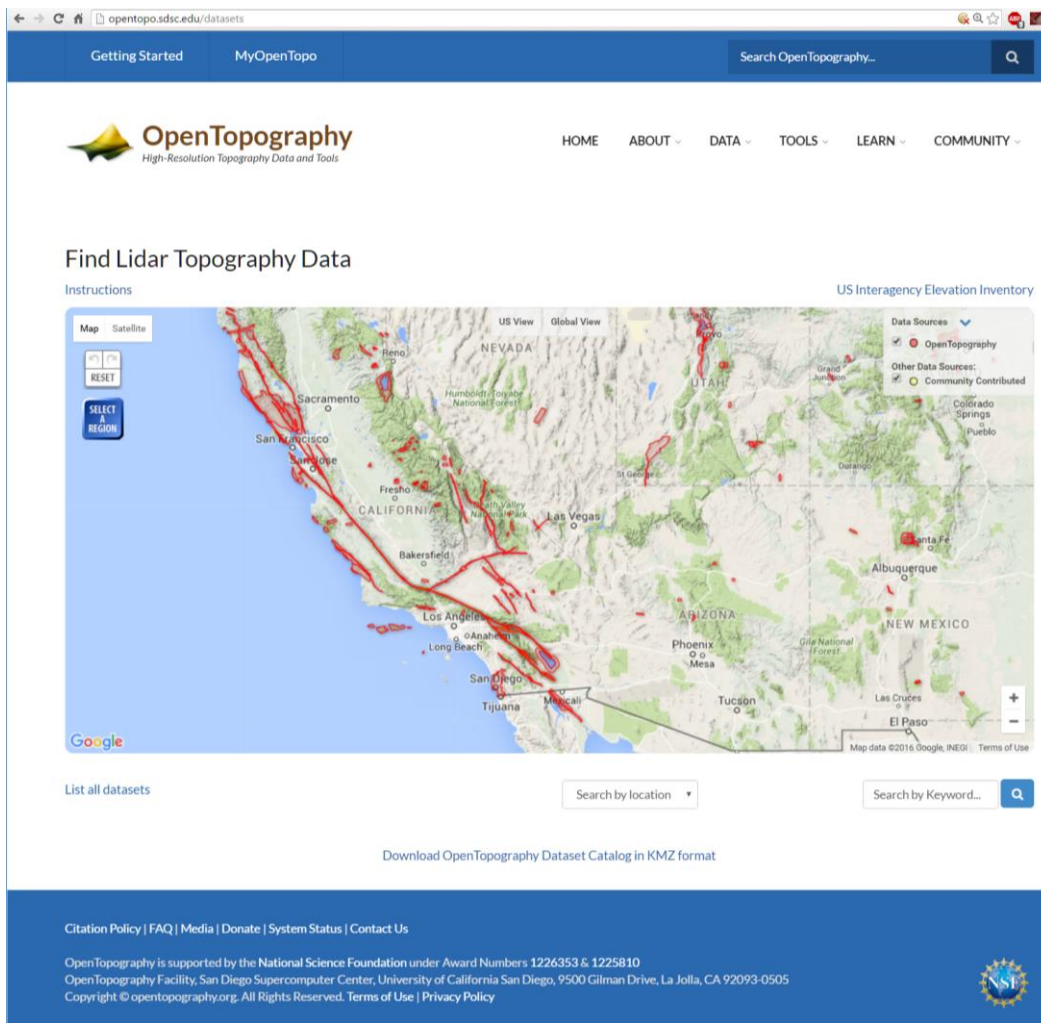
Figure 5.4: OpenTopography website GUI[14]

Actueel Hoogtebestand Nederland (AHN2)

AHN2 is the data warehouse for airborne LiDAR point clouds of the Netherlands (~ 640 billon points). The entire project software is open-source and is available on github.[15] It utilizes other established open software, namely OpenLayers, Python, PostreSQL/PostGIS, and LAStools. Users can download their airborne LiDAR data directly from the website, as shown in Figure 5.5. Detailed usage of their website is available.[16] The details of their system architecture are also available.[17] AHN2 further provides an open-source online point cloud viewer

---

[14] http://opentopo.sdsc.edu/datasets

[15] http://github.com/tudelft-gist/matahn

[16] https://3d.bk.tudelft.nl/pdfs/3dsm/matahn_3dbgt-dag_2014.pdf

[17] https://www.esciencecenter.nl/img/pressroom/PetervanOosterom.pdf

30

(http://ahn2.pointclouds.nl/) developed by the Netherlands eScience Center.[18] Their point cloud viewer source code is available.[19]
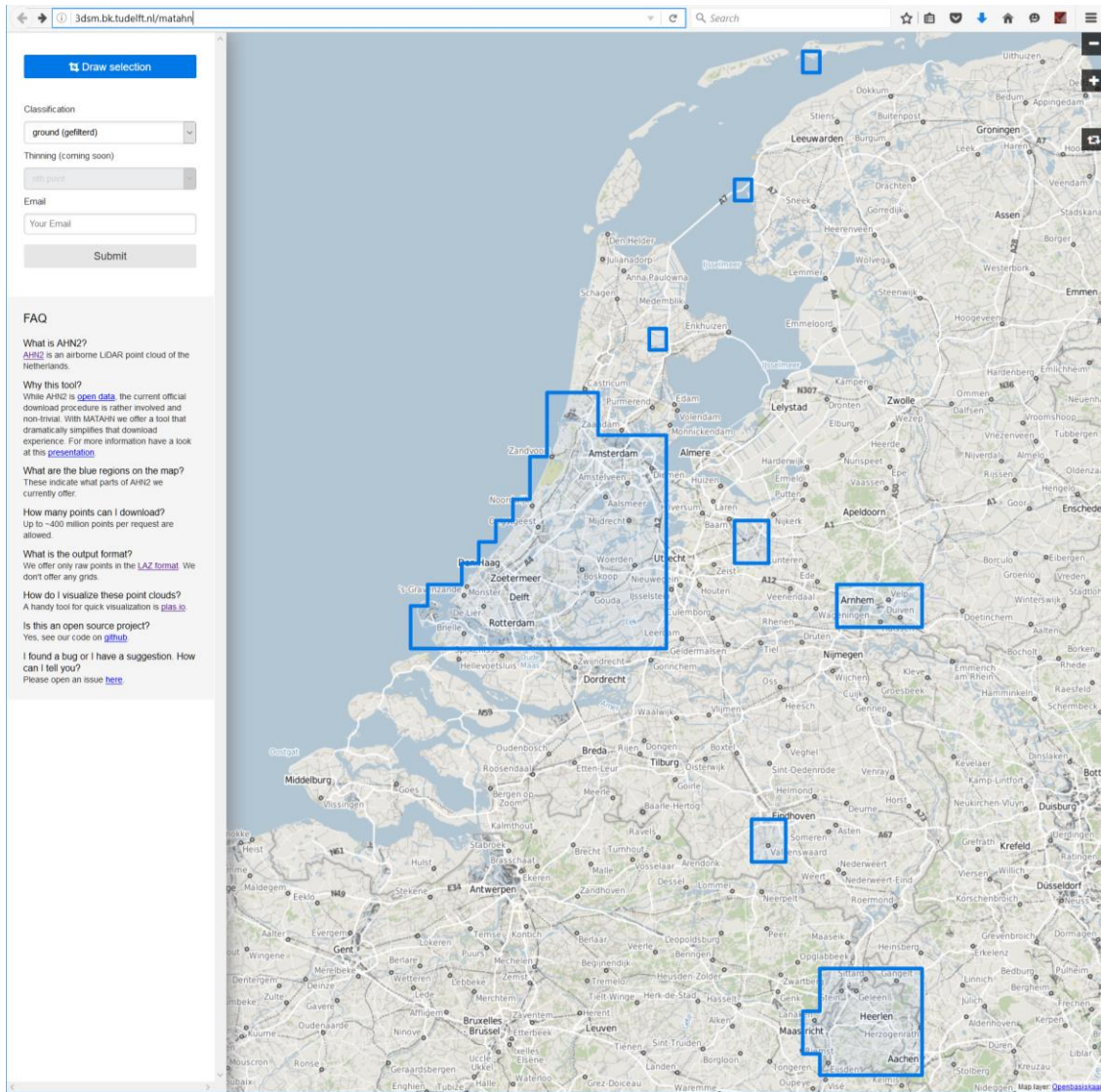
Figure 5.5: Actueel Hoogtebestand Nederland (AHN2) website GUI

Utah Department of Transportation (UDOT) UPlan

UDOT's UPlan, developed in 2009, is a collaborative, web-based tool that supports decision-making, mapping, information analysis, and planning and project development. The tool serves as a "one-stop shop" for UDOT staff, partner agencies, and the public to upload, manage, and share a wide variety of geospatial data. UPlan also facilitates synchronizing planning efforts with other state agencies, local governments, federal agencies, utility companies, and within UDOT's

---

[18] http://www.esciencecenter.nl/
[19] https://github.com/NLeSC/ahn-pointcloud-viewer

functional units. The latest version of UPlan, implemented in 2012, uses the ESRI ArcGIS Online, cloud-based platform. UPlan is available at http://uplan.maps.arcgis.com/home/. The Federal Highway Administration's Quarterly Webcast number 13, "UPlan, an Interactive, Web-Based GIS Tool Developed by the UDOT," is available[20] and provides more detailed information about UPlan. Arizona, Idaho, Montana, and Pennsylvania DOTs have modeled UDOT's efforts and developed their own version of UPlan, specifically: APLAN - Arizona,[21] IPLAN - Idaho,[22] MDT AGOL - Montana,[23] and PennShare - Pennsylvania.[24]

Table 5.1 LiDAR Website feature comparison table

Website	Web link	Software used	Software code availability
GRiD	https://griduc.rsgis.erdc.dren.mil/griduc/corpsmap/	Various open-source software	No
OpenTopography	http://opentopo.sdsc.edu/datasets	Open-source ESRI Geoportal Server, libLAS, and GDAL/OGR	Yes
UDOT UPlan	http://uplan.maps.arcgis.com/home/	arcgis.com (application programming interface available)	No
AHN2	http://3dsm.bk.tudelft.nl/matahn	Open-source software OpenLayers, Python, PostreSQL/PostGIS & LASTools	Yes

<u>**Summary**</u>

Many DOTs, including Caltrans, use the ESRI software suite for their intranet and internet websites for GIS data portals. Other online LiDAR distribution websites have successfully employed suites of open-source software to build their IT infrastructure and to handle large amounts of point cloud data. Their software and system designs can reduce startup time and the risks for an online LiDAR distribution website portal.

---

[20] https://connectdot.connectsolutions.com/p7dw13s1vw5/

[21] http://aii.transportation.org/Documents/UPlan/uplan-fastfacts-aplan-az.pdf and

[22] http://aii.transportation.org/Documents/UPlan/uplan-fastfacts-iplan-id.pdf

[23] http://aii.transportation.org/Documents/UPlan/uplan-fastfacts-mdt-agol-mt.pdf

[24] http://aii.transportation.org/Documents/UPlan/uplan-fastfacts-pennshare-pa.pdf

CHAPTER 6:
SOFTWARE FOR DISTRICT 4 MTLS DATA MANAGEMENT AND VISUALIZATION

Introduction

MTLS GIS data provides a global view of MTLS data availability. It enables users to perform GIS analysis, as well as manage and keep track of MTLS data. The GIS data facilitate the quick retrieval of MTLS point cloud data, which leads to better decision-making, communication, and planning for projects and data gathering.

Software tools are available to generate GIS data from LiDAR data, including the following:

1. LASTools, Virtual Geomatics, Safe Software Feature Manipulation Engine (FME) suite, and ESRI ArcGIS suites can create shape (shp) files for the outline of LAS files.

2. Trimble Trident can export MX8 MTLS data to Keyhole Markup Language (KML) or shp files. AHMCT researchers have not found any third party software that would read the Trimble MX8 / Trident proprietary formatted data.

However, using these tools to extract GIS data from one folder or one project at a time can be quite labor intensive and time consuming, with a complicated workflow. In addition, the user has to integrate all the GIS data from various GIS files created from different software tools into a single GIS dataset. Moreover, a separate process has to be created to perform postmile lookup using the Caltrans PM Lookup service. This process must be automated and simplified to ensure the accuracy, reliability, and timeliness of GIS data from the MTLS dataset in Trimble MX8 / Trident proprietary format. Detailed MTLS data format information is provided in Chapter 4 and Table 4.2

AHMCT LidarCrawl Software for Automated MTLS GIS Data Extraction
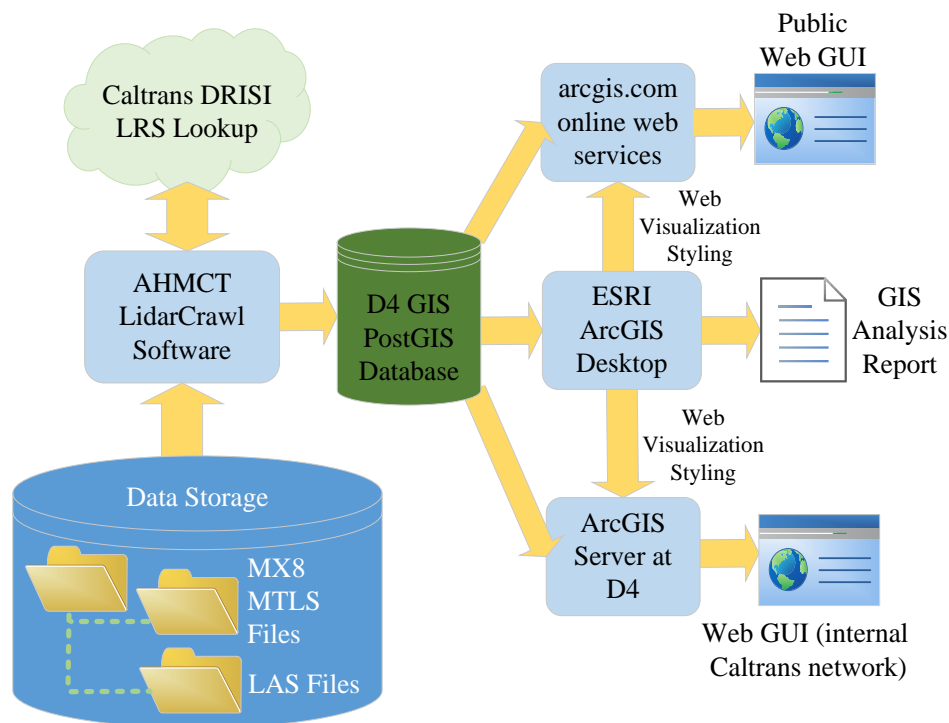
Figure 6.1: LidarCrawl software data flow diagram

Overview

The LidarCrawl software was developed to automate the extraction of project and file metadata from MTLS datasets (Caltrans' Trimble MX8 MTLS data and LAS version 1.0 to 1.4 files). LidarCrawl is written in the Java programming language. The Jackcess open-source Java library[25] is used for reading Microsoft Access database files. In addition, the Bitbucket Java library,[26] employing Apache License version 2.0, is used for reading the SQLite database. The JSON-simple[27] Java library is used to interface with JavaScript Object Notation (JSON) files. The application crawls directory trees, locating and processing MTLS datasets and project database files, and deriving associated metadata and statistics. This data is then output to a PostgreSQL[28] (Structured Query Language) / PostGIS[29] database, which can be subsequently browsed and queried via GIS software tools such as ESRI ArcMap or QGIS.[30] The MTLS GIS data can also be supported by an ESRI ArcGIS server to allow users to view the data using a web browser. Figure 6.1 illustrates the data flow/workflow for MTLS GIS data extraction using LidarCrawl.

---

[25] http://jackcess.sourceforge.net/

[26] https://bitbucket.org/xerial/sqlite-jdbc

[27] https://github.com/fangyidong/json-simple/wiki

[28] https://www.postgresql.org/

[29] http://postgis.net/

[30] http://www.qgis.org/

Table 6.1: Java libraries used in the LidarCrawl software

Java library	Usage	Web link	License
Oracle Java JDK ver. 8	Java programming	http://www.oracle.com/technetwork/java/javase/downloads/index.html	Oracle License[31]
Apache Commons Lang	Helper classes for java.lang	https://commons.apache.org/proper/commons-lang/	Apache License version 2.0
Apache Commons Logging	logging interface	https://commons.apache.org/proper/commons-logging/	Apache License version 2.0
Bitbucket Java library	Reading SQLite database	https://bitbucket.org/xerial/sqlite-jdbc	Apache License version 2.0
Jackcess	Reading Microsoft Access database	http://jackcess.sourceforge.net/	Apache License version 2.1.0
JSON-simple	Reading JSON files	https://github.com/fangyidong/json-simple/wiki	Apache License version 2.0
PostgreSQL JDBC driver	JDBC driver for PostgreSQL databases	https://jdbc.postgresql.org/	BSD license

The LidarCrawl software also performs PM lookup using the Caltrans DRISI Linear Referencing System (LRS) web service. The LidarCrawl function yields the county, route, and PM information for each MTLS project and "run," or MTLS data recording session.

The LidarCrawl software source code is a part of the project deliverables to Caltrans D4 and is available to other Caltrans operational units upon request. The project's digital files deliverable contents and description are provided in Appendix E.

LidarCrawl runs on Windows and Linux platforms. Although it should work on Mac OS X, its functionality has not yet been tested on the platform. LidarCrawl requires Java version 8 or higher. Filesystems to be crawled should be reachable via a drive letter and/or a standard filesystem path.

Recent versions of PostgreSQL and the PostGIS extensions should be used on the database server intended for LidarCrawl use. The PostGIS database (with an open, standard, well-defined and supported GIS format) GIS data can be converted to other commonly used GIS data formats such as ESRI shp files and KML using the ArcGIS software suite (used by Caltrans) or open-source QGIS software. The exported files from ArcGIS may be displayed in Google Earth or Google Maps. D4 ROW Records, Data, and GIS Application Branch maintains a PostgreSQL/PostGIS database server for their existing GIS data. The database server maintenance includes operating system updates, backup, and upgrades, as well as PostgreSQL database software updates and database backups. The use of D4's current PostGIS database

---

[31] http://www.oracle.com/technetwork/java/javase/terms/license/index.html

server setup would eliminate the need for additional IT infrastructure, such as a new server or software, and future maintenance costs. D4 GIS users would access the MTLS GIS data as they would other GIS data. Further, the MTLS GIS data may be integrated and displayed using the existing centralized D4 web GIS portal, removing the user's burden of going to another website for data. The D4 ROW Records, Data, and GIS Application Branch staff would make the MTLS GIS data available via the D4 ROW GIS data web-portal. D4 Surveys is responsible for updating the MTLS GIS database when new MTLS project is added to the mtls_pro volume by running the LidarCrawl software.

LidarCrawl will supply the critical MTLS GIS data. The software development avoided duplication of work. Software modules can be added to accommodate other Caltrans districts' or DRISI's GIS IT infrastructure if the PostGIS database is not being used for their GIS database should statewide deployment of LidarCrawl be desirable for Caltrans.

Mode of Operation

LidarCrawl is a user-level application that is intended for manual, discrete execution. No "system service" component needs to be kept running on a server. LidarCrawl only needs to be run when new LiDAR scan datasets have been added to the filesystem(s) or when existing datasets have been modified, relocated, or removed. LidarCrawl overwrites the exiting MTLS GIS data in the PostGIS database every time it runs. Java 8 must be installed before running LidarCrawl software. The LidarCrawl User Manual is available in Appendix B.

Workflow

1. D4 Surveys MTLS post-processing staff(s) would run the LidarCrawl software once when new post-processed MTLS data is added to the **mtls_pro** volume in order to update the MTLS GIS database. A PostgreSQL database username and password is provided by D4 ROW Records, Data, and GIS Application Branch staff so that D4 survey MTLS post-processing staff(s) can update the MTLS GIS database using the LidarCrawl software. Detailed procedures are provided in Appendix B.

2. D4 ROW Records, Data, and GIS Application Branch staff may change the visual display styling of the MTLS GIS displayed on their GIS web portal.

3. D4 Surveys MTLS post-processing staff(s) should check the GIS web portal to ensure the data gets updated.

Table 6.2: Personnel roles

User	MTLS GIS database access	Role descriptions
D4 ROW Records, Data, and GIS Application Branch staff	Read and Write	• Maintain, update, and backup PostgreSQL database • Modify MTLS GIS data display styling on web portal
D4 IT staff	Read	• Maintain, update, and backup database server operating system
D4 ROW Surveys MTLS post-processing staff(s)	Read and Write	• Update MTLS GIS database when new MTLS data is added

Viewing the Result

The LidarCrawl software creates three layers in the PostGIS MTLS database:

1. The **projects** layer is a polygon shape layer containing all of Caltrans MX8 projects' extent outlines with associate attributes (see Appendix C, Table C.1 for MTLS project attribute table details)

2. The **run** layer is a polyline shape layer comprising all of Caltrans MX8 projects' "run," or recording session, vehicle trajectory paths' polylines with associated attributes (see Appendix C, Table C.2 for "run" attribute table details)

3. The **las** layer is a polygon shape layer holding all LAS file boundary outlines with associate attributes (see Appendix C, Table C.3 for LAS file attribute table details)

The PostGIS MTLS database data can be visualized using ESRI ArcMap or open-source GIS software, such as QGIS. ESRI ArcMap can be used to set and change the visualization styling of the GIS data; the visualization styling may be published to the ArcGIS server for all web users. Figures 6.2, 6.3, and 6.4 show the **project**, **run**, and **las** GIS layers displayed in QGIS overlaid on a Microsoft Bing Map.
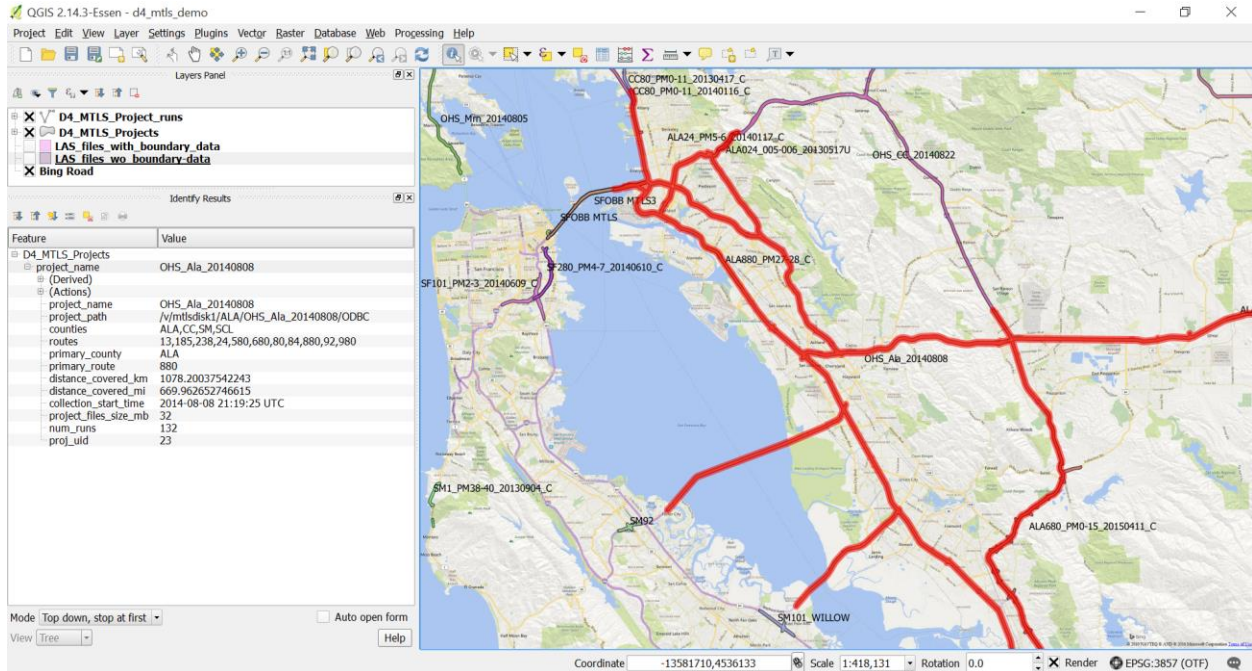
Figure 6.2: MTLS project layer data displayed in QGIS. Project named OHS_Ala_20140808 (highlighted in red on the right). MTLS project attributes are shown in the left panel.
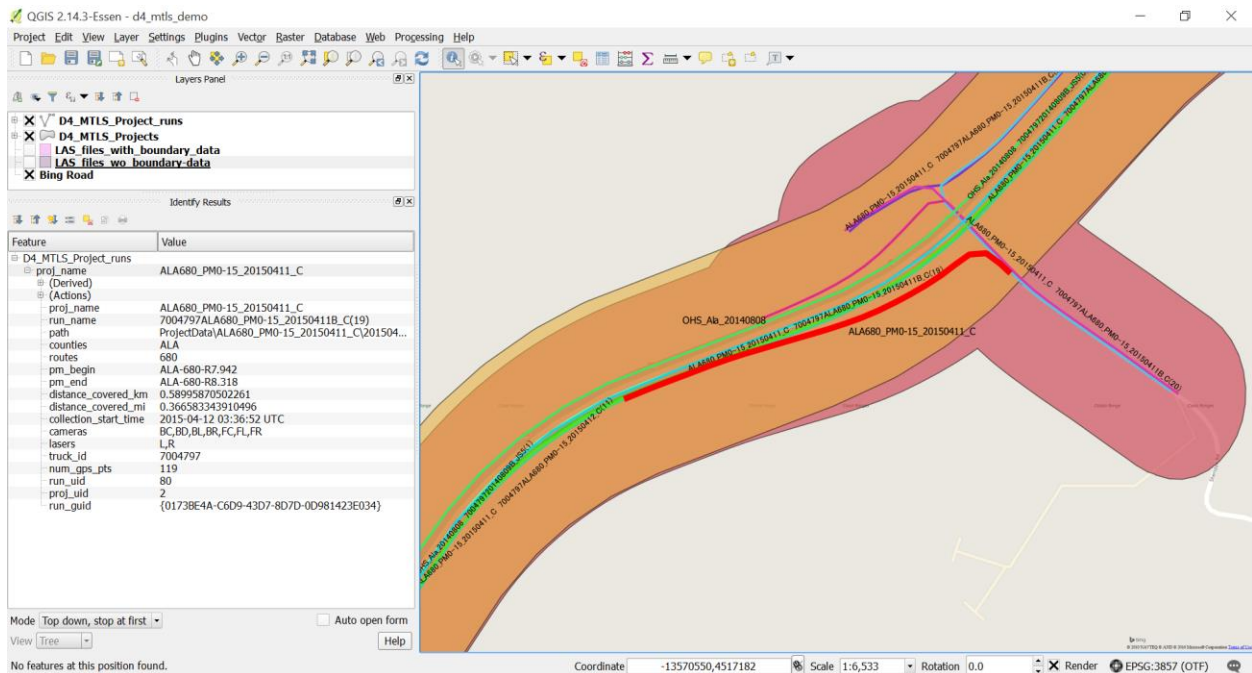
Figure 6.3: MTLS run layer data displayed in QGIS. Run named 7004797ALA680_PM0-15_20150411B_C (highlighted in red on the right). Attributes are shown in the left panel.
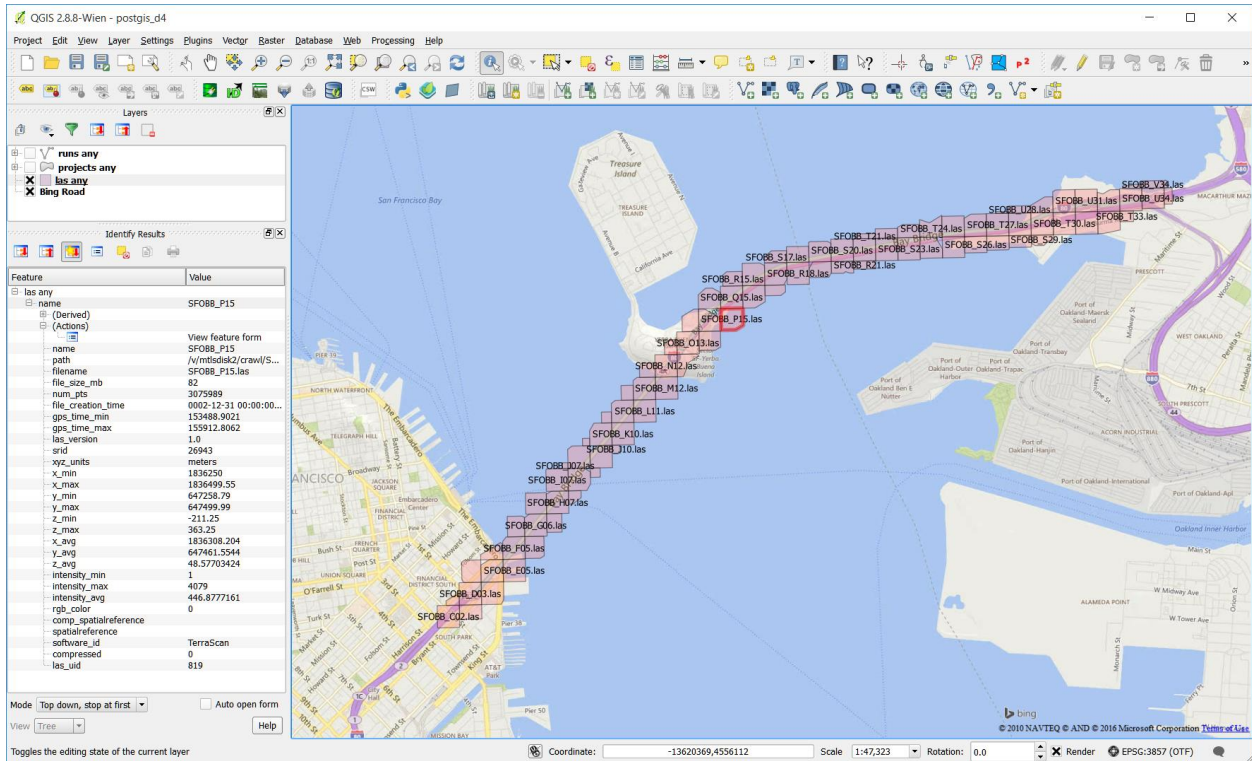
Figure 6.4: MTLS las layer data displayed in QGIS. Las file named SFOBB_P15 (highlighted in red on the right). Attributes are shown in the left panel.

CHAPTER 7:
CONCLUSIONS AND FUTURE RESEARCH

Lesson Learned and Challenges

- Use of existing IT infrastructure and GIS data portal decreased implementation time for the LidarCrawl tool.

- D4 and HQ IT's early involvement was instrumental in the NetApp system procurement and implementation as well as the IT infrastructure upgrade. The support of D4 IT and the D4 Right of Way, Records, Data & GIS Application Branch was crucial to the success of the project.

Key Contributions

- Pilot project to demonstrate the need and the use of MTLS GIS data to improve MTLS data management.

- LiDAR crawler software to automate MTLS GIS data extraction and thereby reduce labor and improve MTLS data management.

- A recommended MTLS storage directory structure for ease of data retrieval.

Future Work

- Present results to Caltrans MTLS users in other Caltrans districts

- Support for other Caltrans districts and HQ:

  o Work with other Caltrans districts to deploy the LidarCrawl software

  o Modify LidarCrawl software as required to best suit each Caltrans district's specific needs

- Work with all Caltrans districts and HQ to develop an MTLS data management plan and policies, including a data retention policy, data storage policy, MTLS data distribution and sharing policy, data security policy for security-sensitive data such as major bridge structures, policy on the use of internet cloud-based storage and backup, and recommendations for the standardization of project survey notes and reporting format. The NCHRP 15-44 guideline and NCHRP Report 784 provided recommendations for the information that should be included in project survey notes and reporting. Noted concerns regarding data security and sensitivity are related mainly to the San Francisco/Oakland Bay Bridge (SFOBB) data.

- Support for web-based point cloud viewing

- Include other point cloud data formats, such as

- o Trimble *.gps, LAS20, and LAS 2.0

- o LAZ (LASZip)

- o Riegl VMX-1HA (the next generation Caltrans MTLS system) raw data files (format to be determined)

- Provide support for photolog images

    - o Photolog images from various sources, such as:

        - Pavement condition survey

        - Trimble MX8 MTLS system

        - Riegl VMX-1HA MTLS system

        - MTLS consultant and contractor-provided images

        - Caltrans Traffic Operations photolog images (legacy photos and current images)

    - o Ladybug 360 degree images from raw Ladybug *.pgr proprietary binary data format

        - HDR image support

        - Parallel data processing

REFERENCES

1.   "Guidelines for the Use of Mobile LiDAR in Transportation,"
        http://learnmobilelidar.com/.
2.   C.G.T. Committee, "Geospatial Strategic Direction 2014," 2014.
3.   K. Yen, B. Ravani, and T. Lasky, "Mobile Terrestrial Laser Scanning Workflow
        Development, Technical Support and Evaluation," Rept. # UCD-ARR-14-06-14-01,
        2014.
4.   K.S. Yen, K. Akin, A. Lofton, B. Ravani, and T.A. Lasky, "Using Mobile Laser Scanning
        to Produce Digital Terrain Models of Pavement Surfaces," AHMCT Rept. # UCD-ARR-
        10-11-30-01, 2010.

APPENDIX A:
NETAPP SYSTEM INSTALLED IN DISTRICT 4

Table A.1 NetApp system components list as procured by D4

Item	Part #	Description	QTY
1	PS-RDSP-OPSMGR-BASE-ZA	Operations Manager Imp! Base RDS+,ZA	1
2	PS-RDSP-OPSMGR-REPORT-ZA	Ops Mgr Customer Reporting Add-on RDS+,ZA	1
3	FAS8040-R6	FAS8040 Model	1
	CNA-OB-PR-FC-TGT-EOE-EOF	CNA Onboard FC TARGET,eOe-eOf	2
	CNA-OB-PR-FC-TGT-EOG-EOH	CNA Onboard FC TARGETE.eOg-eOh	2
	D0C-80XX-C	Documents,80xx.-C	1
	DS4246-0772-24A-OP-R6-C	OSK SHLF. 24x3.0TB.7.2K,6G,OP,-C	5
	FAS 8040A-001-R6	FAS8040 High Availability System	2
	FAS8040A-CNTLR-R6-C	FAS8040,HA.Controller,-C	2
	MULTIPATH-C	Multipath configuration	1
	STACKS	Storage Stacks Attached Quantity	2
	X1132A-EN-R6-C	ADPT 4-Port FCP 8Gb Target Adapter PCIe,EN,-C	2
	X1960-R6-C	ClusterNet lnter-connect.16Pt.1 OGb,-C	2
	X197 4A-R6-C	Flash Cache 1 TB PC le Module 2,-C	2
	X1985-R6-C	12-Node Cluster Cable Label Kit.-C	1
	X2065A-EN-R6-C	HBA SAS 4-Port Copper 3/6 Gb QSFP PCEe,EN,-C	2
	X4525A-R6-C	Nameplate,FAS8040,-C	1
	X6227-R6-C	Chassis,FAS8040/60/80 W/CNTRL Slots.AC PS,-C	1
	X6554-R6-C	Cable, Cntlr-Shelf/Switch, 15m,LC/LC, Op,-C	36
	X6557-R6-C	Cable. SAS Cntlr-Shelf/Shelf-Shelf/HA, O.5 m,-C	6
	X6558-R6-C	Cable. SAS Cntlr-Shelf/Shelf-Shelf/HA, 2m.-C	4
	X6560-R6-C	Cable, Ethernet, 0.5m RJ45 CAT6.-C	6
	X6561-R6-C	Cable, Ethernet, 2m RJ45 CAT6,-C	3
	X6566B-05-R6-C	Cable, Direct Attach CU SFP+ 10G,0.5M,-C	4
	X6584-R6-C	Cable, Ethenet,1m RJ45 CAT6,-C	1
	X6585-R6-C	Cable, Ethernet, 3m RJ45 CAT6,-C	1
	X6589-R6-C	SFP+ Optical 10Gb Shortwave,-C	4
	X6594-R6-C	Cable, SAS Cntlr-Shelf/Shelf-Shelf,HA, 1 m,-C	4
	X6596-R6-C	SFP+ FC Optical 16Gb,-C	8
	X6599A-R6-C	SFP+ Optical 10Gb Shortwave,FAS80XO.-C	8
	X800-42U-R6-C	Power Cable, ln-Cabinet.C13-C14,-C	16
	X870E-EN-R6-C	Power Cable, ln-Cabinet.C13-C14,-C	1
	X871 2C-EN-R6-C	POU, 1-Phase,12 Outlet.30A,NEMA,EN,-C,R6	4
	X8773-R6-C	Mounting Bracket, Tie-Down, Multiple,-C, R6	1
	X8783A-R6-C	Rail Kit 111,Cabinet,-C	6
	OS-ONTAP-CAP1 -0P-C	OS Enable, Per-0. 1 TB,ONTAP,Cap-Stor,OP.-C	3600
	OS-SW-ONTAP-CAP1-0P	OS SW,Per-0.1 TB.ONTAP,Capacity-Stor,OP	3600
	PREMBDLPRICE-2-8040A	Pricing Part-2,PREMBDL,8040A	2
	SW-2-8040A-C I FS-PB-C	SW-2,CIFS,PB,8040A,-C	2
	SW-2-8040A-FCP-PB-C	SW-2,FCP,PB,8040A,-C	2
	SW-2-8040A-FLEXCLONE-PB-C	SW-2, Flexclone, PB,8040A,-C	2
	SW-2--8040A-ISCSI-PB-C	SW-2,iSCSI , P6,804CA,-C	2
	SW-2-8040A-NFS-PB-C	SW-2,NFS.PB,8040A,-C	2
	SW-2-8040A-PREMBNDL-C	SW-2,Premium 8NDL,8040A,-C	2
	SW-2-8040A-SMBR-P8-C	SW-2,Single Mailbox Recovery.PB,8040A,-C	2
	SW-2-8040A-SMGR-PB-C	SW-2,SMGR Suite.PB,8040A,-C	2
	SW-2-8040A-SMIRROR-PB-C	SW-2, SnapMirror. PB,8040A, -C	2

	SW-2-8040A-SREST0RE-PB-C	SW-2,SnapRestore.PB,8040A,-C	2
	SW-2-8040A-SVAULT-PS-C	SW-2.SnapVault.P8,8040A,-C	2
	SW-2-8040A-VN-STOR-C	SW-2,FlexArray Virtualization.8040A,-C	2
	SW-ONTAP8.2. 1-CLM	SW, Data ONTAPB.2 .1 .Cluster-Mode	2
	0S-SSP-ONTAP-CAP1-0P	OS SSP.Per-0.1 TB,ONTAP,Capacity-Stor,OP	3600
	CS-02-4HR-VA	SupportEdge Premium 4hr Onsite,VA 5 YR	1
	CS-02-4HR-VA	SupportEdge Premium 4hr Onsite,VA 2 YR	1
4	SW-2-CL-LICENSEMGR	License Manager,SW-2,Data ONTAP 8.x.Nocte	1
	SW-2-CL-BASE	SW-2,Base,C L. Node	1
5	PS-RDSP-STOR-IMPL-4NDE-ZA	1-4 FAS Cntlr or 2HA Add-on RDS+, ZA	1
6	PS-RDSP-STOR-IMPL-BASE-ZA	Storage Implementation Base RDS+,ZA	1
7	X6554-R6	Cable,Cntlr-Shelf/Switch, 15m,LC/LC,Op	10
8	X6546-R6	Cable, Cntlr-Shelf/Switch, 30m, LC/SC.Op	10
9	X6589-R6	SFP+Optical 10Gb Shortwave	4
10	ED-TU-1-ZA	Training Units, 1, ZA. Exp.1 yr from invoice date	160

APPENDIX B:
LIDARCRAWL SOFTWARE USER MANUAL

Steps

1. Double click on the "LidarCrawl.jar" Java executable file to run the LidarCrawl software. The user would see the LidarCrawl GUI appear on screen, as shown in Figure B.1.
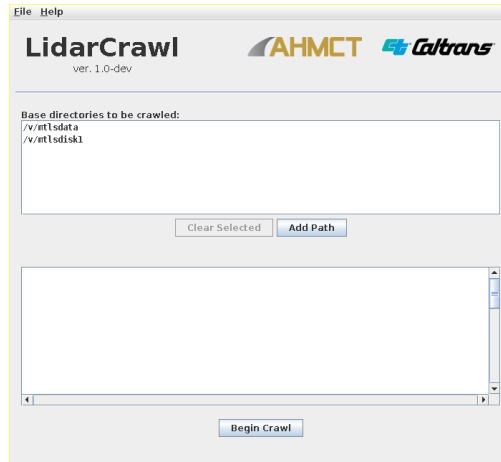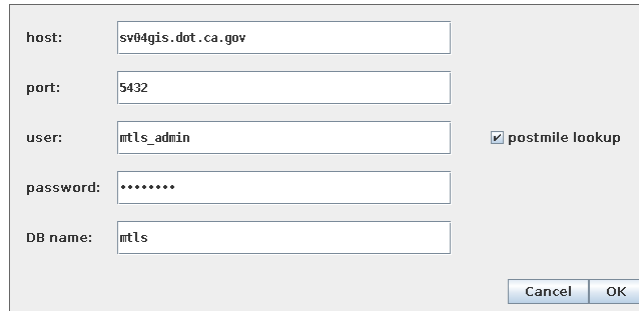
Figure B.1: LidarCrawl GUI

2. The user is required to configure the GIS database connection (once per user per machine)

    a. Click on "File" and then "Settings" on the menu bar, as shown in Figure B.2.
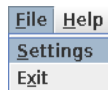
Figure B.2: LidarCrawl setting GUI

    b. Enter the appropriate PostGIS host machine name, port number, database username, password, and the database name in the dialog box shown in Figure B.3. The user may elect to perform postmile lookup by checking the "postmile lookup" check box on the right.

Figure B.3: LidarCrawl database setup GUI

    i. **host**: the hostname or IP address of the PostgreSQL service to be used

    ii. **port**: the port number of the PostgreSQL service to be used (usually 5432)

    iii. **user**: the PostgreSQL database username to be used to write to the LidarCrawl database

    iv. **password**: the password of the user specified in the "user" field

    v. **DB name**: the name of the PostGIS database to which the LidarCrawl output will be written (e.g., mtls as in Figure B.3)

    vi. **postmile lookup**: whether or not to perform postmile lookups for the crawl, using the Caltrans postmile lookup service

  c. Click "OK" when completed.

3. Select the directories that the user would like LidarCrawl to recursively search for MTLS data.

  a. One or more base directories must be assigned for the crawl. These are directories that LidarCrawl will recursively search (i.e., including all the directories beneath them). All the LiDAR datasets that the user wants to be included in the crawl should be located somewhere beneath these directories.

  b. The current list of base directories is displayed at the top of the main window. To add a new base directory to the list, click on the "Add Path" button shown in Figure B.1, then select the desired directory/folder and click on the "Open" button, as shown in Figure B.4.

  c. To remove one or more base directories from the list, select them with the mouse, then click on the "Clear Selected" button.
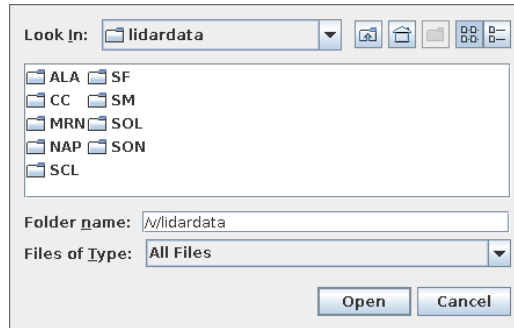
Figure B.4: LidarCrawl "add path" GUI

4. At this point, the application is ready to perform a crawl. LidarCrawl stores its configuration in a directory named ".lidarcrawl" located in the user's home directory, so that the users won't need to reconfigure it before every run. The configuration is associated with the user and the machine. If the LidarCrawl software is run by a different user or the same user on a different machine, the configuration process will have to be repeated.

5. To initiate the crawling process, click on the "Begin Crawl" button located at the bottom of the LidarCrawl GUI window, as shown in Figure B.1. The crawl progress can then be monitored via the crawl log and the progress bar dialog that appears once the crawl has begun.

6. To cancel a crawl, press the "Cancel" button. All the database changes that LidarCrawl makes will take place in a single database transaction at the end. Therefore, if a crawl is cancelled or aborted due to any error, no changes will be written to the database and the database contents will remain consistent. However, it is important to note that LidarCrawl regenerates the entire database upon each successful run. Thus, if the crawl is successful, all the previous contents of the database will be wiped and rewritten with the results from the new crawl.

7. Click on "File" and then "Exit" to quit the LidarCrawl program.

APPENDIX C:
MTLS POSTGIS DATABASE ATTRIBUTE TABLES

Table C.1 Project attributes

Attribute name	Attribute description	Example attribute data
project_name	Project name	OHS_Sol_20140722
project_path	Folder path where the MTLS project data are stored	/v/mtlsdata/projdb
counties	County abbreviation where the project is located	CC,SOL,YOL
routes	CA highway route number the project covers	37,505,680,780,80
primary_county	County abbreviation where the majority of project is located	SOL
primary_route	CA highway route number the majority of the project covers	80
distance_covered_km	Project total distance traveled in kilometers	560.3285
distance_covered_mi	Project total distance traveled in miles	348.172
collection_start_time	Project data collection start time in Coordinated Universal Time (UTC)	2014-07-23 00:16:17 UTC
project_files_size_mb	Total project files size in megabytes	292
num_runs	Number of "runs" in the project	44
proj_uid	Project unique identification generated by LidarCrawl	4

Table C.2 Run attributes

Attribute name	Attribute description	Example attribute data
proj_name	Project name	SCL101_PM0-27_20130207_U
run_name	"Run" name	7004797SCL101_PM0-27_20130207_U(2)
path	Folder path where the MTLS data are stored	ProjectData\SCL101_PM0-27_20130207_U\20130207
counties	County abbreviation where run is located	SCL
routes	CA highway route number the run covers	101
pm_begin	Run begin postmile	SCL-101-R18.447
pm_end	Run end postmile	SCL-101-R26.585
distance_covered_km	Run distance traveled in kilometers	13.10558
distance_covered_mi	Run distance traveled in miles	8.143428
collection_start_time	Run data collection start time in UTC	2013-02-08 02:56:03 UTC
cameras	Cameras used	BC,BD,BL,BR,FC,FL,FR
lasers	LiDAR sensors used (L: left laser and R: right laser)	L,R
truck_id	Caltrans MTLS vehicle identification #	7004797
num_gps_pts	Number of GPS points (once / second) in the run vehicle trajectory record in the project database	3260
run_uid	Run unique identification generated by LidarCrawl	1
proj_uid	Project unique identification generated by LidarCrawl	1
run_guid	"Run" unique identification generated by Trimble Trident Capture MTLS data collection software	430890D1-7617-4C35-A329-4F4C8CF2D39A

Table C.3 LAS attributes

Attribute name	Attribute description	Example attribute data
name	File name	SJ101_3 - Scanner 1 - 140919_190908_Scanner_1 - originalpoints
path	Folder path where the LAS file is stored	/v/mtlsdisk2/crawl/Psomas_SanJose -2ND Delivery/San Jose_101/Color_LAS/Northbound_Fast_Lane
filename	File name with extension	SJ101_3 - Scanner 1 - 140919_190908_Scanner_1 - originalpoints.las
file_size_mb	LAS file size in megabytes	991
num_pts	Number of points in the LAS file	30564981
file_creation_time	File creation time in UTC	2014-10-06 00:00:00 UTC
gps_time_min	Minimum GPS time of the points in the LAS file	500948.2
gps_time_max	Maximum GPS time of the points in the LAS file	501048.2
las_version	LAS file version number	1.2
srid	Spatial reference identification number of the points in the LAS file	26910
xyz_units	LAS file XYZ point units	US survey ft
x_min	Minimum X value	1998361
x_max	Maximum X value	2006371
y_min	Minimum Y value	13519733
y_max	Maximum Y value	13524951
z_min	Minimum Z value	-45.5675
z_max	Maximum Z value	510.8525
x_avg	Average X value	2002399
y_avg	Average Y value	13522423
z_avg	Average Z value	247.2547
intensity_min	Minimum intensity value	0
intensity_max	Maximum intensity value	65535
intensity_avg	Average intensity value	18508.2
rgb_color	Is the point cloud colorized?	1
comp_spatialreference	LAS file spatial reference description in Well-Known Text (WKT) format	LOCAL_CS["NAD83, GRS80, UTM zone 10N, Geoid2012A_CONUS.gdf",GEOGCS["unknown",DATUM ["unknown",SPHEROID["unretrievable - using WGS84",6378137,298.257223563]],PRIMEM["Greenwich" ,0],UNIT["degree",0.01745329251994433]],UNIT["US survey foot",0.3048006096012192]]
spatialreference	LAS file spatial reference description in Well-Known Text (WKT) format	LOCAL_CS["NAD83, GRS80, UTM zone 10N, Geoid2012A_CONUS.gdf",GEOGCS["unknown",DATUM ["unknown",SPHEROID["unretrievable - using WGS84",6378137,298.257223563]],PRIMEM["Greenwich" ,0],UNIT["degree",0.01745329251994433]],UNIT["US survey foot",0.3048006096012192]]
software_id	Software used to create the LAS file	RiPROCESS 1.6.1
compressed	Is data compressed?	0
las_uid	LAS file unique identification generated by LidarCrawl	1

APPENDIX D:
CLOUD-BASED BACKUP SOLUTION DETAILS

Table D.1 Detailed cloud-based backup solution cost calculation information

Company	Service	Location	Response Time	Storage (GB/mo)	Write (GB/mo)	Retreival (GB/mo)	Network Ingress (GB/mo)	Network Egress (GB/mo)	Operations Put (/10,000 ops)	Operations Get (/10,000 ops)	Free Peak Transfer Rate	Overage Transfer Rate	Minimum Storage (mo)
Amazon	Glacier	OR	3-5hr	$0.007	$0.000	5% Free then $0.01	$0.000	$0.081			0.05 * Storage / 30 / 24/hr	Huge	3
Google	Nearline	West	3 sec	$0.010	$0.000	$0.010	$0.000	$0.084	$0.100	$0.010	4MB/s per TB	None	1
CalCloud	Teir 3	CA	100 ms	$0.161									
Microsoft	Azure Cool	Central		$0.010	$0.003	$0.010	$0.000	$0.079	$0.100	$0.010			

Detailed Pricing Information

1. https://cloud.google.com/storage/pricing/
2. https://aws.amazon.com/glacier/pricing/
3. https://azure.microsoft.com/en-us/pricing/details/storage/
4. http://www.servicecatalog.dts.ca.gov/services/cloud/calcloud/iaas/specifications.html#storage

An evaluation of Cloud-based storage performance is available at http://blog.zachbjornson.com/2015/12/29/cloud-storage-performance.html.

APPENDIX E:
PROJECT DIGITAL FILES DELIVERABLE CONTENTS

Overview

The LidarCrawl application is a standalone, desktop application written in Java that is distributed as a single Java Archive (JAR) file. The source code for LidarCrawl is distributed as a ZIP file, and contains the following directories:

- **assets**: static application assets (images, etc.)
- **bin**: build tools
- **capm**: a library to perform postmile queries against the Caltrans public postmile data service
- **external**: various external libraries used by LidarCrawl
- **lidarcrawl**: the main component of the LidarCrawl application
- **simplog**: a library to manage LidarCrawl logging
- **sql**: SQL scripts required to initialize the database
- **uibase**: UI base classes (exported from IntelliJ IDEA)

Building LidarCrawl

Building LidarCrawl from source requires the following software packages:
- Java SE Development Kit 8 (JDK 8)[32]
- Apache Ant 1.9[33]

The delivered version of LidarCrawl was built using the Debian 8 (amd64) OS. To build LidarCrawl from source, expand the LidarCrawl source ZIP archive and navigate to the base directory of the source tree. Then execute the following commands:

```
bin/clean
bin/build
```

The output of the build process will be a single file, `lidarcrawl.jar`, located in the base directory of the source tree.

---

[32] http://www.oracle.com/java
[33] http://ant.apache.org/

<u>**External Libraries Used by LidarCrawl**</u>

The external libraries (Table E.1) used by LidarCrawl can be found in the `external` directory of the LidarCrawl source tree. These are as follows:

Table E.1: Java library used LidarCrawl software

Java library	Usage	Java library file name	Web link
Apache Commons Lang	Helper classes for java.lang	commons-lang-2.6.jar	https://commons.apache.org/proper/commons-lang/
Apache Commons Logging	logging interface	commons-logging-1.1.3.jar	https://commons.apache.org/proper/commons-logging/
Bitbucket Java library	Reading SQLite database	sqlite-jdbc-3.8.11.2.jar	https://bitbucket.org/xerial/sqlite-jdbc
Jackcess	Reading Microsoft Access database	jackcess-2.1.3.jar	http://jackcess.sourceforge.net/
JSON-simple	Reading JSON files	json-simple-061116.jar	https://github.com/fangyidong/json-simple/wiki
PostgreSQL JDBC driver	JDBC driver for PostgreSQL databases	postgresql-9.4.1208.jar	https://jdbc.postgresql.org/

<u>**Database Initialization**</u>

The SQL code required to initialize (or reinitialize) the LidarCrawl database can be found in the sql/d4 directory of the LidarCrawl source tree in a file called schema_reset.sql. This code can be executed on the database in whichever way the user is accustomed to running SQL code. One such way is to use the PostgreSQL psql tool. For example:

psql –h sv04gis.dot.ca.gov mtls mtls_admin –f sql/d4/schema_reset.sql

This command should only be used to initialize the database or reinitialize it. **Running this command will delete all existing data.**